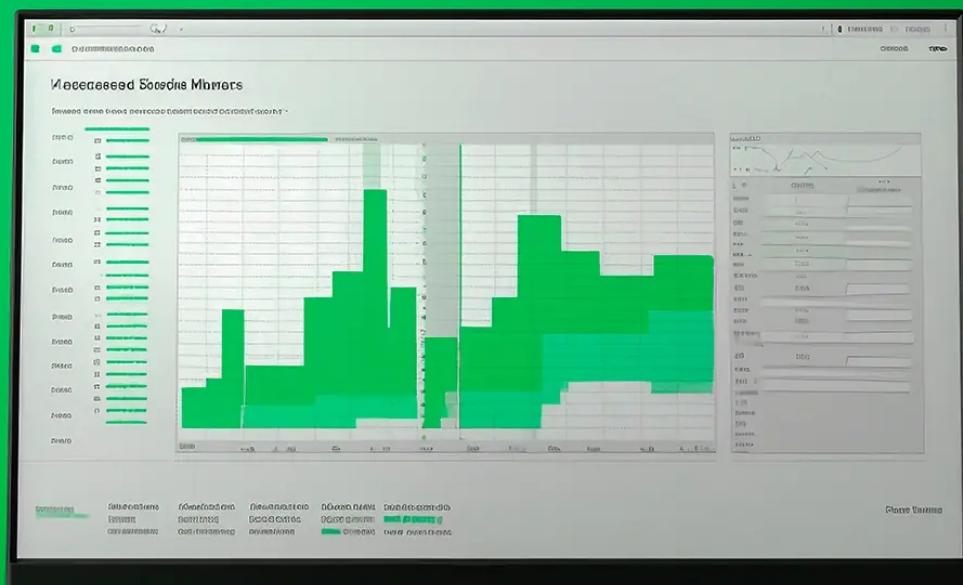




YAYASAN PRIMA AGUS TEKNIK



# PRAKTIK MODEL KEUANGAN DENGAN EXCEL & VBA



Dr. Agus Wibowo, M.Kom, M.Si, MM.

Dr. Agus Wibowo, M.Kom, M.Si, MM.

# PRAKTIK MODEL KEUANGAN DENGAN EXCEL & VBA



YAYASAN PRIMA AGUS TEKNIK

**PENERBIT :**  
YAYASAN PRIMA AGUS TEKNIK  
Jl. Majapahit No. 605 Semarang  
Telp. (024) 6723456. Fax. 024-6710144  
Email : penerbit\_ypat@stekom.ac.id

ISBN 978-623-8642-91-5 (PDF)



9

786238

642915

## **Praktik Model Keuangan dengan Excel & VBA**

### **Penulis :**

Dr. Agus Wibowo, M.Kom, M.Si, MM.

**ISBN : 978-623-8642-91-5**

### **Editor :**

Dr. Joseph Teguh Santoso, S.Kom., M.Kom.

### **Penyunting :**

Dr. Mars Caroline Wibowo. S.T., M.Mm.Tech

### **Desain Sampul dan Tata Letak :**

Irdha Yuniato, S.Ds., M.Kom

### **Penebit :**

Yayasan Prima Agus Teknik Bekerja sama dengan  
Universitas Sains & Teknologi Komputer (Universitas STEKOM)

**Anggota IKAPI No:** 279 / ALB / JTE / 2023

### **Redaksi :**

Jl. Majapahit no 605 Semarang

Telp. 08122925000

Fax. 024-6710144

Email : [penerbit\\_ypat@stekom.ac.id](mailto:penerbit_ypat@stekom.ac.id)

### **Distributor Tunggal :**

#### **Universitas STEKOM**

Jl. Majapahit no 605 Semarang

Telp. 08122925000

Fax. 024-6710144

Email : [info@stekom.ac.id](mailto:info@stekom.ac.id)

Hak cipta dilindungi undang-undang

Dilarang memperbanyak karya tulis ini dalam bentuk dan dengan cara  
apapun tanpa ijin dari penulis

## KATA PENGANTAR

Puji syukur kami panjatkan ke hadirat Tuhan Yang Maha Esa, karena berkat rahmat dan karunia-Nya, kami dapat menyelesaikan karya ini yang berjudul "*Praktik Model Keuangan*". Karya ini disusun untuk memberikan pemahaman yang lebih mendalam mengenai penerapan model-model keuangan dalam pengelolaan keuangan yang efektif dan efisien di berbagai sektor.

Dalam dunia yang semakin berkembang pesat, baik secara global maupun dalam hal teknologi, pengelolaan keuangan menjadi semakin kompleks dan memerlukan pendekatan yang tepat. Oleh karena itu, pemahaman tentang praktik model keuangan yang baik dan benar sangat penting, baik untuk individu maupun organisasi, agar dapat membuat keputusan-keputusan yang tepat dan mendukung keberlanjutan serta kesuksesan jangka panjang. Melalui karya ini, kami berharap dapat memberikan wawasan tentang berbagai teknik dan pendekatan dalam penerapan model keuangan, yang dapat disesuaikan dengan kebutuhan dan tujuan masing-masing pihak.

Buku ini memberikan panduan komprehensif tentang berbagai fungsi dan fitur dalam Excel yang digunakan untuk analisis data, pemodelan keuangan, dan otomatisasi proses kerja. Dimulai dengan fungsi aritmatika dan logika inti di Bab 1, buku ini mengajarkan cara menggunakan fungsi dasar seperti IF, AND, OR, serta agregat seperti SUM, AVERAGE, dan COUNT untuk mempermudah analisis data. Bab 2 melanjutkan dengan penggunaan fungsi dan rumus array, yang penting dalam pengolahan data kompleks dan perhitungan berbasis matriks seperti alokasi biaya dan peramalan sumber daya.

Bab 3 memperkenalkan fungsi matematika seperti EXP, ABS, dan MOD, yang digunakan dalam perhitungan numerik dan analisis keuangan lebih lanjut. Fungsi keuangan di Bab 4 mengulas berbagai fungsi terkait investasi, arus kas, dan pengambilan keputusan finansial menggunakan FV, PV, NPV, dan IRR. Bab 5 mendalami fungsi statistik, mengajarkan aplikasi dalam analisis distribusi dan regresi untuk peramalan.

Bab 6 mengajarkan fungsi informasi, yang memungkinkan pengguna untuk mengevaluasi kondisi sel dan mengelola data dalam spreadsheet dengan lebih efisien. Fungsi tanggal dan waktu di Bab 7 memberikan dasar bagi perencanaan proyek dan estimasi sumber daya, termasuk pembuatan sumbu waktu yang tepat. Fungsi teks di Bab 8 mengajarkan cara menggabungkan dan memanipulasi teks, yang penting untuk membersihkan dan menyusun data. Bab 9 memfokuskan pada fungsi pencarian dan referensi, memberikan pengetahuan tentang teknik pencarian dan pemetaan data dalam rentang yang luas, sementara Bab 10 membahas filter, fungsi basis data, dan Pivottable, yang berguna untuk analisis dan penyajian data.

Bab 11 memperkenalkan pintasan dan fitur Excel lainnya, meningkatkan produktivitas dan efisiensi dengan alat bantu dan tips yang praktis. Bab 12 mengajarkan dasar-dasar VBA

(Visual Basic for Applications) untuk mengotomatisasi tugas dan meningkatkan efisiensi kerja. Lanjut ke Bab 13, di mana pembaca belajar bekerja dengan objek dan rentang dalam VBA untuk meningkatkan fleksibilitas dan dinamisitas manipulasi data. Akhirnya, Bab 14 mengajarkan cara mengendalikan eksekusi kode VBA, memberikan kontrol lebih besar atas proses otomatisasi dengan kotak input dan pesan, serta penggunaan fungsionalitas lainnya dalam VBA.

Secara keseluruhan, buku ini memberi wawasan mendalam dan keterampilan praktis untuk memaksimalkan penggunaan Excel, dari fungsi dasar hingga teknik pemrograman lanjutan dengan VBA, yang sangat berguna dalam analisis data dan pemodelan keuangan di dunia profesional.

Semarang, Maret 2025

Penulis

Dr. Agus Wibowo, M.Kom, M.Si, MM.

# DAFTAR ISI

Halaman Judul .....	i
Kata Pengantar .....	ii
Daftar Isi .....	iv
<b>BAB 1 FUNGSI ARITMATIKA DAN LOGIKA INTI .....</b>	<b>1</b>
1.1 Pendahuluan .....	1
1.2 Aplikasi Praktis.....	1
1.2.1 Contoh: IF, AND, OR, NOT .....	2
1.2.2 Contoh: MIN, MAX, MINA, MAXA .....	5
1.2.3 Contoh: MINIFS dan MAXIFS .....	5
1.2.4 Contoh: COUNT, COUNTA, COUNTIF dan Fungsi Serupa .....	6
1.2.5 Contoh: SUM, AVERAGE, AVERAGEA .....	7
1.2.6 Contoh: SUMIF, SUMIFS, AVERAGEIF, AVERAGEIFS .....	7
<b>BAB 2 FUNGSI DAN RUMUS ARRAY .....</b>	<b>18</b>
2.1 Fungsi Dan Rumus .....	18
2.2 Implementasi .....	18
2.3 Fungsi Array .....	20
2.3.1 Contoh: Jadwal Capex dan Penyusutan Menggunakan TRANSPOSE .....	20
2.3.2 Contoh: Alokasi Biaya dengan Perkalian Matriks Menggunakan MMULT ....	21
2.3.3 Contoh: Perhitungan Biaya Aktivitas dan Peramalan Sumber Daya .....	22
2.3.4 Contoh: Menjumlahkan Pangkat Bilangan Bulat dari 1 dan seterusnya .....	24
2.4 Aplikasi Praktis: Rumus Array .....	27
2.4.1 Contoh: Menemukan Item Positif Pertama dalam Daftar .....	27
2.4.2 Contoh: Menemukan Nilai Maksimum Bersyarat .....	29
2.4.3 Contoh: Nilai Maksimum Bersyarat dengan AGGREGATE .....	29
<b>BAB 3 FUNGSI MATEMATIKA.....</b>	<b>32</b>
3.1 Pendahuluan .....	32
3.2 Aplikasi Praktis .....	32
3.2.1 Contoh: EXP dan LN .....	32
3.2.3 Contoh: ABS dan SIGN .....	35
3.2.3 Contoh: MROUND, CEILING.MATH, dan FLOOR.MATH .....	38
3.2.4 Contoh: MOD .....	39
3.2.5 Contoh: SQRT dan POWER .....	40
3.2.6 Contoh: FACT dan COMBIN .....	40
3.2.7 Contoh: RAND () .....	42
3.2.8 Contoh: SINE, ASIN, DEGREES, dan PI () .....	43
3.2.9 Contoh: BASIS dan DESIMAL .....	44
<b>BAB 4 FUNGSI KEUANGAN .....</b>	<b>47</b>
4.1 Pendahuluan.....	47
4.2 Aplikasi Praktis .....	47

4.2.1 Contoh: FVSCCHEDULE .....	46
4.2.2 Contoh: FV dan PV .....	48
4.2.3 Contoh: PMT, IPMT, PPMT, CUMIPMT, CUMPRINC, dan NPER .....	50
4.2.4 Contoh: NPV dan IRR untuk Keputusan Beli atau Sewa .....	51
4.2.5 Contoh: SLN, DDB, dan VDB .....	54
4.2.6 Contoh: YIELD .....	55
4.2.7 Contoh: Durasi Arus Kas .....	56
4.2.8 Contoh: DURATION dan MDURATION .....	57
4.2.9 Contoh: PDURATION dan RRI .....	57
4.3 Fungsi Keuangan Lainnya .....	58
<b>BAB 5 FUNGSI STATISTIK .....</b>	<b>60</b>
5.1 Pendahuluan .....	60
5.2 Aplikasi Praktis: Posisi, Peringkat, Dan Nilai Pusat .....	60
5.3 Aplikasi Praktis: Penyebaran Dan Bentuk .....	67
5.4 Aplikasi Praktis: Hubungan Sama Dan Ketergantungan .....	76
5.5 Aplikasi Praktis: Distribusi Kemungkinan .....	83
5.6 Aplikasi Praktis: Informasi Tentang Analisis Regresi Dan Peramalan .....	94
<b>BAB 6 FUNGSI INFORMASI .....</b>	<b>102</b>
6.1 Pendahuluan .....	102
6.2 Aplikasi Praktis .....	102
6.2.1 Contoh: Komentar Rumus dengan ISTEXT, ISNUMBER, atau N.....	102
6.2.2 Contoh: Penggunaan “N/A” yang Konsisten dalam Model .....	104
6.2.3 Contoh: Aplikasi Fungsi INFO dan CELL: Tinjauan Umum .....	106
<b>BAB 7 FUNGSI TANGGAL DAN WAKTU .....</b>	<b>110</b>
7.1 Pendahuluan.....	110
7.2 Aplikasi Praktis .....	111
7.2.1 Contoh: Durasi Tugas, Estimasi Sumber Daya dan Biaya .....	111
7.2.2 Contoh: Membuat Sumbu Waktu yang Tepat .....	112
<b>BAB 8 FUNGSI DAN FUNGSIONALITAS TEKS .....</b>	<b>117</b>
8.1 Pendahuluan.....	117
8.2 Aplikasi Praktis .....	118
8.2.1 Contoh: Menggabungkan Teks Menggunakan CONCAT dan TEXTJOIN .....	118
8.2.2 Contoh: Memisahkan Data Menggunakan Panduan Text-to-columns .....	119
8.2.3 Contoh: Membandingkan LEFT, RIGHT, MID, dan LEN .....	122
8.2.4 Contoh: Fungsi UPPER dan LOWER .....	123
8.2.5 Contoh: Fungsi EXACT .....	124
<b>BAB 9 FUNGSI PENCARIAN DAN REFERENSI .....</b>	<b>129</b>
9.1 Pendahuluan .....	129
9.2 Aplikasi Praktis: Proses Referensi Dasar .....	130
9.3 Aplikasi Praktis: Proses Referensi Lebih Lanjut .....	132
9.4 Aplikasi Praktis: Menggabungkan Proses Pencocokan Dan Referensi .....	139
9.5 Aplikasi Praktis: Informasi Tentang Fungsi Offset Dan Rentang Dinamis .....	149
9.6 Aplikasi Praktis: Fungsi Indirect Dan Struktur Buku Kerja Atau Data Fleksibel ..	153

9.7	Contoh Praktis: Penggunaan Hyperlink Untuk Navigasi Model Dan Link Data .	156
<b>BAB 10</b>	<b>FILTER, FUNGSI BASIS DATA, DAN PIVOTTABLE</b>	<b>158</b>
10.1	Pendahuluan.....	158
10.2	Masalah Umum Yang Berlaku Dalam Bekerja Dengan Set Data .....	158
10.3	Basis Data Dan Tabel Excel .....	160
10.4	Aplikasi Praktis: Filter .....	163
10.5	Aplikasi Praktis: Fungsi Basis Data .....	173
10.6	Aplikasi Praktis: Pivottable .....	179
<b>BAB 11</b>	<b>PINTASAN TERPILIH DAN FITUR LAINNYA</b>	<b>190</b>
11.1	Pendahuluan .....	190
11.2	Pintasan Utama Dan Penggunaannya .....	190
11.3	Pemformatan .....	193
11.4	Keytips Excel .....	197
11.5	Alat Dan Fitur Excel Lain Yang Berguna .....	197
<b>BAB 12</b>	<b>MEMULAI DENGAN VBA</b>	<b>199</b>
12.1	Pendahuluan .....	199
12.2	Otomatisasi Tugas .....	199
12.3	Operasi Utama .....	201
12.4	Adaptasi Umum Yang Diperlukan Saat Menggunakan Kode Yang Direkam .....	204
12.5	Contoh Sederhana .....	209
<b>BAB 13</b>	<b>BEKERJA DENGAN OBJEK DAN RENTANG</b>	<b>215</b>
13.1	Tinjauan Umum Model Objek .....	215
13.2	Bekerja Dengan Objek Range: Beberapa Elemen Utama .....	219
13.3	Inputbox Dan MsgBox .....	224
<b>BAB 14</b>	<b>MENGENDALIKAN EKSEKUSI</b>	<b>228</b>
14.1	Kotak Input Dan Kotak Pesan .....	228
14.2	Goto .....	231
14.3	Mengakses Fungsi Worksheet Excel .....	237
14.4	Aplikasi Praktis .....	239
<b>Daftar Pustaka</b>	.....	<b>244</b>



# BAB 1

## FUNGSI ARITMATIKA DAN LOGIKA INTI

### 1.1 PENDAHULUAN

Bab ini membahas serangkaian fungsi inti yang diperlukan dalam berbagai aplikasi pemodelan keuangan, termasuk estimasi pendapatan dan biaya, analisis statistik dan data dasar, peramalan umum, pemodelan laporan keuangan terpadu, penilaian arus kas, pemodelan keuangan proyek, dan sebagainya. Fungsi-fungsi tersebut diambil dari berbagai kategori Excel, dan meliputi:

- *IF, AND, OR, dan NOT.*
- *MIN, MAX, MINA, MAXA, MINIFS, dan MAXIFS.*
- *COUNT, COUNTA, COUNTBLANK, COUNTIF, dan COUNTIFS.*
- *SUM, AVERAGE, AVERAGEA, SUMIF, SUMIFS, AVERAGEIF, dan AVERAGEIFS.*
- *PRODUCT dan SUMPRODUCT.*
- *AGGREGATE dan SUBTOTAL.*
- *IFERROR.*
- *SWITCH.*

Sebagian besar fungsi ini kemungkinan besar sudah dikenal oleh banyak pembaca, dan pada dasarnya sudah jelas. Meskipun demikian, beberapa fitur khusus perlu diperhatikan, termasuk perilakunya dalam kasus khusus (seperti cara menangani bidang non-numerik (teks) atau kosong, atau nilai kesalahan), serta beberapa masalah yang terkait dengan pilihan fungsi mana yang akan digunakan ketika beberapa fungsi tersedia.

### 1.2 APLIKASI PRAKTIS

Bagian ini akan membahas penerapan praktis berbagai fungsi Excel yang telah dijelaskan sebelumnya dalam konteks pemodelan data. Sebagai ilustrasi, Gambar 1.1 menampilkan kumpulan data yang berfokus pada jumlah kunjungan pasien tertentu ke berbagai dokter.

Data ini akan digunakan dalam beberapa contoh untuk menunjukkan bagaimana fungsi-fungsi Excel dapat diterapkan secara langsung dalam analisis dan pengolahan data medis. Dalam contoh-contoh berikut, pembaca akan diperkenalkan pada cara menggunakan fungsi-fungsi seperti SUM, COUNTIF, dan lainnya untuk menarik kesimpulan yang relevan dan menghasilkan analisis yang berguna dalam konteks pemodelan data kesehatan.

Dengan menggunakan data yang tercantum pada Gambar 1.1, kita akan menggali berbagai cara untuk menganalisis pola kunjungan pasien ke dokter. Data ini terdiri dari informasi penting yang akan membantu kita mengidentifikasi tren, frekuensi, serta perbandingan jumlah kunjungan antara dokter yang berbeda. Melalui penerapan fungsi-fungsi Excel yang telah dijelaskan sebelumnya, kita akan mengilustrasikan bagaimana cara

menghitung total kunjungan pasien, mencari dokter dengan kunjungan terbanyak, dan menganalisis distribusi kunjungan berdasarkan kategori tertentu, seperti waktu atau jenis layanan.

	A	B	C	D	E
1					
2					
3		<b>Family Name</b>	<b>First Name</b>	<b>Primary Physician</b>	<b>Annual Visits 2016</b>
4	1	Smith	Amelia	Cooper	3
5	2	Jones	Olivia	Clarke	2
6	3	Taylor	Emily	Patel	1
7	4	Williams	Ava	Mitchell	4
8	5	Brown	Isla	Clarke	7
9	6	Davies	Jessica	James	2
10	7	Evans	Poppy	Cooper	4
11	8	Wilson	Isabella	Clarke	8
12	9	Thomas	Sophie	James	
13	10	Roberts	Mia	Cooper	12
14	11	Johnson	Ruby	Clarke	0
15	12	Lewis	Lily	James	4
16	13	Walker	Grace	Cooper	3
17	14	Robinson	Evie	Clarke	6
18	15	Wood	Sophia	Patel	3
19	16	Thompson	Ella	Patel	7
20	17	White	Scarlett	Mitchell	4
21	18	Watson	Chloe	Clarke	2
22	19	Jackson	Isabelle	Cooper	8
23	20	Wright	Freya	Cooper	4

**Gambar 1.1** Kumpulan Data Yang Digunakan Sebagai Dasar Untuk Beberapa Contoh Di Bagian Awal (Bab 1, Jilid I)

### 1.2.1 Contoh: IF, AND, OR, NOT

Pada prinsipnya, penggunaan fungsi seperti IF, AND, OR, dan NOT mudah. Namun, beberapa hal spesifik perlu diperhatikan:

- Bentuk pendek. Banyak fungsi logika dapat digunakan dalam bentuk pendeknya, seperti `=AND(E19=1, E20=1)` atau `=F7>F6`. Fungsi ini secara implisit melibatkan pernyataan IF, yang dievaluasi menjadi TRUE atau FALSE. Fungsi ini bukan string teks, tetapi ditafsirkan sebagai satu atau nol masing-masing saat digunakan dalam sebagian besar perhitungan berikutnya di Excel. Misalnya, `=50*(F7>F6)` akan menghasilkan 50 atau nol. Bentuk panjang yang setara secara eksplisit melibatkan pernyataan IF, seperti `=IF(F7>F6,1,0)`. (Perhatikan juga, bahwa ini berbeda dengan `=IF(F7>F6,"TRUE","FALSE")`, karena ini mengembalikan string teks, dan karenanya umumnya tidak praktis ketika hasil ekspresi tersebut akan digunakan dalam kalkulasi numerik lebih lanjut.) Namun, hasil uji logika bentuk pendek (mis. FALSE atau

TRUE) dapat bertindak dengan cara yang tidak terduga (mis. tidak secara langsung setara dengan 0 atau 1) ketika disematkan dalam fungsi Excel lainnya. Oleh karena itu, preferensi penulis adalah menggunakan bentuk logika lengkap, yang menghasilkan serangkaian nilai keluaran yang eksplisit, lebih transparan, dan lebih kuat.

- Uji untuk kekosongan. Pernyataan `=IF(G3,1,0)` tidak boleh digunakan untuk menguji apakah sel kosong atau tidak. Ini akan mengembalikan 1 jika G3 berisi nilai positif (atau negatif), 0 jika G3 sama dengan 0 atau kosong, dan `#VALUE` dalam kasus entri teks. Selain itu, tidak dijelaskan secara eksplisit properti G3 mana yang diuji oleh pernyataan tersebut. Fungsi dengan pengujian logika yang lebih eksplisit dan jelas (seperti `ISBLANK`, `ISNUMBER` atau `ISTEXT`, mungkin dikombinasikan dengan `AND`, `OR` atau `NOT`) lebih disukai.

Contoh beberapa fungsi ini, dengan cuplikan layar yang ditunjukkan pada Gambar 1.2 dan Gambar 1.3. Kita akan melihat penerapan langsung beberapa fungsi Excel yang telah dibahas sebelumnya, dengan menggunakan data yang ditunjukkan pada Gambar 1.2 dan Gambar 1.3. Melalui cuplikan layar ini, pembaca dapat mempelajari bagaimana fungsi-fungsi tertentu seperti *SUM*, *COUNTIF*, *AVERAGE*, dan lainnya digunakan untuk menganalisis kumpulan data kunjungan pasien.

#### 1. Fungsi *SUM* dan *COUNTIF*:

Dalam Gambar 1.2, kita akan melihat bagaimana fungsi *SUM* digunakan untuk menghitung total kunjungan pasien ke setiap dokter, sementara fungsi *COUNTIF* akan digunakan untuk menghitung jumlah kunjungan pasien yang memenuhi kriteria tertentu, seperti kunjungan lebih dari tiga kali ke dokter yang sama. Fungsi-fungsi ini memungkinkan kita untuk mendapatkan gambaran umum tentang distribusi kunjungan pasien.

#### 2. Fungsi *AVERAGE* dan *AVERAGEIF*:

Gambar 1.3 menunjukkan penerapan fungsi *AVERAGE* untuk menghitung rata-rata jumlah kunjungan pasien ke seluruh dokter, dan *AVERAGEIF* untuk menghitung rata-rata kunjungan pasien hanya untuk dokter-dokter dengan lebih dari jumlah kunjungan tertentu. Fungsi *AVERAGEIF* memungkinkan kita untuk memfokuskan analisis pada subset data yang memenuhi kriteria khusus.

Melalui cuplikan layar ini, pembaca akan memperoleh pemahaman lebih mendalam tentang bagaimana setiap fungsi bekerja dan bagaimana mereka dapat diterapkan dalam konteks praktis untuk analisis data kunjungan pasien. Contoh-contoh ini juga menunjukkan pentingnya memilih fungsi yang tepat untuk jenis analisis yang ingin dilakukan, serta cara mendapatkan informasi yang relevan dan berguna dari data yang ada.

	D	E	F	G	H
			6	Cooper	
Primary Physician	Annual Visits 2016	Threshold passed?	Physician Match	Both conditions ?	
Cooper	3	=IF(E4>F\$1,1,0)	=IF(D4=G\$1,1,0)	=IF(AND(F4=1,G4=1),1,0)	
Clarke	2	=IF(E5>F\$1,1,0)	=IF(D5=G\$1,1,0)	=IF(AND(F5=1,G5=1),1,0)	
Patel	1	=IF(E6>F\$1,1,0)	=IF(D6=G\$1,1,0)	=IF(AND(F6=1,G6=1),1,0)	
Mitchell	4	=IF(E7>F\$1,1,0)	=IF(D7=G\$1,1,0)	=IF(AND(F7=1,G7=1),1,0)	
Clarke	7	=IF(E8>F\$1,1,0)	=IF(D8=G\$1,1,0)	=IF(AND(F8=1,G8=1),1,0)	
James	2	=IF(E9>F\$1,1,0)	=IF(D9=G\$1,1,0)	=IF(AND(F9=1,G9=1),1,0)	
Cooper	4	=IF(E10>F\$1,1,0)	=IF(D10=G\$1,1,0)	=IF(AND(F10=1,G10=1),1,0)	
Clarke	8	=IF(E11>F\$1,1,0)	=IF(D11=G\$1,1,0)	=IF(AND(F11=1,G11=1),1,0)	
James		=IF(E12>F\$1,1,0)	=IF(D12=G\$1,1,0)	=IF(AND(F12=1,G12=1),1,0)	
Cooper	12	=IF(E13>F\$1,1,0)	=IF(D13=G\$1,1,0)	=IF(AND(F13=1,G13=1),1,0)	
Clarke	0	=IF(E14>F\$1,1,0)	=IF(D14=G\$1,1,0)	=IF(AND(F14=1,G14=1),1,0)	
James	4	=IF(E15>F\$1,1,0)	=IF(D15=G\$1,1,0)	=IF(AND(F15=1,G15=1),1,0)	
Cooper	3	=IF(E16>F\$1,1,0)	=IF(D16=G\$1,1,0)	=IF(AND(F16=1,G16=1),1,0)	
Clarke	6	=IF(E17>F\$1,1,0)	=IF(D17=G\$1,1,0)	=IF(AND(F17=1,G17=1),1,0)	
Patel	3	=IF(E18>F\$1,1,0)	=IF(D18=G\$1,1,0)	=IF(AND(F18=1,G18=1),1,0)	
Patel	7	=IF(E19>F\$1,1,0)	=IF(D19=G\$1,1,0)	=IF(AND(F19=1,G19=1),1,0)	
Mitchell	4	=IF(E20>F\$1,1,0)	=IF(D20=G\$1,1,0)	=IF(AND(F20=1,G20=1),1,0)	
Clarke	2	=IF(E21>F\$1,1,0)	=IF(D21=G\$1,1,0)	=IF(AND(F21=1,G21=1),1,0)	
Cooper	8	=IF(E22>F\$1,1,0)	=IF(D22=G\$1,1,0)	=IF(AND(F22=1,G22=1),1,0)	
Cooper	4	=IF(E23>F\$1,1,0)	=IF(D23=G\$1,1,0)	=IF(AND(F23=1,G23=1),1,0)	

Gambar 1.2 Contoh Fungsi If Dan And Menggunakan Tampilan Rumus

	D	E	F	G	H
			6	Cooper	
Primary Physician	Annual Visits 2016	Threshold passed?	Physician Match	Both conditions ?	
Cooper	3	0	1	0	
Clarke	2	0	0	0	
Patel	1	0	0	0	
Mitchell	4	0	0	0	
Clarke	7	1	0	0	
James	2	0	0	0	
Cooper	4	0	1	0	
Clarke	8	1	0	0	
James		0	0	0	
Cooper	12	1	1	1	
Clarke	0	0	0	0	
James	4	0	0	0	
Cooper	3	0	1	0	
Clarke	6	0	0	0	
Patel	3	0	0	0	
Patel	7	1	0	0	
Mitchell	4	0	0	0	
Clarke	2	0	0	0	
Cooper	8	1	1	1	
Cooper	4	0	1	0	

Gambar 1.3 Contoh Fungsi If Dan And Yang Menampilkan Hasil

### 1.2.2 Contoh: MIN, MAX, MINA, MAXA

Item	Result	Formula
Maximum number of visits	12	=MAX(E4:E23)
Minimum number of visits	0	=MIN(E4:E23)
Minimum number of visits including blank cells, but with no zeros in the range	1	=MIN(D4:E12)
Minimum number of visits including blank cells, but with no zeros in the range	0	=MINA(D4:E12)

**Gambar 1.4 Contoh Fungsi Min Dan Max**

Gambar 1.4 menunjukkan File berisi contoh fungsi MAX dan MIN untuk menghitung nilai maksimum dan minimum dalam satu set data, dengan cuplikan layar yang ditunjukkan pada Gambar 1.4. File ini juga menunjukkan penggunaan fungsi MAXA dan MINA. Perlu dicatat bahwa:

- Fungsi MIN dan MAX mengabaikan sel dan teks kosong. Secara khusus, sel kosong tidak diperlakukan seolah-olah berisi nol. Misalnya, nilai minimum (MIN) dari satu set item yang berisi nilai kosong dan nilai positif (bukan nol) akan menjadi angka positif terkecil (bukan nol).
- Fungsi MINA dan MAXA tidak mengabaikan nilai kosong, sehingga MINA dari satu set item yang berisi nilai kosong dan nilai positif (bukan nol) akan menjadi nol. Bidang teks tidak diabaikan oleh fungsi, tetapi keberadaannya tidak berpengaruh pada keluaran yang dikembalikan (ini berbeda dengan COUNTA, di mana keberadaan bidang teks akan mengubah hasil).

### 1.2.3 Contoh: MINIFS dan MAXIFS

Gambar 1.5 menunjukkan File berisi contoh fungsi MAXIFS untuk melakukan kueri bersyarat. Gambar 1.5 menunjukkan jumlah kunjungan maksimum untuk beberapa dokter.

Item	Result	Formula
Maximum number of visits		
Cooper	12	=MAXIFS(E54:E523,D54:D523,G5)
Clarke	8	=MAXIFS(E54:E523,D54:D523,G6)
Patel	7	=MAXIFS(E54:E523,D54:D523,G7)
Mitchell	4	=MAXIFS(E54:E523,D54:D523,G8)

**Gambar 1.5 Contoh Fungsi Maxifs**

Perhatikan bahwa fungsi MINIFS dan MAXIFS diperkenalkan ke Excel pada awal tahun 2016; sebelum ini, fungsionalitas yang paling mirip dapat dicapai baik dengan menggunakan fungsi Basis Data (DMIN dan DMAX), rumus array, atau bentuk array dari fungsi AGGREGATE, yang semuanya dibahas nanti dalam teks ini.

### 1.2.4 Contoh: COUNT, COUNTA, COUNTIF dan Fungsi Serupa

Item	Result	Formula
Number of non-blank visit entries	19	=COUNT(E4:E23)
Number of non-blank visit entries	19	=COUNTA(E4:E23)
Number of blank visit entries	1	=COUNTBLANK(E4:E23)
Number of patients, using COUNT (incorrect for patient numbers using list of names)	0	=COUNT(B4:B23)
Number of patients, using COUNTA	20	=COUNTA(B4:B23)
Number of non-blank cells in range B11:E11	4	=COUNTA(B11:E11)
Number of non-blank cells in range B12:E12	3	=COUNTA(B12:E12)
Number of non-blank cells in range B13:E13 that contain only numbers	1	=COUNT(B13:E13)
Number of patients with blank entry	1	=COUNTIF(E4:E23,"")
Number of patients with 0 entry	1	=COUNTIF(E4:E23,0)
Number of patients with 2 visits (taken from cell G23)	3	=COUNTIF(E4:E23,G24)
Number of patients who actually visited, excluding blanks and those with 0 entry	18	=COUNTIF(E4:E23,">0")
Number of patients who visited between 2 and 4 time inclusive (blanks and those with 0 entry would not count)	11	=COUNTIFS(E4:E23,">=2",E4:E23,"<=4")
Using cell references (trial 1): Number of patients who visited between 2 and 4 times inclusive	0	=COUNTIFS(E4:E23,">=G24",E4:E23,"<=G25")
Using cell references (trial 2): Number of patients who visited between 2 and 4 times inclusive	11	=COUNTIFS(E4:E23,">="&G24,E4:E23,"<="&G25)

**Gambar 1.6 Contoh Fungsi Count, Counta, Countif, Dan Fungsi Serupa**

Gambar 1.6 menunjukkan File berisi contoh fungsi tipe COUNT termasuk COUNT, COUNTA, COUNTBLANK, COUNTIF dan COUNTIFS (lihat Gambar 1.6).

Beberapa poin yang perlu diperhatikan:

- COUNT menghitung angka dalam suatu rentang, mengabaikan sel kosong, yaitu sel kosong tidak diperlakukan seolah-olah berisi nol, tetapi seolah-olah tidak ada sama sekali (yaitu seolah-olah tidak dimasukkan dalam set sel input).
- COUNTA menghitung teks dan angka dalam suatu rentang, mengabaikan sel kosong.
- COUNTBLANK dapat digunakan untuk menghitung jumlah sel kosong dalam suatu rentang, seperti halnya COUNTIF.
- COUNTIF dapat digunakan untuk melakukan perhitungan bersyarat, dan beberapa kasus spesifik perlu diperhatikan:
- Penerapan kondisi kesetaraan (seperti menghitung jumlah item yang sama persis dengan angka tertentu) dapat dicapai dengan penggunaan langsung nilai yang relevan atau referensi sel dalam argumen fungsi yang sesuai.
- Kondisi ketidaksetaraan memerlukan entri simbol yang sesuai (seperti >, <, >=, atau <=), yang terdapat dalam tanda kutip, seperti ">=2".
- Ketika kondisi ketidaksetaraan harus merujuk ke sel (alih-alih menggunakan nilai yang dikodekan secara kaku), pengeditan manual fungsi diperlukan. Misalnya, kriteria yang diinginkan untuk merujuk ke sel seperti ">=G24" akan memerlukan pengeditan untuk membuat ">="&G24.
- Kondisi "Between" dapat dibuat dengan menggunakan fungsi COUNTIFS dan menerapkan dua set kondisi ke rentang acuan yang sama (seperti memeriksa ">=" dan "<=" dalam rentang yang sama, menggunakan dua bidang kriteria fungsi).
- Jika ada satu kondisi, pada dasarnya tidak relevan apakah seseorang menggunakan

COUNTIF atau COUNTIFS. Fungsi yang terakhir berfungsi jika hanya ada satu kriteria, dan penggunaan yang pertama mudah diadaptasi jika kriteria tambahan diperlukan (dengan menambahkan "S" ke fungsi).

### 1.2.5 Contoh: SUM, AVERAGE, AVERAGEA

Item	Result	Formula
Number of annual visits (total)	84	=SUM(E4:E23)
Sum, including text within the range	8	=SUM(D11:E11)
Average of two cells: a zero and another number	6	=AVERAGE(E13:E14)
Average of a cell containing a number and blank cell	8	=AVERAGE(E11:E12)
Average of only numerical fields	8	=AVERAGE(D11:E11)
Average number of annual visits, including those with 0 entry	4.42	=AVERAGE(E4:E23)
Average number of annual visits, including those with 0 entry	4.42	=SUM(E4:E23)/COUNT(E4:E23)
Average of all fields	4	=AVERAGEA(D11:E11)

**Gambar 1.7 Contoh Fungsi SUM Dan AVERAGE**

Gambar 1.7 menunjukkan contoh fungsi SUM dan AVERAGE, dan juga perhitungan eksplisit rata-rata menggunakan COUNT; cuplikan layar ditunjukkan pada Gambar 1.7. File ini juga menunjukkan contoh AVERAGEA.

Beberapa poin perlu diperhatikan secara khusus:

- Fungsi SUM akan memperlakukan sel yang kosong atau berisi teks sebagai nol (atau mengabaikan, yang sama untuk fungsi ini).
- Fungsi AVERAGE mengabaikan sel yang kosong atau berisi teks (sehingga sel tersebut tidak termasuk dalam jumlah sel dalam penyebut perhitungan); fungsi ini memberikan hasil yang sama seperti menggunakan SUM dibagi dengan COUNT.
- Fungsi AVERAGEA memperlakukan teks seolah-olah bernilai nol, sehingga teks tersebut dimasukkan dalam hitungan (untuk penyebut) tanpa memengaruhi total (pembilang), dan karenanya berpotensi memberikan jawaban yang berbeda untuk AVERAGE. Misalnya, saat diterapkan pada rentang yang hanya berisi angka positif dan beberapa teks, AVERAGEA akan mengembalikan nilai yang lebih rendah daripada AVERAGE.

### 1.2.6 Contoh: SUMIF, SUMIFS, AVERAGEIF, AVERAGEIFS

Gambar 1.8 menunjukkan contoh fungsi SUMIF, SUMIFS, AVERAGEIF, dan AVERAGEIFS, dengan cuplikan layar yang ditunjukkan pada Gambar 17.8. Perlu dicatat bahwa:

- Fungsi COUNTIF dapat dengan mudah diubah menjadi fungsi COUNTIFS hanya dengan menyertakan kriteria tambahan. Namun, proses yang setara tidak mungkin dilakukan jika fungsi SUMIF atau AVERAGEIF dicoba diadaptasi menjadi SUMIFS dan AVERAGEIFS; pada fungsi-fungsi yang terakhir, rentang yang akan dijumlahkan atau dirata-ratakan harus menjadi argumen (parameter) pertama, sedangkan rentang tersebut harus menjadi parameter terakhir (dan opsional) dari fungsi SUMIF dan AVERAGEIF.

- Pada versi Excel sebelum Excel 2013, rumus multikriteria dapat dibuat menggunakan Conditional Sum Wizard (utilitas langkah demi langkah); rumus yang dibuat dengan cara ini adalah rumus array yang masih kompatibel dengan versi Excel yang lebih baru, tetapi utilitas itu sendiri sekarang pada dasarnya berlebihan karena fungsi tipe "IFS" dapat dengan mudah digunakan sebagai gantinya.

	G	H	I
Item		Result	Formula
Total number of visits to physician Cooper using SUMIF:			
Cooper		34	=SUMIF(\$D\$4:\$D\$23,G6,\$E\$4:\$E\$23)
Total number of visits to physician Cooper using SUMIFS with one criteria:			
Cooper		34	=SUMIFS(\$E\$4:\$E\$23,\$D\$4:\$D\$23,G9)
Average number of visits to physician Cooper using AVERAGEIF:			
Cooper		5.67	=AVERAGEIF(\$D\$4:\$D\$23,G12,\$E\$4:\$E\$23)
Average number of visits to physician Cooper using AVERAGEIFS with one criteria:			
Cooper		5.67	=AVERAGEIFS(\$E\$4:\$E\$23,\$D\$4:\$D\$23,G12)

**Gambar 1.8 Contoh Fungsi SUMIF, SUMIFS, AVERAGEIF, dan AVERAGEIFS**

#### Contoh: PRODUCT

Gambar 1.8 menunjukkan beberapa contoh perhitungan menggunakan fungsi PRODUCT. Fungsi ini mengalikan argumennya, yang dapat berupa satu rentang sel yang bersebelahan. Tentu saja, jika hanya dua sel yang akan dikalikan, seseorang dapat menggunakan operasi perkalian standar. Dengan demikian, fungsi ini paling berguna jika item dalam rentang sel yang bersebelahan harus dikalikan.

Dalam kasus seperti itu, penggunaan fungsi ini menghindari keharusan membuat rumus dengan banyak operator perkalian (seperti =E5\*E6\*E7\*E8). Selain rumit untuk dibuat, rumus seperti itu rawan kesalahan jika seseorang mungkin perlu memasukkan sel nanti. (Hal ini sama dengan menggunakan fungsi SUM sebagai pengganti penggunaan berulang operator penjumlahan yang akan merujuk ke sel-sel individual.)

Salah satu penerapannya adalah pada kalkulasi probabilitas: jika beberapa kejadian independen dapat terjadi dengan probabilitas tertentu, probabilitas bahwa semuanya terjadi adalah hasil perkalian dari probabilitas individual. Demikian pula, probabilitas bahwa tidak ada satu pun kejadian yang terjadi adalah hasil perkalian dari probabilitas bahwa masing-masing kejadian tidak terjadi.



	A	B	C	D	E
1					
2			Probability(occur)	Probability(not occur)	Formulae
3		Event number 1	60%	40%	=1-C3
4		Event number 2	80%	20%	=1-C4
5		Event number 3	60%	40%	=1-C5
6		Event number 4	50%	50%	=1-C6
7		Event number 5	90%	10%	=1-C7
8		P(all)	13.0%	0.2%	=PRODUCT(D3:D7)
9			=PRODUCT(C3:C7)	=PRODUCT(D3:D7)	

**Gambar 1.9 Menggunakan PRODUCT Untuk Menghitung Probabilitas Kemunculan Dan Ketidakmunculan**

Gambar 1.9 menunjukkan sebuah contoh. Dengan menggunakan kalkulasi serupa, seseorang dapat menunjukkan bahwa dalam kelompok yang terdiri dari 23 orang (atau lebih), peluang bahwa dua orang memiliki hari ulang tahun yang sama hanya lebih dari 50%. Dengan kata lain, agar ada peluang lebih dari 50% bahwa dua orang dalam kelompok memiliki hari ulang tahun yang sama, ukuran kelompok harus 23 orang atau lebih (lihat Gambar 1.10).

F	G	H	I	J	K
	Person Number	No. days to choose from	Prob. Different date-No. days/365	Cumulative Probability	Formulae
	1	365	100.0%	100.0%	=PRODUCT(\$I3:I3)
	2	364	99.7%	99.7%	=PRODUCT(\$I3:I4)
	3	363	99.5%	99.2%	=PRODUCT(\$I3:I5)
	4	362	99.2%	98.4%	=PRODUCT(\$I3:I6)
	5	361	98.9%	97.3%	=PRODUCT(\$I3:I7)
	6	360	98.6%	96.0%	=PRODUCT(\$I3:I8)
	7	359	98.4%	94.4%	=PRODUCT(\$I3:I9)
	8	358	98.1%	92.6%	=PRODUCT(\$I3:I10)
	9	357	97.8%	90.5%	=PRODUCT(\$I3:I11)
	10	356	97.5%	88.3%	=PRODUCT(\$I3:I12)
	11	355	97.3%	85.9%	=PRODUCT(\$I3:I13)
	12	354	97.0%	83.3%	=PRODUCT(\$I3:I14)
	13	353	96.7%	80.6%	=PRODUCT(\$I3:I15)
	14	352	96.4%	77.7%	=PRODUCT(\$I3:I16)
	15	351	96.2%	74.7%	=PRODUCT(\$I3:I17)
	16	350	95.9%	71.6%	=PRODUCT(\$I3:I18)
	17	349	95.6%	68.5%	=PRODUCT(\$I3:I19)
	18	348	95.3%	65.3%	=PRODUCT(\$I3:I20)
	19	347	95.1%	62.1%	=PRODUCT(\$I3:I21)
	20	346	94.8%	58.9%	=PRODUCT(\$I3:I22)
	21	345	94.5%	55.6%	=PRODUCT(\$I3:I23)
	22	344	94.2%	52.4%	=PRODUCT(\$I3:I24)
	23	343	94.0%	49.3%	=PRODUCT(\$I3:I25)

**Gambar 1.10 Menggunakan PRODUCT Untuk Menghitung Probabilitas Ulang Tahun Yang Sama**

Contoh lain dari fungsi PRODUCT adalah menghitung nilai akhir dari sesuatu yang tumbuh dengan laju  $g_1$  pada periode 1,  $g_2$  pada periode 2, dan seterusnya. Nilai pada akhir periode N adalah:

$$V_N = V_0(1 + g_1)(1 + g_2)\dots(1 + g_N)$$

Pada Gambar 1.11, kami menunjukkan perhitungan ini, dengan asumsi bahwa bidang yang berisi item 1+g pertama kali dihitung di Excel (Kolom O).

L	M	N	O	P
	<b>Period Number</b>	<b>Growth Rates</b>	<b>Value &amp; adjustment</b>	<b>Formulae</b>
	0		100	
	1	2.0%	102.0%	=1+N4
	2	3.0%	103.0%	=1+N5
	3	5.0%	105.0%	=1+N6
	4	8.0%	108.0%	=1+N7
	5	7.0%	107.0%	=1+N8
	6	6.0%	106.0%	=1+N9
	7	5.0%	105.0%	=1+N10
	8	4.0%	104.0%	=1+N11
	9	2.0%	102.0%	=1+N12
	10	2.0%	102.0%	=1+N13
			153.5	=O\$3*PRODUCT(O\$4:O13)

Gambar 1.11 Menggunakan Product Untuk Menghitung Nilai Masa Depan

A	B	C	D	E
1				
2	<b>Asset Number</b>	<b>% in Each Asset</b>	<b>Return (expected)</b>	
3	1	40.0%	10%	
4	2	25.0%	15%	
5	3	20.0%	18%	
6	4	15.0%	20%	
7	<b>Total</b>	<b>100%</b>	<b>14.4%</b>	
8		=SUM(C3:C6)	=SUMPRODUCT(C3:C6,D3:D6)	

Gambar 1.12 Pengembalian Portofolio Menggunakan Sumproduct

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
1																
2	<b>Years</b>		2015	2016	2017	2018	2019	2020								
3	<b>Capex</b>		100	105	120	80	50	100								
4																
5	<b>Generic Periods</b>		1	2	3	4	5									
6	<b>Depreciation Schedule</b>	100%	30%	25%	20%	15%	10%									
7																
8	<b>Depreciation</b>		2015	2016	2017	2018	2019	2020	2021	2022	2023	2024				
9		2015	100	30%	25%	20%	15%	10%								
10		2016	105	30%	25%	20%	15%	10%								
11		2017	120		30%	25%	20%	15%	10%							
12		2018	80			30%	25%	20%	15%	10%						
13		2019	50				30%	25%	20%	15%	10%					
14		2020	100					30%	25%	20%	15%	10%				
15	<b>Total</b>		30.0	56.5	82.3	=SUMPRD	84.8	87.0	59.0	35.5	20.0	10.0	=SUMPRODUCT(\$C9:\$C14,M9:M14)			
16																

Gambar 1.13 Perhitungan Depresiasi Menggunakan Sumproduct

#### Contoh: SUMPRODUCT

Fungsi SUMPRODUCT bekerja dengan serangkaian rentang baris atau kolom dengan ukuran yang sama (dan yang semuanya berada dalam kolom atau semuanya berada dalam

baris), dan mengalikan elemen pertama di setiap rentang, elemen kedua di setiap rentang, dan seterusnya, lalu menjumlahkannya. Meskipun dapat diterapkan saat ada beberapa rentang, penggunaan yang paling sering adalah saat hanya ada dua rentang.

(Jika seseorang ingin mengalikan data baris dengan data kolom, seseorang dapat menggunakan fungsi array, seperti TRANSPOSE atau MMULT, seperti yang dibahas dalam Bab 2.) dalam gambar 1.12 menunjukkan contoh dalam analisis portofolio, di mana fungsi tersebut digunakan untuk menghitung pengembalian portofolio berdasarkan rata-rata tertimbang dari masing-masing komponennya (aset) (lihat Gambar 1.12).

Serupa dengan itu, pada gambar 1.13 menunjukkan contoh di mana struktur "tipe segitiga" digunakan untuk menghitung profil penyusutan total ketika profil pengeluaran modal diketahui dan jadwal penyusutan generik diberikan (lihat Gambar 1.13).

#### Contoh: SUBTOTAL

Fungsi SUBTOTAL memungkinkan berbagai perhitungan ringkasan dilakukan pada sekumpulan data yang disusun dalam kolom (bukan baris). Fungsi ini mengabaikan fungsi SUBTOTAL lain dalam rentangnya, sehingga penghitungan ganda (dan referensi melingkar) dapat dihindari. Ini juga memungkinkan seseorang untuk menentukan bagaimana baris tersembunyi akan diperlakukan.

Function	FunctionNumber	
	To include hidden values	To ignore hidden values
AVERAGE	1	101
COUNT	2	102
COUNTA	3	103
MAX	4	104
MIN	5	105
PRODUCT	6	106
STDEV	7	107
STDEVP	8	108
SUM	9	109
VAR	10	110
VARP	11	111

**Gambar 1.14 Angka Fungsi Saat Menggunakan SUBTOTAL**

Selanjutnya dalam gambar 1.14 menunjukkan ringkasan berbagai kemungkinan perhitungan untuk fungsi tersebut, termasuk cara menangani data yang tersembunyi, yang didefinisikan menggunakan parameter pertama fungsi tersebut; hal ini juga ditunjukkan pada Gambar 1.14. Perlu diperhatikan bahwa seseorang perlu ekstra hati-hati saat menggunakan area tersembunyi dan filter Excel secara bersamaan (lihat Bab 10). Fungsi tersebut dapat dimasukkan baik:

- Dengan penyisipan langsung ke dalam sel. Hal ini dapat masuk akal, misalnya dalam model laporan keuangan, di mana total aset dapat dihitung dari (subtotal) aset tetap dan aset lancar, yang masing-masing dapat menjadi subtotal dari perhitungan yang

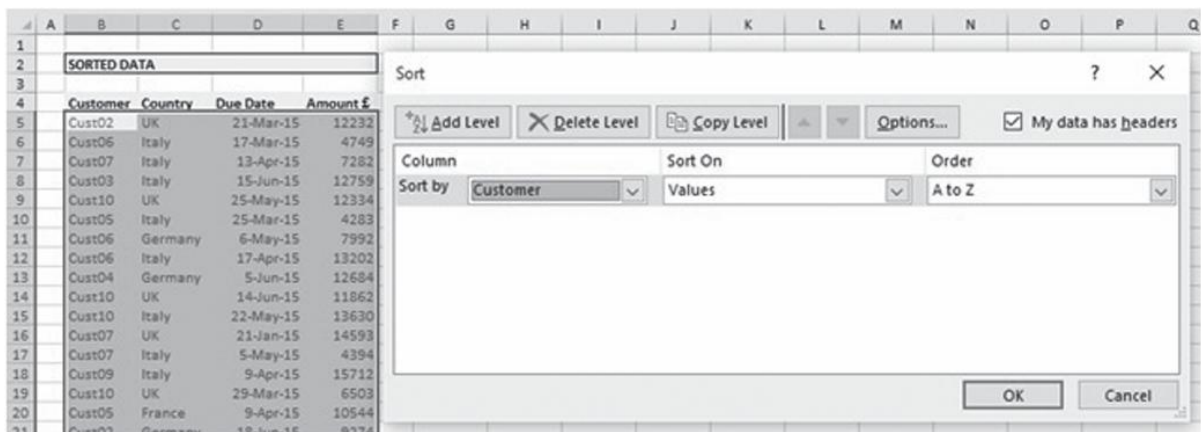
lebih terperinci (seperti peralatan, modal kerja, dll.).

- Menggunakan Panduan Subtotal pada tab Data Excel. Penggunaan Wizard sangatlah ampuh ketika diterapkan pada sekumpulan data yang salah satu kolomnya berisi variabel kategori yang telah diurutkan (misalnya menggunakan Data/Sort) sehingga semua item dalam kategori berada dalam baris yang bersebelahan; dalam kasus ini, Wizard dapat digunakan untuk membuat subtotal menurut kategori.

Gambar 1.15 menunjukkan contoh pendekatan pertama, yang mana fungsi tersebut telah disisipkan secara manual ke dalam beberapa baris, untuk menghitung jumlah total halaman buku, sekaligus menunjukkan jumlah halaman di setiap bagian (lihat Gambar 1.15). File gambar 1.16 menunjukkan contoh pendekatan kedua, yang mana fungsi Wizard digunakan.

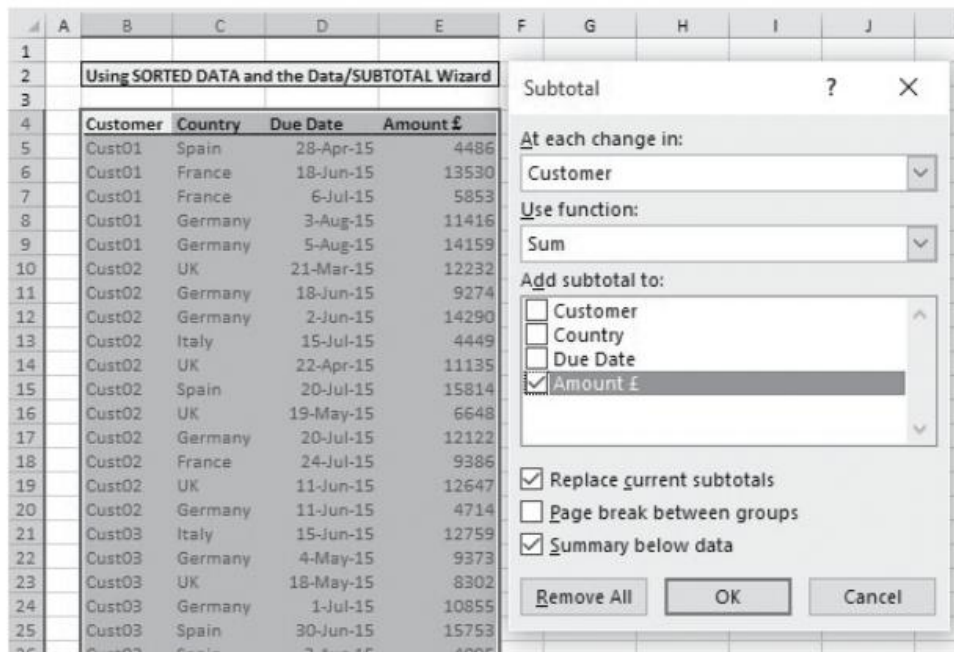
	A	B	C	D
1				
2			Pages	Formulae
3		PART I	80	=SUBTOTAL(9,C4:C6)
4		Chapter 1	25	
5		Chapter 2	20	
6		Chapter 3	35	
7		PART II	100	=SUBTOTAL(9,C8:C12)
8		Chapter 4	15	
9		Chapter 5	20	
10		Chapter 6	30	
11		Chapter 7	20	
12		Chapter 8	15	
13		PART III	55	=SUBTOTAL(9,C14:C15)
14		Chapter 8 ....	25	
15		Chapter 9 ....	30	
16		PART IV	100	=SUBTOTAL(9,C17:C18)
17		Chapter 10 ....	75	
18		Chapter 11 ....	25	
19		Total	=SUBTOTAL(9,C3:C18)	=SUBTOTAL(9,C3:C18)

**Gambar 1.15 Penggunaan Subtotal Dengan Penyisipan Langsung**



**Gambar 1.16 Penyisipan Fungsi Subtotal Menurut Kategori Menggunakan Wizard**

Pada langkah pertama, kumpulan data diurutkan per variabel kategori yang diinginkan yang memerlukan perhitungan subtotal (menggunakan menu Sort pada tab Data); Hal ini ditunjukkan dalam cuplikan layar pada Gambar 1.16, yang di dalamnya dilakukan penyortiran berdasarkan Pelanggan. Pada langkah kedua, Wizard (pada tab Data, dalam kategori Outline) digunakan untuk membuat subtotal jumlah berdasarkan identitas pelanggan (lihat Gambar 1.17).



**Gambar 1.17 Penyisipan Fungsi Subtotal Berdasarkan Kategori Menggunakan Wizard**

Pada Gambar 1.18, kami menunjukkan hasil saat proses Wizard telah selesai.

#### **Contoh: AGGREGATE**

Fungsi AGGREGATE (di Excel 2013 dan seterusnya), seperti SUBTOTAL, dirancang untuk kolom (bukan baris) data. Meskipun dalam bentuk dasarnya, fungsi ini umumnya akan bekerja dengan data yang disusun dalam baris, seseorang mungkin mendapatkan hasil yang tidak diharapkan jika kolom disembunyikan.

Perbandingan utama dengan fungsi SUBTOTAL adalah:

- Fungsi AGGREGATE memiliki lebih banyak opsi tentang bagaimana data akan diperlakukan, termasuk mengabaikan nilai kesalahan. Ini adalah poin penting; sebagian besar fungsi Excel mengharuskan tidak ada kesalahan dalam kumpulan data (misalnya SUM, MAX, SUBTOTAL). Properti ini digunakan dalam Bab 2 untuk menunjukkan bagaimana bentuk array fungsi dapat digunakan untuk mereplikasi kalkulasi bersyarat.
- Ada serangkaian kemungkinan yang sedikit lebih besar untuk fungsi yang mendasarinya.
- Fungsi tersebut tidak terpengaruh oleh penerapan filter di Excel.
- Tidak ada Wizard untuk memfasilitasi penyisipan di berbagai tempat dalam kumpulan data.

	A	B	C	D	E	F	G	
1								
2		Using SORTED DATA and the Data/SUBTOTAL Wizard						
3								
4		<b>Customer</b>	<b>Country</b>	<b>Due Date</b>	<b>Amount £</b>			
5		Cust01	Spain	28-Apr-15	4486			
6		Cust01	France	18-Jun-15	13530			
7		Cust01	France	6-Jul-15	5853			
8		Cust01	Germany	3-Aug-15	11416			
9		Cust01	Germany	5-Aug-15	14159			
10		<b>Cust01 Total</b>			<b>49444</b>			
11		Cust02	UK	21-Mar-15	12232			
12		Cust02	Germany	18-Jun-15	9274			
13		Cust02	Germany	2-Jun-15	14290			
14		Cust02	Italy	15-Jul-15	4449			
15		Cust02	UK	22-Apr-15	11135			
16		Cust02	Spain	20-Jul-15	15814			
17		Cust02	UK	19-May-15	6648			
18		Cust02	Germany	20-Jul-15	12122			
19		Cust02	France	24-Jul-15	9386			
20		Cust02	UK	11-Jun-15	12647			
21		Cust02	Germany	11-Jun-15	4714			
22		<b>Cust02 Total</b>			<b>112711</b>			
23		Cust03	Italy	15-Jun-15	12759			
24		Cust03	Germany	4-May-15	9373			
25		Cust03	UK	18-May-15	8302			
26		Cust03	Germany	1-Jul-15	10855			
27		Cust03	Spain	30-Jun-15	15753			
28		Cust03	Spain	3-Aug-15	4095			
29		Cust03	UK	6-Aug-15	12737			
30		Cust03	UK	22-Jul-15	8814			
31		Cust03	Spain	4-Aug-15	13064			

**Gambar 1.18 Hasil Akhir Setelah Penyisipan Fungsi SUBTOTAL Menggunakan Wizard**

Gambar 1.19 mencantumkan opsi fungsi dan menunjukkan beberapa contoh. Gambar 1.19 menunjukkan kemungkinan untuk berbagai perhitungan yang mungkin ingin dilakukan; perhitungan ini diimplementasikan dengan menggunakan angka yang sesuai sebagai argumen FunctionNumber. Mengenai fungsi SUBTOTAL, meskipun teks pada titik ini hanya membahas beberapa kemungkinan fungsi, penerapannya pada keseluruhan rangkaian fungsi seharusnya jelas pada prinsipnya.

Function	FunctionNumber
AVERAGE	1
COUNT	2
COUNTA	3
MAX	4
MIN	5
PRODUCT	6
STDEV.S	7
STDEV.P	8
SUM	9
VAR.S	10
VAR.P	11
MEDIAN	12
MODE.SNGL	13
LARGE(array,k)	14
SMALL(array,k)	15
PERCENTILE.INC(array,k)	16
QUARTILE.INC(array,quart)	17
PERCENTILE.EXC(array,k)	18
QUARTILE.EXC(array,quart)	19

**Gambar 17.19 Kemungkinan Perhitungan yang Dilakukan oleh Fungsi AGREGATE**

File tersebut juga menunjukkan opsi yang berkaitan dengan cara data masukan diperlakukan dalam perhitungan. Misalnya, opsi nomor 6 menginstruksikan agar nilai kesalahan diabaikan (lihat Gambar 1.20).

Values to ignore	OptionNumber
Ignore nested SUBTOTAL and AGGREGATE functions	0 or omitted
Ignore hidden rows, nested SUBTOTAL and AGGREGATE functions	1
Ignore error values, nested SUBTOTAL and AGGREGATE functions	2
Ignore hidden rows, error values, nested SUBTOTAL and AGGREGATE functions	3
Ignore nothing	4
Ignore hidden rows	5
Ignore error values	6
Ignore hidden rows and error values	7

**Gambar 1.20** Kemungkinan Mengabaikan Data Input saat Fungsi AGREGATE Mengevaluasi

File tersebut juga menyediakan beberapa contoh fungsi. Klip layar dari lembar kerja kedua ditunjukkan pada Gambar 1.21. Perhatikan parameter opsional, k, yang mewakili parameter tambahan, memerlukan fungsi nomor 14 hingga 19.

Array or Ref1	Ref2 for Reference Form
678	400
993	110
5	580
-59	738
#DIV/0!	227
1199	1239
347	1029
#NUM!	804
363	400
416	1770
11	1701

Array Form: (FunctionNumber, Options, Array, [k or quart])					
FunctionNumber	OptionNumber	k or quart	Result	Formula	Comments
4	0	NA	1199	=AGGREGATE(B17, C17, \$B\$1:\$B\$11)	Largest value, ignoring errors
12	0	NA	363	=AGGREGATE(B18, C18, \$B\$1:\$B\$11)	Calculated median, ignoring errors
15	6	2	11	=AGGREGATE(B19, C19, \$B\$1:\$B\$11, D19)	2nd smallest, ignoring errors

Reference Form: (FunctionNumber, Options, Ref1, [Ref2, ...])					
FunctionNumber	OptionNumber	Ref2	Result	Formula	Comments
12	0		534.5	=AGGREGATE(B23, C23, \$B\$1:\$B\$11, \$C\$3:\$C\$11)	Calculated median of all data sets, ignoring errors
12	0	NA	534.5	=AGGREGATE(B24, C24, \$B\$3:\$C\$11)	Calculated median, ignoring errors

**Gambar 1.21** Contoh Penerapan Fungsi AGREGATE

### Contoh: IFERROR

Fungsi IFERROR terdapat dalam kategori Logika Excel. Fungsi ini memungkinkan nilai alternatif untuk dikeluarkan jika rangkaian perhitungan default menghasilkan kesalahan. Gambar 1.22 menunjukkan contoh (lihat Gambar 1.22). Ini didasarkan pada fakta bahwa fungsi PPMT (yang menghitung bagian pokok dari pembayaran tetap yang diperlukan untuk membayar kembali pinjaman selama beberapa periode lihat Bab 7 untuk detailnya) menghasilkan nilai numerik yang valid hanya untuk periode yang berada dalam jangka waktu pinjaman.

Di luar jangka waktu pinjaman berlaku, fungsi tersebut menghasilkan kesalahan

#NUM! (meskipun saat ini pinjaman telah dilunasi). Dalam contoh ini, model telah ditetapkan untuk memiliki 10 periode, meskipun panjang pinjaman hanya tujuh periode.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1														
2		Periodic interest rate	5.0%											
3		Periods to end of term	7											
4		Principal amount (\$)	1,000											
5														
6		Principal by period	Sum	1	2	3	4	5	6	7	8	9	10	Formula
7														
8		Principal repayment using PPMT	#NUM!	123	129	135	142	148	157	165	#NUM!	#NUM!	#NUM!	=PPMT(\$C2,M6,\$C3,\$C4)
9		Principal repayment using PPMT (with IF)	1,000	123	129	135	142	148	157	165	0	0	0	=IF(M6=>\$C3,M8,0)
10		Principal repayment using PPMT (with IFERROR)	1,000	123	129	135	142	148	157	165	0	0	0	=IFERROR(PPMT(\$C2,M6,\$C3,\$C4),0)
11														

**Gambar 1.22 Contoh Fungsi IFERROR**

Oleh karena itu, bagian pokok pembayaran untuk tahun ke-8, ke-9, dan ke-10 (sel K8:M8) dikembalikan sebagai #NUM! Untuk sebagian besar tujuan praktis atau perhitungan, fakta (struktural) bahwa tidak ada pembayaran yang jatuh tempo setelah akhir kontrak pinjaman setara dengan jumlah pembayaran nol. Tentu saja, dalam pengertian teknis atau legalistik, tidak adanya kewajiban mungkin tidak sama dengan memiliki kewajiban sejumlah nol (kasus terakhir ini adalah kasus yang paling mirip dengan perhitungan fungsi).

Oleh karena itu, sering kali paling praktis untuk mempertimbangkan bahwa nilai kesalahan benar-benar nol, terutama jika nilai-nilai ini harus digunakan dalam perhitungan model lebih lanjut atau tampilan grafis. Di Baris 9, ini telah dicapai dengan menggunakan pernyataan IF, yang mengembalikan nilai alternatif untuk pembayaran (yaitu nol) jika jangka waktu melampaui masa berlaku pinjaman.

Perhatikan bahwa fungsi IF tidak mendeteksi secara langsung apakah fungsi PPMT telah mengembalikan kesalahan; melainkan, ia menangani kasus tertentu di mana diketahui sebelumnya bahwa fungsi tersebut akan menghasilkan kesalahan. Sebagai alternatif, seseorang dapat menggunakan fungsi IFERROR secara langsung (Baris 11) untuk sekadar mengganti nilai apa pun dari PPMT yang merupakan kesalahan dengan nol. Perlu dicatat juga bahwa sebelum Excel 2007, fungsi yang setara dengan IFFERROR dapat dicapai dengan menyematkan fungsi ISERROR (Informasi) dalam pernyataan IF.

Meskipun mudah digunakan, seseorang harus berhati-hati dalam penggunaan IFERROR. Secara umum, lebih baik mengadaptasi rumus yang seharusnya berlaku untuk kemunculan spesifik dari "kesalahan yang valid". Jika seseorang mengabaikan semua kesalahan apa pun yang muncul, seseorang dapat mengabaikan kesalahan lain yang sebenarnya harus diperbaiki. Jadi, dalam contoh di atas, pendekatan menggunakan pernyataan IF bisa dibilang lebih unggul, karena digunakan untuk menghilangkan secara eksplisit hanya kasus kesalahan yang muncul dari jangka waktu pinjaman, daripada menghilangkan semua kesalahan apa pun sifatnya.

#### **Contoh: SWITCH**

Fungsi SWITCH dapat digunakan untuk menghindari pernyataan IF yang tertanam, dan dalam hal ini mirip dengan banyak fungsi pencarian (lihat Bab 9). Dalam gambar 1.23



menunjukkan sebuah contoh (lihat Gambar 1.23).

	A	B	C	D	E	F	G
1							
2		<b>Month</b>	<b>Main Activity</b>				
3		January	Work				
4		February	Work				
5		March	Work				
6		April	Vacation				
7		May	Work				
8		June	Work				
9		July	Vacation				
10		August	Vacation				
11		September	Work				
12		October	Work				
13		November	Work				
14		December	Vacation				
15							
16		<b>Month</b>	<b>Main Activity</b>				
17		April	Vacation	=SWITCH(B17,B3,C3,B4,C4,B5,C5,B6,C6,B7,C7)			

**Gambar 1.23 Contoh Umum Fungsi SWITCH**

	A	B	C	D
1				
2		<b>Revenues Scenarios</b>		
3		Low	1	
4		Base	2	
5		High	3	
6				
7		Base	2	=SWITCH(\$B\$7,B3,C3,B4,C4,B5,C5)
8				

**Gambar 1.24 Menggunakan SWITCH untuk Mengubah Deskripsi Teks Menjadi Angka**

Gambar 1.24 menggunakan fungsi SWITCH untuk beralih dari deskripsi teks skenario menjadi deskripsi bernomor (lihat Gambar 1.24). (Dalam contoh nyata, nilai skenario numerik umumnya akan digunakan dalam perhitungan berikutnya yang mencari nilai masukan model dalam skenario tersebut dan yang mendorong perhitungan model akhir; lihat Bab 9.)

## BAB 2

### FUNGSI DAN RUMUS ARRAY

Bab ini menyajikan pembahasan umum tentang fungsi dan rumus array. Sebaiknya bahas hal ini sebelum membahas banyak fungsi Excel lainnya: fungsi array tidak hanya ada dalam beberapa kategori fungsi Excel, tetapi pada dasarnya fungsi Excel apa pun dapat digunakan sebagai bagian dari rumus array. Bab ini membahas prinsip-prinsip inti, yang digunakan di berbagai tempat lain dalam teks ini.

#### 2.1 FUNGSI DAN RUMUS

##### Definisi

Fitur penting dari fungsi dan rumus array adalah bahwa "di balik layar" mereka melakukan perhitungan yang seharusnya dilakukan dalam beberapa rentang atau tabel multisel. Output (pernyataan return) dari fungsi array umumnya meluas ke rentang beberapa sel yang bersebelahan; namun, beberapa hanya mengembalikan nilai ke satu sel. Perbedaan antara fungsi dan rumus adalah:

- Fungsi array adalah fungsi Excel bawaan yang secara inheren memerlukan penggunaan array yang bersebelahan dalam perhitungan atau bentuk outputnya. Fungsi-fungsi tersebut merupakan jenis fungsi, bukan kategori fungsi yang terpisah. Contohnya meliputi TRANSPOSE (kategori Lookup and Reference), MMULT, MINVERSE (kategori Math&Trig) dan FREQUENCY dan LINEST (kategori Statistik). Beberapa fungsi yang ditentukan pengguna yang dibuat menggunakan makro VBA juga dapat ditulis sebagai fungsi array.
- Rumus array menggunakan fungsi atau operasi Excel standar (seperti SUM, MIN atau IF), tetapi ditulis sedemikian rupa sehingga langkah-langkah perhitungan antara dilakukan secara inheren "di balik layar", daripada dihitung secara eksplisit dalam rentang Excel multisel.

#### 2.2 IMPLEMENTASI

Fungsi atau rumus array di Excel memungkinkan pengguna untuk melakukan perhitungan kompleks yang melibatkan banyak data sekaligus, tanpa perlu membuat rumus individual untuk setiap sel. Untuk mengimplementasikan rumus array, Anda perlu mengikuti langkah-langkah tertentu untuk memastikan rumus bekerja dengan benar.

Saat memasukkan rumus array, sangat penting untuk menempatkan kursor di dalam Bilah Rumus, dan bukan di dalam sel itu sendiri. Setelah rumus ditulis, Anda perlu menekan kombinasi tombol Ctrl+Shift+ENTER, yang membedakan rumus array dari rumus biasa. Ini akan memberi tanda kurung kurawal {} di sekitar rumus yang menunjukkan bahwa rumus tersebut

adalah array formula.

Contoh sederhana adalah penggunaan rumus array untuk menjumlahkan hasil perkalian dua kolom. Misalnya, jika Anda ingin mengalikan setiap nilai dalam Kolom A dengan nilai yang sesuai di Kolom B dan kemudian menjumlahkan hasilnya, rumus array yang digunakan adalah:

$$= \text{SUM} (A1:A5 * B1:B5)$$

Namun, rumus ini hanya dapat berfungsi jika Anda menekan **Ctrl+Shift+ENTER** setelah mengetikkan rumus. Setelah itu, Excel akan menampilkan rumus dalam format array, seperti ini:

$$\{=\text{SUM} (A1:A5 * B1:B5) \}$$

Selain itu, ada dua cara untuk memasukkan rumus array. Pertama, Anda dapat memilih rentang sel tempat rumus akan dimasukkan sebelum menulis rumus, dan kemudian tekan Ctrl+Shift+ENTER setelah menulisnya. Kedua, Anda dapat mengetikkan rumus dalam satu sel, kemudian memilih seluruh rentang sel tempat rumus akan diterapkan, dan tekan Ctrl+Shift+ENTER lagi untuk memperluas hasil rumus ke seluruh rentang.

Penting untuk diperhatikan bahwa rumus array dapat mengonsumsi banyak sumber daya komputer, terutama jika digunakan untuk menghitung rentang data yang besar. Oleh karena itu, berhati-hatilah dalam penggunaan rumus array untuk memastikan kinerja Excel tetap optimal.

Terakhir, perlu dicatat bahwa mulai dari Excel 365 dan Excel 2021, rumus array dinamis (dynamic array formulas) diperkenalkan. Dengan fitur ini, Anda tidak lagi memerlukan kombinasi Ctrl+Shift+ENTER, karena Excel secara otomatis mengenali rumus array dan mengembalikan hasil yang sesuai ke rentang yang diperlukan tanpa langkah tambahan. Meskipun demikian, penting untuk memahami cara kerja rumus array klasik agar tetap dapat bekerja dengan berbagai versi Excel dan untuk kebutuhan khusus.

### **Keuntungan dan Kerugian**

Dalam beberapa konteks tertentu, fungsi array hampir tidak dapat dihindari, karena alternatifnya jauh kurang efisien (contoh yang diberikan kemudian dalam bab ini dan dalam teks berikutnya seharusnya memperjelas hal ini). Di sisi lain, penggunaan rumus array umumnya merupakan pilihan. Manfaat yang mungkin termasuk:

- Untuk membuat model yang lebih transparan, di mana bagian yang terlihat secara eksplisit difokuskan pada tampilan input, output utama, dan kalkulasi ringkasan, sedangkan tabel kalkulasi antara (yang nilai individualnya tidak memiliki minat khusus) dilakukan di balik layar.
- Untuk membuat rumus yang lebih fleksibel jika ada perubahan pada ukuran rentang

data atau sumbu waktu.

- Untuk menghindari keharusan mengimplementasikan beberapa jenis kalkulasi sebagai fungsi yang ditentukan pengguna VBA. Ada beberapa kerugian dalam menggunakan fungsi dan rumus array:
- Memasukkan rumus secara tidak benar (misalnya dengan hanya menggunakan ENTER alih-alih Ctrl+Shift+ENTER) dapat mengakibatkan nilai yang salah (seperti nol) muncul di sel; Bahaya utamanya adalah nilai-nilai tersebut mungkin tidak jelas salah, dan karenanya dapat diabaikan (berbeda dengan kasus-kasus ketika pesan #VALUE muncul). Dengan demikian, kesalahan yang tidak disengaja dapat muncul.
- Banyak pengguna yang tidak familier dengan nilai-nilai tersebut, yang dapat mengakibatkan model menjadi lebih sulit dipahami atau ditafsirkan oleh orang lain (atau ketika pengguna mungkin secara tidak sengaja mengedit rumus dan mengembalikan ENTER).
- Dalam beberapa kasus, keberadaan nilai-nilai tersebut dapat memperlambat perhitungan buku kerja.

## 2.3 FUNGSI ARRAY

### 2.3.1 Contoh: Jadwal Capex dan Penyusutan Menggunakan TRANSPOSE

Fungsi TRANSPOSE dapat digunakan untuk mengubah data atau perhitungan yang berada dalam satu baris menjadi kolom, dan sebaliknya. Transposisi semacam itu dapat dilakukan secara eksplisit dalam rentang Excel, atau disematkan dalam rumus. Gambar 2.1 menunjukkan contoh setiap pendekatan dalam konteks perhitungan “tipe segitiga” untuk biaya penyusutan (seperti yang disajikan dalam Bab 1).

Pada lembar kerja “Range”, fungsi TRANSPOSE digunakan untuk secara eksplisit membuat rentang yang berisi data belanja modal dalam kolom (sel C9:C14), sehingga kumpulan data ini dapat dikalikan dengan data kolom persentase penyusutan menggunakan fungsi SUMPRODUCT (lihat Gambar 2.1). Pada lembar kerja “Formula”, data belanja modal yang ditransposisikan tidak dimasukkan secara eksplisit di Excel; melainkan secara implisit disematkan sebagai array, dengan menggunakan TRANSPOSE dalam rumus SUMPRODUCT (lihat Gambar 2.2).

### Contoh: Alokasi Biaya Menggunakan SUMPRODUCT dengan TRANSPOSE

Gambar 2.3 menunjukkan aplikasi untuk alokasi biaya yang diproyeksikan dari serangkaian departemen overhead pusat ke unit bisnis, berdasarkan matriks alokasi; cuplikan layar file tersebut ditunjukkan pada Gambar 2.3.

	A	B	C	D	E	F	G	H	I	J
1										
2		Years		2015	2016	2017	2018	2019	2020	
3		Capex		100	105	120	80	50	100	
4										
5		Generic Periods		1	2	3	4	5		
6		Depreciation Schedule	100%	30%	25%	20%	15%	10%		
7										
8		Depreciation		2015	2016	2017	2018	2019	2020	2021
9		2015	100	30%	25%	20%	15%	10%		
10		2016	105		30%	25%	20%	15%	10%	
11		2017	120			30%	25%	20%	15%	10%
12		2018	80				30%	25%	20%	15%
13		2019	50					30%	25%	20%
14		2020	100						30%	25%
15		Total, using direct reference from Row 3		30.0	56.5	82.3	=SUMPRODUCT(\$C9:\$C14,G9:G14)			

Gambar 2.1 Menggunakan TRANSPOSE untuk Mengubah Rentang Data

	A	B	C	D	E	F	G	H	I	J	K
1											
2		Years		2015	2016	2017	2018	2019	2020		
3		Capex		100	105	120	80	50	100		
4											
5		Generic Periods		1	2	3	4	5			
6		Depreciation Schedule	100%	30%	25%	20%	15%	10%			
7											
8		Depreciation		2015	2016	2017	2018	2019	2020	2021	2022
9		2015		30%	25%	20%	15%	10%			
10		2016			30%	25%	20%	15%	10%		
11		2017				30%	25%	20%	15%	10%	
12		2018					30%	25%	20%	15%	
13		2019						30%	25%	20%	
14		2020							30%	25%	
15		Total, using direct reference from Row 3		30.0	56.5	82.3	=SUMPRODUCT(TRANSPOSE(\$D3:\$I3),G9:G14)				

Gambar 2.2 Menggunakan TRANSPOSE yang Tertanam dalam Rumus

Sekali lagi, fungsi TRANSPOSE digunakan dalam SUMPRODUCT untuk memastikan bahwa kumpulan data yang dikalikan secara implisit merupakan data baris (atau kolom); alternatifnya adalah dengan mengubah semua data secara eksplisit dalam salah satu dari dua rentang data input.

### 2.3.2 Contoh: Alokasi Biaya dengan Perkalian Matriks Menggunakan MMULT

Excel memiliki beberapa fungsi array yang berhubungan dengan pendekatan berorientasi matematis. Faktanya, contoh alokasi biaya yang ditunjukkan pada Gambar 2.3 juga dapat dianggap sebagai perkalian matriks di mana satu vektor baris dan satu vektor kolom dikalikan bersama menggunakan MMULT.

Perhatikan bahwa saat melakukannya, menurut konvensi matematika standar, vektor baris harus menjadi yang pertama dari dua entri argumen fungsi, dan tidak diperlukan perubahan data. Gambar 2.4 menunjukkan implementasi ini dalam contoh alokasi biaya yang digunakan di atas; klip layar ditunjukkan pada Gambar 2.4.

	A	B	C	D	E	F	G	H
1								
2		Example 1: Using Array functions						
3								
4		<b>Indirect Expenses</b>	2017	2018	2019	2020	2021	
5		Development	6,000	6,300	6,615	6,946	7,293	
6		Research	25,000	26,250	27,563	28,941	30,388	
7		Supply Chain	7,000	7,350	7,718	8,103	8,509	
8		Group IT	12,000	12,600	13,230	13,892	14,586	
9		Sales & Marketing	25,000	26,250	27,563	28,941	30,388	
10		Group Support	8,000	8,400	8,820	9,261	9,724	
11		<b>Total (\$k)</b>	<b>83,000</b>	<b>87,150</b>	<b>91,508</b>	<b>96,083</b>	<b>100,887</b>	
12								
13		<b>Business Units</b>	Development	Research	Supply Chain	Group IT	Sales & Marketing	Group Support
14		BU1	35%	20%	20%	25%	30%	25%
15		BU2	20%	25%	35%	20%	20%	20%
16		BU3	30%	35%	25%	25%	20%	35%
17		BU4	15%	20%	20%	30%	30%	20%
18		<b>Total</b>	<b>100%</b>	<b>100%</b>	<b>100%</b>	<b>100%</b>	<b>100%</b>	<b>100%</b>
19								
20		<b>Business Units</b>	2017	2018	2019	2020	2021	
21		BU1	21,000	22,050	23,153	24,310	25,526	
22		BU2	18,900	19,845	=SUMPRODUCT(E55:E510,TRANSPOSE(SC15:SH15))			
23		BU3	23,100	24,255	25,468	26,741	28,078	
24		BU4	20,000	21,000	22,050	23,153	24,310	
25		<b>Total (\$k)</b>	<b>83,000</b>	<b>87,150</b>	<b>91,508</b>	<b>96,083</b>	<b>100,887</b>	

Gambar 2.3 Penggunaan TRANSPOSE Dengan SUMPRODUCT Dalam Alokasi Biaya

	A	B	C	D	E	F	G	H
1								
2		Example 1: Using Array functions						
3								
4		<b>Indirect Expenses</b>	2017	2018	2019	2020	2021	
5		Development	6,000	6,300	6,615	6,946	7,293	
6		Research	25,000	26,250	27,563	28,941	30,388	
7		Supply Chain	7,000	7,350	7,718	8,103	8,509	
8		Group IT	12,000	12,600	13,230	13,892	14,586	
9		Sales & Marketing	25,000	26,250	27,563	28,941	30,388	
10		Group Support	8,000	8,400	8,820	9,261	9,724	
11		<b>Total (\$k)</b>	<b>83,000</b>	<b>87,150</b>	<b>91,508</b>	<b>96,083</b>	<b>100,887</b>	
12								
13		<b>Business Units</b>	Development	Research	Supply Chain	Group IT	Sales & Marketing	Group Support
14		BU1	35%	20%	20%	25%	30%	25%
15		BU2	20%	25%	35%	20%	20%	20%
16		BU3	30%	35%	25%	25%	20%	35%
17		BU4	15%	20%	20%	30%	30%	20%
18		<b>Total</b>	<b>100%</b>	<b>100%</b>	<b>100%</b>	<b>100%</b>	<b>100%</b>	<b>100%</b>
19								
20		<b>Business Units</b>	2017	2018	2019	2020	2021	
21		BU1	21,000	22,050	23,153	24,310	25,526	
22		BU2	18,900	19,845	=MMULT(SC15:SH15,E55:E510)			
23		BU3	23,100	24,255	25,468	26,741	28,078	
24		BU4	20,000	21,000	22,050	23,153	24,310	
25		<b>Total (\$k)</b>	<b>83,000</b>	<b>87,150</b>	<b>91,508</b>	<b>96,083</b>	<b>100,887</b>	

Gambar 2.4 Penggunaan MMULT dalam Alokasi Biaya

### 2.3.3 Contoh: "Perhitungan Biaya Aktivitas dan Peramalan Sumber Daya"

Saat membangun model untuk meramalkan sumber daya yang diperlukan agar dapat mencapai dan memberikan profil pendapatan yang diinginkan, seseorang mungkin memiliki beberapa sumber daya yang aktivitasnya ditentukan oleh beberapa faktor yang mendasarinya.

Seseorang mungkin memiliki beberapa contoh, atau dapat membuat estimasi skenario yang memungkinkan dalam hal tingkat sumber daya yang diperlukan untuk berbagai tingkat

penggerak aktivitas; dari sini, seseorang dapat menentukan koefisien penskalaan (dengan asumsi penskalaan linier dari tingkat aktivitas ke persyaratan sumber daya). Misalnya, jika tingkat sumber daya (staf) 30 diamati untuk melayani 25 pelanggan yang tersebar di 150 lokasi, sedangkan 35 staf diperlukan untuk melayani 20 pelanggan di 200 lokasi, seseorang dapat menuliskannya dalam bentuk koefisien aktivitas:

$$R = A_0S + A_1C$$

di mana R menunjukkan tingkat sumber daya, S jumlah lokasi, dan C jumlah pelanggan. Jika persamaan yang sesuai dengan dua kasus spesifik ditulis secara lengkap, maka aljabar matriks dapat digunakan untuk mencari koefisien A0 dan A1. Berkas Ch18.4.CostDriverForecasting.xlsx berisi implementasi ini (lihat Gambar 18.5). Perhatikan bahwa jika koefisien A0 dan A1 dianggap sebagai komponen vektor kolom, keduanya merupakan solusi persamaan:

$$R^T = D^T A$$

di mana R adalah kumpulan sumber daya yang digunakan (sel C5:D5), D adalah data terperinci dari lokasi dan pelanggan (sel C3:D4), dan T mewakili operasi transpos. Oleh karena itu, A dapat ditemukan dengan menginversikan transpos D dan mengalikannya dengan vektor sumber daya:

$$A = (D^T)^{-1}R^T$$

	A	B	C	D	E
1					
2			BU1	BU2	
3		No of sites	150	200	
4		No of customers	25	20	
5		Staff Level	30	35	Staff=A0.Sites+A1.Customers
6					
7		Transpose	150	25	{=TRANSPOSE(C3:D4)}
8			200	20	
9					
10		Inverse	-0.01	0.01	{=MINVERSE(C7:D8)}
11			0.10	-0.08	
12					
13		Transpose	30		{=TRANSPOSE(C5:D5)}
14			35		
15					
16		Coefficients			
17		A0	0.14		{=MMULT(C10:D11,C13:C14)}
18		A1	0.38		
19					
20		Example			
21		No of sites	320		
22		No of customers	40		
23		Staff Level	59		{=SUMPRODUCT(C17:C18,C21:C22)}

**Gambar 2.5 Menggunakan Fungsi Array untuk Perhitungan Matriks guna Menghitung Koefisien Penggerak Biaya**

Baris 7 hingga Baris 18 menunjukkan langkah-langkah individual dari operasi ini (yang dapat

digabungkan menjadi rentang kolom tunggal jika diinginkan, daripada dipecah sebagai langkah-langkah individual). Setelah komponen ditentukan, komponen tersebut dapat diterapkan pada kasus (atau beberapa kasus) baru yang akan muncul dalam model perkiraan (di mana jumlah pelanggan dan lokasi diperkirakan, dan bagian model ini juga menentukan persyaratan sumber daya), seperti yang ditunjukkan dalam sel C21 hingga C23. Fungsi array yang digunakan adalah TRANSPOSE, MINVERSE, dan MMULT.

### 2.3.4 Contoh: Menjumlahkan Pangkat Bilangan Bulat dari 1 dan seterusnya

Fungsi array MINVERSE menemukan invers dari matriks persegi (pembaca yang cenderung matematika mungkin juga ingin menggunakan MDETERM untuk memeriksa bahwa determinannya bukan nol, dan karenanya invers dapat ditemukan). Meskipun fungsi tersebut memiliki beberapa kegunaan dalam aplikasi keuangan yang lebih maju (seperti pengoptimalan portofolio aset keuangan, dan dalam menentukan harga keadaan yang netral terhadap risiko, antara lain), pada titik ini dalam teks kita akan menggunakannya untuk menemukan solusi untuk masalah numerik umum. Misalnya, diketahui bahwa jumlah bilangan bulat dari 1 hingga N diberikan oleh:

$$1 + 2 + 3 + \dots + N = \frac{N(N + 1)}{2}$$

Atau

$$S_N^1 = \frac{N(N + 1)}{2}$$

Seperti yang umum diketahui, rumus ini dapat diperoleh hanya dengan menuliskan himpunan bilangan bulat dua kali, sekali dalam urutan alami dan sekali dalam urutan terbalik: ini menghasilkan N pasang bilangan bulat, yang masing-masing berjumlah N+1. Untuk menghubungkannya dengan pembahasan di bawah ini dengan lebih jelas, kami mencatat bahwa:

$$S_N^1 = \frac{1}{2}N^2 + \frac{1}{2}N^1 + 0N^0$$

Pada langkah berikutnya, seseorang mungkin ingin menemukan ekspresi untuk jumlah pangkat dari setiap bilangan bulat dari 1 hingga bilangan apa pun N (misalnya jumlah kuadrat dari setiap bilangan bulat, atau jumlah pangkat tiga dari setiap bilangan bulat). Dengan kata lain, untuk pangkat yang dipilih, p, kita bertujuan untuk menemukan ekspresi untuk:

$$S_N^p = 1^p + 2^p + 3^p + \dots + N^p$$

Karena ada N suku dan pangkatnya adalah p, kita dapat berhipotesis bahwa ekspresi yang



diinginkan berbentuk:

$$S_N^P = C_{P+1}N^{P+1} + C_P N^P + \dots + C_1 N^1 + C_0 N^0$$

Misalnya, dalam kasus  $p = 2$ , kita dapat berhipotesis bahwa:

$$S_N^2 = C_3 N^3 + C_2 N^2 + C_1 N^1 + C_0 N^0$$

dengan koefisien  $C_i$  yang masih harus ditentukan (untuk  $i=0$  hingga 3).

Jika jumlah kuadrat dievaluasi untuk setiap bilangan bulat ( $i$ ) dari 0 hingga 3, kita memperoleh:

$$\begin{aligned} S_0^2 &= 0^2 = 0 \\ S_1^2 &= 1^2 = 1 \\ S_2^2 &= 1^2 + 2^2 = 5 \\ S_3^2 &= 1^2 + 2^2 + 3^2 = 14 \end{aligned}$$

Dengan demikian:

$$\begin{aligned} 0 &= S_0^2 = C_0 \\ 1 &= S_1^2 = C_3 + C_2 + C_1 \\ 5 &= S_2^2 = C_3 2^3 + C_2 2^2 + C_1 2^1 \\ 14 &= S_3^2 = C_3 3^3 + C_2 3^2 + C_1 3^1 \end{aligned}$$

Kita dapat langsung melihat bahwa  $C_0 = 0$  (dan ini akan menjadi kasus untuk pangkat  $p$  apa pun); oleh karena itu koefisien ini tidak akan dibahas di masa mendatang. Salah satu cara untuk menemukan nilai koefisien yang tersisa adalah dengan menyelesaikannya dengan substitusi berulang (pertama dengan lintasan maju, dan kemudian lintasan mundur): untuk lintasan maju, pertama,  $C_1$  dihilangkan dengan menggunakan persamaan kedua untuk menulis:

$$C_1 = 1 - C_3 - C_2$$

Persamaan ketiga kemudian menjadi:

$$5 = C_3 2^3 + C_2 2^2 + (1 - C_3 - C_2) 2^1$$

Hal ini dapat disusun ulang untuk mengekspresikan  $C_2$  dalam bentuk  $C_3$ :

$$C_2 = \frac{3 - 6C_3}{2}$$

Persamaan keempat kemudian dapat ditulis seluruhnya dalam bentuk  $C_3$ , dengan mensubstitusikan suku-suku yang melibatkan  $C_1$  untuk  $C_2$ , dan kemudian dengan cara yang sama  $C_2$  untuk  $C_3$ , untuk memperoleh persamaan yang hanya melibatkan  $C_3$ , yaitu:

$$14 = C_3 3^3 + \left( \frac{3 - 6C_3}{2} \right) 3^2 + \left( 1 - C_3 - \left( \frac{3 - 6C_3}{2} \right) \right) 3^1$$

Ini memberikan:

$$C_3 = \frac{1}{3}$$

Kemudian lintasan mundur menghitung  $C_2$  dari  $C_3$  (untuk memberikan

$$C_2 = \frac{1}{2}$$

), dan kemudian  $C_1$ , dari  $C_3$  dan  $C_2$  (untuk memberikan

$$C_1 = \frac{1}{6}$$

), sehingga:

$$S_N^2 = \frac{1}{3}N^3 + \frac{1}{2}N^2 + \frac{1}{6}N$$

Di sisi lain, daripada memecahkan persamaan secara eksplisit dengan cara ini, seseorang dapat menggunakan aljabar matriks untuk menyatakan tiga persamaan terakhir (karena persamaan pertama memiliki solusi  $C_0 = 0$ ) sebagai:

$$\begin{pmatrix} 1 & 1 & 1 \\ 8 & 4 & 2 \\ 27 & 9 & 3 \end{pmatrix} \begin{pmatrix} C_3 \\ C_2 \\ C_1 \end{pmatrix} = \begin{pmatrix} 1 \\ 5 \\ 14 \end{pmatrix}$$

Oleh karena itu, vektor  $C_i$  dapat ditemukan dengan inversi matriks:

$$\begin{pmatrix} C_3 \\ C_2 \\ C_1 \end{pmatrix} = \begin{pmatrix} 1 & 1 & 1 \\ 8 & 4 & 2 \\ 27 & 9 & 3 \end{pmatrix}^{-1} \begin{pmatrix} 1 \\ 5 \\ 14 \end{pmatrix}$$

Hal ini dapat dengan mudah dihitung di Excel menggunakan fungsi MINVERSE. Gambar 2.6 menunjukkan perhitungan ini, serta koefisien yang dihasilkan untuk pangkat pertama, kedua, ketiga, keempat, dan kelima. Perhatikan bahwa hasilnya diformat menggunakan opsi

pemformatan Pecahan Excel; Gambar 2.6 menunjukkan cuplikan layar.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
1																		
2		Sum of integers	1											Totals		Result		i.e. $=(1/2)n^2+(1/2)n+1$
3	=		1		1	1	3	C0						1		0		
4	=		2		1	2	4	C1						9		1/2		
5	=		3		1	3	9	C2						6		1/2		
6																		
7		Sum of squares	2											Total		Result		i.e. $=(1/3)n^3+(1/2)n^2+(1/6)n$
8	=		1		1	1	1	C0						1		0		
9	=		2		1	2	4	C1						5		1/6		
10	=		3		1	3	9	C2						14		1/2		
11	=		4		1	4	16	C3						30		1/3		
12																		
13		Sum of cubes	3											Total		Result		i.e. $=(1/4)n^4+(1/2)n^3+(1/4)n^2$
14	=		1		1	1	1	C0						1		-0		
15	=		2		1	2	4	C1						9		0		
16	=		3		1	3	9	C2						36		1/4		
17	=		4		1	4	16	C3						100		1/2		
18	=		5		1	5	25	C4						225		1/4		
19																		
20		Sum of fourth powers	4											Total		Result		i.e. $=(1/5)n^5+(1/2)n^4+(1/3)n^3-(1/30)n$
21	=		1		1	1	1	C0						1		-0		
22	=		2		1	2	4	C1						17		- 1/30		
23	=		3		1	3	9	C2						98		-0		
24	=		4		1	4	16	C3						354		1/3		
25	=		5		1	5	25	C4						979		1/2		
26	=		6		1	6	36	C5						2275		1/5		
27																		
28		Sum of fifth powers	5											Total		Result		i.e. $=(1/6)n^6+(1/2)n^5+(5/12)n^4-(1/12)n^2$
29	=		1		1	1	1	C0						1		-0		
30	=		2		1	2	4	C1						31		0		
31	=		3		1	3	9	C2						276		- 1/12		
32	=		4		1	4	16	C3						1300		-0		
33	=		5		1	5	25	C4						4425		5/12		
34	=		6		1	6	36	C5						12201		1/2		
35	=		7		1	7	49	C6						29008		1/6		

Gambar 2.6 Perhitungan Jumlah Pangkat Bilangan Bulat Menggunakan Minverse

Dari hasil yang diperoleh, kita dapat melihat:

$$S_N^1 = \frac{1}{2}N^2 + \frac{1}{2}N$$

$$S_N^2 = \frac{1}{3}N^3 + \frac{1}{2}N^2 + \frac{1}{6}N$$

$$S_N^3 = \frac{1}{4}N^4 + \frac{1}{2}N^3 + \frac{1}{2}N^2$$

$$S_N^4 = \frac{1}{5}N^5 + \frac{1}{2}N^4 + \frac{1}{3}N^3 - \frac{1}{30}N$$

$$S_N^5 = \frac{1}{6}N^6 + \frac{1}{2}N^5 + \frac{5}{12}N^4 - \frac{1}{12}N$$

Tentu saja, persamaan untuk pangkat yang lebih tinggi juga dapat dengan mudah diturunkan dengan memperluas prosedur.

## 2.4 APLIKASI PRAKTIS: RUMUS ARRAY

### 2.4.1 Contoh: Menemukan Item Positif Pertama dalam Daftar

Pada Bab 1, kami mencatat bahwa fungsi "MAXIF" atau "MAXIFS" diperkenalkan ke Excel pada tahun 2016. Sebelumnya (dan karenanya dalam beberapa model yang dibuat

sebelum waktu ini), fungsionalitas tersebut dapat dicapai dengan menggabungkan fungsi MAX (atau MIN) dan IF dalam rumus array. Pada gambar 2.7 menunjukkan contoh, yang digunakan dalam konteks menemukan tanggal nilai negatif pertama dan nilai positif pertama dalam deret waktu; Gambar 2.7 menunjukkan klip layar. Rumus array menyematkan pernyataan IF dalam fungsi MIN untuk menemukan nilai minimum (dalam serangkaian tanggal numerik) yang memenuhi syarat (baik arus kas positif maupun negatif). Misalnya, rumus di Sel C7 adalah:

C7 {MIN(IF(E7 : BL7 < 0, E2 : BL2))}

	A	B	C	D	E	F	G	H	I	J	K	L	M
1													
2			Period Start		1-Jan-16	1-Apr-16	1-Jul-16	1-Oct-16	1-Jan-17	1-Apr-17	1-Jul-17	1-Oct-17	1-Jan-18
3			Period End		31-Mar-16	30-Jun-16	30-Sep-16	31-Dec-16	31-Mar-17	30-Jun-17	30-Sep-17	31-Dec-17	31-Mar-18
4													
5			First Negative Cash Flow	First Positive Cash Flow	Cash Flows								
6	Project 1		1-Jul-16	1-Apr-17	-	-	(200)	(80)	(90)	20	30	45	45
7	Project 2		1-Jan-17	1-Jan-18	-	-	-	(100)	(90)	(45)	(40)	50	
8	Project 3		1-Jan-16	1-Oct-16	(400)	(300)	(200)	50	80	110	150	150	170
9													
10	Project 1	First Negative Cash Flow	Formula										
11	Correct	1-Jul-16	=MIN(IF(E6:BL6<0;E\$2:S\$2))										
12	Incorrect	0-Jan-00	=MIN(IF(E6:BL6<0;1,0)*E\$2:S\$2))										
13			Returns zeros due to zeros being the default multiplier when the condition is not met.										
14													
15	Using MINIFS	First Negative Cash Flow	First Positive Cash Flow										
16	Project 1		1-Jul-16	1-Apr-17									
17	Project 2		1-Jan-17	1-Jan-18									
18	Project 3		1-Jan-16	1-Oct-16									

**Gambar 2.7 Menggabungkan MIN Dan IF Dalam Rumus Array Untuk Menemukan Tanggal Arus Kas Pertama**

Perhatikan sintaksis fungsi IF dalam rumus: tidak ada pernyataan eksplisit mengenai penanganan kasus-kasus di mana kondisi (seperti E7:BL7<0) belum terpenuhi; rumus secara eksplisit merujuk ke rentang tanggal hanya untuk kasus-kasus di mana kondisi terpenuhi. Seseorang dapat mencoba membuat kasus-kasus tersebut lebih eksplisit, misalnya dengan menggunakan rumus seperti:

C7 {MIN(IF(E7 : BL7 < 0,1,0)\*(E2 : BL2))}

Namun, hasilnya akan menjadi minimum nol (dalam kasus ini), karena setiap kali kondisi tidak terpenuhi, hasil pernyataan IF akan dievaluasi menjadi nol, yang kemudian akan dikalikan dengan tanggal numerik (positif) yang sesuai, untuk memberikan serangkaian data yang positif atau nol (dan dengan demikian memiliki minimum nol). Fakta bahwa penanganan kasus di mana kondisi tidak terpenuhi tidak ditunjukkan secara eksplisit memang mengurangi transparansi, karena seseorang harus tahu (atau mungkin harus berhenti sejenak untuk mempertimbangkan atau menguji) bagaimana rumus array menangani kasus tersebut. Ini mungkin menjadi alasan untuk berhati-hati dalam menggunakan pendekatan tersebut jika alternatif lain tersedia. Berkas tersebut juga menunjukkan perhitungan yang setara menggunakan fungsi MAXIFS dan MINIFS (yang tidak ditampilkan dalam klip layar, tetapi dapat dirujuk oleh pembaca yang tertarik).

### 2.4.2 Contoh: Menemukan Nilai Maksimum Bersyarat

Seseorang dapat menggeneralisasi metode dalam contoh sebelumnya untuk menemukan nilai maksimum atau minimum bersyarat, tergantung pada penerapan beberapa kriteria. Gambar 2.8 menunjukkan contoh di mana seseorang ingin menemukan nilai maksimum dari kolom "Jumlah £", baik untuk pelanggan atau negara, maupun untuk kombinasi pelanggan-negara. Gambar 2.8 memperlihatkan cuplikan layar, dengan berkas yang berisi pernyataan rumus IF seperti:

C5 {MAX(IF(B7 : B106 =B3, IF(C7 : C106 =C3, E7 : E106)))}

C6 {MIN(IF(B7 : B106 =B3, IF(C7 : C106 =C3, E7 : E106)))}

	A	B	C	D	E	F	G	H	I
1									
2		Customer	Country			Item		As array formulae	Formula
3		Cust10	UK			Maximum amount: Cust10		14647	{=MAX(IF((B7:B106=B3),E7:E106))}
4						Minimum amount: Cust10		-11862	{=MIN(IF((B7:B106=B3),E7:E106))}
5						Maximum amount: Cust10 and UK		14001	{=MAX(IF(B7:B106=B3,IF(C7:C106=C3,E7:E106)))}
6		Customer	Country	Due Date	Amount £	Minimum amount: Cust10 and UK		-11862	{=MIN(IF(B7:B106=B3,IF(C7:C106=C3,E7:E106)))}
7		Cust02	UK	21-Mar-15	12232				
8		Cust06	Italy	17-Mar-15	4749				
9		Cust07	Italy	13-Apr-15	7282	Incorrect:			
10		Cust03	Italy	15-Jun-15	12759	Minimum amount: Cust10 and UK		0	{=MIN(IF(AND(B7:B106=B3,C7:C106=C3),1,0)*E7:E106)}
11		Cust10	UK	25-May-15	12334	Non-embedded conditions		15851	{=MAX(IF(B7:B106=B3,C7:C106=C3),E7:E106)}
12		Cust05	Italy	25-Mar-15	4283				
13		Cust06	Germany	6-May-15	7992				
14		Cust06	Italy	17-Apr-15	13202				
15		Cust04	Germany	5-Jun-15	12684				
16		Cust10	UK	14-Jun-15	-11862				

**Gambar 2.8 Penerapan Beberapa Kriteria Menggunakan Rumus Array untuk Menghitung Maksimum dan Minimum Bersyarat**

Perhatikan bahwa, seperti yang ditunjukkan dalam sel G10 dan G11:

- Sekali lagi, mencoba membuat kasus yang kriterianya tidak terpenuhi (sel G10) lebih eksplisit kemungkinan akan menghasilkan perhitungan yang salah (nilai minimum tampak nol, karena kondisional tidak terpenuhi, bukan nilai minimum sebenarnya, yaitu -11862 di Sel E16).
- Menggunakan fungsi IF tunggal (non-embedded) juga tidak akan benar.

### 2.4.3 Contoh: Nilai Maksimum Bersyarat dengan AGGREGATE

Contoh-contoh di atas telah menunjukkan bahwa meskipun seseorang dapat menggunakan rumus array untuk menghitung nilai maksimum dan minimum bersyarat, ada beberapa kelemahan, termasuk bahwa penanganan kasus-kasus yang kondisinya tidak terpenuhi tidak terlalu eksplisit, dan bahwa penerapan beberapa kriteria akan sering memerlukan pernyataan IF yang disematkan. Fungsi AGGREGATE (Excel 2013 dan seterusnya) dapat digunakan sebagai alternatif. Salah satu keuntungannya adalah sifat perhitungan dapat diubah atau disalin dengan cepat (misalnya, alih-alih nilai maksimum atau minimum, seseorang dapat beralih ke nilai rata-rata) menggunakan serangkaian jenis fungsi yang

memungkinkan (seperti yang ditunjukkan pada Bab 1, Gambar 1.19).

Untuk menggunakan fungsi dalam konteks kueri bersyarat, seseorang membuat perhitungan yang menghasilkan kesalahan saat kondisi tidak terpenuhi, dan menggunakan fungsi dalam bentuk yang mengabaikan kesalahan (Opsi 6). Karena kita menggunakan rumus array, bentuk array (bukan bentuk referensi) dari fungsi AGGREGATE harus digunakan; yaitu, untuk nilai maksimum, kita menggunakan fungsi LARGE (bukan MAX) dan untuk nilai minimum kita menggunakan SMALL (bukan MIN).

	A	B	C	D	E	F	G	H
1								
2		Customer	Country				As array function	Formula
3		Cust10	UK					14001 {=AGGREGATE(14,6,((E7:E106)/((B7:B106=B\$3)*(C7:C106=C\$3))),1)}
4								
5								
6		Customer	Country	Due Date	Amount £			
7		Cust02	UK	21-Mar-15	12232			
8		Cust06	Italy	17-Mar-15	4749			
9		Cust07	Italy	13-Apr-15	7282			
10		Cust03	Italy	15-Jun-15	12759			
11		Cust10	UK	25-May-15	12334			
12		Cust05	Italy	25-Mar-15	4283			
13		Cust06	Germany	6-May-15	7992			
14		Cust06	Italy	17-Apr-15	13202			
15		Cust04	Germany	5-Jun-15	12684			
16		Cust10	UK	14-Jun-15	11862			
17		Cust10	Italy	22-May-15	13630			
18		Cust07	UK	21-Jan-15	14593			
19		Cust07	Italy	5-May-15	4394			

**Gambar 2.9 Contoh Penggunaan Rumus Array dengan AGGREGATE untuk Menghitung Nilai Maksimum Bersyarat dari Kumpulan Data**

Kolom J hingga M pada berkas yang sama menunjukkan bagaimana perhitungan juga dapat dilakukan secara langsung menggunakan bentuk non-array dari fungsi AGGREGATE, selama perhitungan yang tersirat (di balik layar) untuk rumus array dilakukan secara eksplisit di Excel (lihat Gambar 2.10). Dengan demikian, rumus array dapat menghemat banyak ruang dan mungkin lebih mudah dimodifikasi dalam beberapa kasus (misalnya saat ukuran kumpulan data diubah). Perhatikan bahwa perhitungan yang setara tidak dapat dilakukan dengan fungsi MIN atau MAX, karena fungsi-fungsi ini tidak diperbolehkan memiliki kesalahan dalam kumpulan datanya.

J	K	L	M	N
<b>Without an Array Formula</b>		<b>Result</b>	<b>Formula</b>	
Conditional Maximum using MAX		14001	=AGGREGATE(4,6,K7:K106)	
Conditional Maximum using LARGE		14001	=AGGREGATE(14,6,K7:K106,1)	
<b>CriteriaMet (1 or #DIV/0!)</b>		<b>New Amount</b>	<b>Column J</b>	<b>Column K</b>
#DIV/0!	#DIV/0!		=IF(AND(B7=B\$3,C7=C\$3),1,1/0)	=E7*J7
#DIV/0!	#DIV/0!		=IF(AND(B8=B\$3,C8=C\$3),1,1/0)	=E8*J8
#DIV/0!	#DIV/0!		=IF(AND(B9=B\$3,C9=C\$3),1,1/0)	=E9*J9
#DIV/0!	#DIV/0!		=IF(AND(B10=B\$3,C10=C\$3),1,1/0)	=E10*J10
	1	12334	=IF(AND(B11=B\$3,C11=C\$3),1,1/0)	=E11*J11
#DIV/0!	#DIV/0!		=IF(AND(B12=B\$3,C12=C\$3),1,1/0)	=E12*J12
#DIV/0!	#DIV/0!		=IF(AND(B13=B\$3,C13=C\$3),1,1/0)	=E13*J13
#DIV/0!	#DIV/0!		=IF(AND(B14=B\$3,C14=C\$3),1,1/0)	=E14*J14
#DIV/0!	#DIV/0!		=IF(AND(B15=B\$3,C15=C\$3),1,1/0)	=E15*J15
	1	11862	=IF(AND(B16=B\$3,C16=C\$3),1,1/0)	=E16*J16
#DIV/0!	#DIV/0!		=IF(AND(B17=B\$3,C17=C\$3),1,1/0)	=E17*J17
#DIV/0!	#DIV/0!		=IF(AND(B18=B\$3,C18=C\$3),1,1/0)	=E18*J18
#DIV/0!	#DIV/0!		=IF(AND(B19=B\$3,C19=C\$3),1,1/0)	=E19*J19
#DIV/0!	#DIV/0!		=IF(AND(B20=B\$3,C20=C\$3),1,1/0)	=E20*J20
	1	6503	=IF(AND(B21=B\$3,C21=C\$3),1,1/0)	=E21*J21
#DIV/0!	#DIV/0!		=IF(AND(B22=B\$3,C22=C\$3),1,1/0)	=E22*J22
#DIV/0!	#DIV/0!		=IF(AND(B23=B\$3,C23=C\$3),1,1/0)	=E23*J23
#DIV/0!	#DIV/0!		=IF(AND(B24=B\$3,C24=C\$3),1,1/0)	=E24*J24
#DIV/0!	#DIV/0!		=IF(AND(B25=B\$3,C25=C\$3),1,1/0)	=E25*J25
#DIV/0!	#DIV/0!		=IF(AND(B26=B\$3,C26=C\$3),1,1/0)	=E26*J26
#DIV/0!	#DIV/0!		=IF(AND(B27=B\$3,C27=C\$3),1,1/0)	=E27*J27
#DIV/0!	#DIV/0!		=IF(AND(B28=B\$3,C28=C\$3),1,1/0)	=E28*J28
#DIV/0!	#DIV/0!		=IF(AND(B29=B\$3,C29=C\$3),1,1/0)	=E29*J29
	1	14001	=IF(AND(B30=B\$3,C30=C\$3),1,1/0)	=E30*J30
#DIV/0!	#DIV/0!		=IF(AND(B31=B\$3,C31=C\$3),1,1/0)	=E31*J31

**Gambar 2.10 Penggunaan Bentuk Non-array dari Fungsi AGGREGATE Berdasarkan Perhitungan Individu Eksplisit**

## BAB 3

### FUNGSI MATEMATIKA

#### 3.1 PENDAHULUAN

Bab ini membahas serangkaian fungsi Excel yang luas yang berkaitan dengan perhitungan aritmatika dan matematika. Kami berfokus pada fungsi-fungsi yang termasuk dalam kategori Matematika & Trigonometri Excel, yang umumnya paling sering digunakan dalam aplikasi pemodelan keuangan praktis. Kami tidak mengulang pembahasan untuk fungsi-fungsi yang dibahas dalam Bab 1.

Kami juga menyebutkan beberapa fungsi lain, untuk memberikan petunjuk tentang jenis kemungkinan yang ada secara lebih umum; pembaca dapat memilih untuk meninjau serangkaian fungsi lengkap dalam kategori ini untuk mengidentifikasi apakah fungsi lain mungkin berguna bagi mereka dalam konteks mereka sendiri. Contoh yang diberikan meliputi penggunaan:

- LN, EXP.
- ABS, SIGN.
- INT, ROUND, ROUNDUP, ROUNDDOWN, TRUNC.
- MROUND, CEILING.MATH, FLOOR.MATH.
- MOD.
- SQRT, POWER.
- FAKTA, KOMBINASI.
- SINE, ASIN, COS, ACOS, TAN, ATAN.
- DERAJAT, PI, SQRTPI.
- BASIS, DESIMAL.
- ARAB, ROMA.

#### 3.2 APLIKASI PRAKTIS

##### 3.2.1 Contoh: EXP dan LN

Fungsi EXP menghitung eksponen alami dari angka apa pun, yaitu untuk setiap input  $x$  fungsi ini menghitung  $y$ , di mana:

$$y = e^x$$

Fungsi LN adalah kebalikan dari fungsi ini, yaitu menghitung logaritma natural dari sebuah angka,  $y$ , yang merupakan nilai  $x$  yang akan menyelesaikan persamaan di atas untuk  $x$  ketika  $y$  adalah nilai input, sehingga  $x = \text{LN}(y)$ . Perhatikan bahwa logaritma dari hasil perkalian dua angka adalah jumlah dari logaritma masing-masing angka. Yaitu:



$$LN(y_1y_2) = LN(y_1) + LN(y_2)$$

Hal ini dapat dilihat dengan mudah, karena jika

$$y_1 = e^{x_1} \text{ dan } y_2 = e^{x_2} \\ (\text{i. e. } x_1 = LN(y_1) \text{ dan } x_2 = LN(y_2))$$

Kemudian

$$y_1y_2 = e^{x_1}e^{x_2} = e^{x_1+x_2}$$

sehingga (mengambil logaritma kedua sisi persamaan pertama dalam persamaan berikut, dan menggunakan definisi x dalam bentuk y untuk persamaan kedua):

$$LN(y_1y_2) = x_1 + x_2 = LN(y_1) + LN(y_2)$$

Penggunaan penting dari metode ini adalah dalam kalkulasi atau kalibrasi deret waktu. Salah satu ukuran pengembalian (atau perubahan atau tingkat pertumbuhan) dalam nilai aset (atau item positif lainnya, seperti pendapatan) ditemukan dengan mengambil logaritma dari rasio nilai-nilai:

Pengembalian, perubahan atau tingkat pertumbuhan  $LN(\text{NilaiAkhir} / \text{NilaiAwal})$ .

Kemudian, karena sifat aditif dari logaritma, jika seseorang memiliki sekumpulan data pengembalian untuk periode waktu individual (harian, bulanan, tahunan), seseorang dapat menghitung pengembalian untuk setiap subset data (seperti pengembalian total antara bulan ke-3 dan ke-7), dengan hanya menambahkan pengembalian periodik individual. Hal ini berbeda dengan banyak aplikasi "tradisional" atau "keuangan korporat", di mana seseorang sering mengukur pengembalian (atau perubahan, atau tingkat pertumbuhan) sebagai:

Pengembalian, perubahan, atau tingkat pertumbuhan  $(\text{NilaiAkhir} / \text{NilaiAwal}) - 1$ .

Dalam pendekatan terakhir ini, sifat aditif dari pengembalian hilang, sehingga ketika hanya diberikan data pengembalian (pertumbuhan atau perubahan harga) seseorang umumnya perlu menerapkan langkah perhitungan tambahan untuk merekonstruksi harga aset absolut guna menghitung pengembalian untuk subset data multiperiodik. Gambar 3.1 menunjukkan kedua jenis perhitungan tersebut. Dalam lembar kerja bernama LNEXP, seseorang memulai dari serangkaian harga aset periodik, dan menghitung pengembalian periodik menggunakan fungsi LN. Seolah-olah bekerja mundur dari pengembalian aset yang diberikan, fungsi SUM digunakan untuk mengakumulasikan total pengembalian ke titik waktu

mana pun, dari mana rangkaian harga dibuat ulang menggunakan fungsi EXP; klip layar ditunjukkan pada Gambar 3.1.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1														
2		Period No.	Prices at Period End		Returns Calculations	Formula		Cumulative Return	Formula	Comments		Prices Calculated using Additive Returns	Formula	
3		0	100.00							Formulae do NOT rely on prior period calculation of cumulative return		100.00		
4		1	105.55		5.40%	=LN(D4/D3)		5.40%	=SUM(F\$4:F4)	*		105.55	=M\$3*EXP(I4)	
5		2	97.28		-8.16%	=LN(D5/D4)		-2.76%	=SUM(F\$4:F5)	*		97.28	=M\$3*EXP(I5)	
6		3	94.95		-2.42%	=LN(D6/D5)		-5.18%	=SUM(F\$4:F6)	*		94.95	=M\$3*EXP(I6)	
7		4	94.01		-1.00%	=LN(D7/D6)		-6.18%	=SUM(F\$4:F7)	*		94.01	=M\$3*EXP(I7)	
8		5	95.99		2.09%	=LN(D8/D7)		-4.09%	=SUM(F\$4:F8)	*		95.99	=M\$3*EXP(I8)	
9		6	96.76		0.80%	=LN(D9/D8)		-3.29%	=SUM(F\$4:F9)	*		96.76	=M\$3*EXP(I9)	
10		7	96.92		0.16%	=LN(D10/D9)		-3.13%	=SUM(F\$4:F10)	*		96.92	=M\$3*EXP(I10)	
11		8	98.35		1.47%	=LN(D11/D10)		-1.67%	=SUM(F\$4:F11)	*		98.35	=M\$3*EXP(I11)	
12		9	99.49		1.16%	=LN(D12/D11)		-0.51%	=SUM(F\$4:F12)	*		99.49	=M\$3*EXP(I12)	

**Gambar 3.1 Pengembalian Periodik dan Harga Perkiraan Ulang Menggunakan LN dan EXP**

Dalam lembar kerja bernama OnePlus dari file yang sama, seseorang memulai dari kumpulan harga aset periodik yang sama, tetapi menghitung pengembalian periodik menggunakan pendekatan keuangan perusahaan tradisional. Seseorang melihat bahwa ketika bekerja mundur untuk menyusun kembali harga aset dari data tersebut, rumus untuk pengembalian kumulatif pada periode tertentu mengharuskan seseorang untuk mengetahui pengembalian kumulatif pada periode sebelumnya (bukan hanya angka pengembalian mentah, seperti dalam metode pengembalian logaritmik).

Dengan demikian, diperlukan langkah perhitungan tambahan (secara ekuivalen atau alternatif, seseorang harus mengetahui nilai aset akhir perkiraan ulang periode sebelumnya untuk memperkirakan ulang nilai periode saat ini menggunakan angka pengembalian periodik); ini ditunjukkan dalam klip layar Gambar 3.2.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1														
2		Period No.	Prices at Period End		Returns Calculations (Logarithmic)	Formula		Cumulative Return	Formula	Comments		Prices Calculated using Additive Returns	Formula	
3		0	100.00							Formulae DO rely on prior period calculation of cumulative return		100.00		
4		1	105.55		5.55%	=D4/D3-1		5.55%		*		105.55	=M\$3*(1+I4)	
5		2	97.28		-7.83%	=D5/D4-1		-2.72%	=(1+F5)*(1+I4)-1	*		97.28	=M\$3*(1+I5)	
6		3	94.95		-2.39%	=D6/D5-1		-5.05%	=(1+F6)*(1+I5)-1	*		94.95	=M\$3*(1+I6)	
7		4	94.01		-0.99%	=D7/D6-1		-5.99%	=(1+F7)*(1+I6)-1	*		94.01	=M\$3*(1+I7)	
8		5	95.99		2.11%	=D8/D7-1		-4.01%	=(1+F8)*(1+I7)-1	*		95.99	=M\$3*(1+I8)	
9		6	96.76		0.80%	=D9/D8-1		-3.24%	=(1+F9)*(1+I8)-1	*		96.76	=M\$3*(1+I9)	
10		7	96.92		0.16%	=D10/D9-1		-3.08%	=(1+F10)*(1+I9)-1	*		96.92	=M\$3*(1+I10)	
11		8	98.35		1.48%	=D11/D10-1		-1.65%	=(1+F11)*(1+I10)-1	*		98.35	=M\$3*(1+I11)	
12		9	99.49		1.16%	=D12/D11-1		-0.51%	=(1+F12)*(1+I11)-1	*		99.49	=M\$3*(1+I12)	

**Gambar 3.2 Pengembalian Periodik dan Harga Perkiraan Ulang Menggunakan Pendekatan Keuangan Korporat Tradisional**

Perhatikan juga hal berikut:

- Tentu saja, nilai aset perkiraan ulang sama dalam setiap kasus, jika seseorang menggunakan setiap pendekatan dengan benar dan konsisten. Selain itu, jika volatilitas perubahan periodik relatif kecil, maka profil pengembaliannya serupa, baik fungsi LN diterapkan pada rasio, maupun perubahannya diukur menggunakan pendekatan keuangan korporat tradisional.
- Saat menggunakan definisi pengembalian tradisional (non-logaritmik), perhitungan pengembalian untuk sub-periode langsung dari data pengembalian periodik (yaitu tanpa secara eksplisit memperkirakan nilai setiap periode) sebenarnya dapat dicapai dengan menggunakan fungsi FVCHEDULE, yang dibahas dalam Bab 4.

Karena perannya dalam kalkulasi yang melibatkan waktu berkelanjutan, fungsi EXP dan LN juga berguna dalam banyak bidang, seperti pemodelan pengoptimalan operasional, kerusakan, dan pemeliharaan.

### 3.2.2 Contoh: ABS dan SIGN

Fungsi ABS memberikan nilai absolut dari sebuah angka. Yaitu, ia mengembalikan angka ketika positif, dan negatif ketika negatif. Fungsi SIGN mengembalikan tanda angka (1 untuk angka positif, -1 untuk angka negatif, dan 0 untuk angka nol). Salah satu aplikasi pemodelan penting dari fungsi ABS adalah untuk membangun rumus pemeriksaan kesalahan yang kuat; rumus ini membandingkan nilai dari satu variabel, yang dihitung menggunakan dua (atau lebih) rute kalkulasi dalam sebuah model.

Dalam kebanyakan kasus, akan lebih mudah untuk menghitung selisih antara dua nilai, yang seharusnya selalu nol (bahkan ketika analisis sensitivitas dilakukan). Terutama ketika membangun banyak pemeriksaan kesalahan di berbagai tempat menjadi model besar, akan berguna untuk memiliki satu pemeriksaan global, yang merupakan jumlah dari semua pemeriksaan kesalahan individual (dan yang juga seharusnya selalu nol). Dalam kasus seperti itu, fungsi ABS dapat diterapkan pada setiap nilai pemeriksaan kesalahan individual, sebelum digunakan untuk menghitung nilai ringkasan global.

Penggunaan nilai absolut akan membantu menghindari kasus di mana kesalahan dalam perhitungan terperinci akan saling meniadakan pada tingkat global (karena beberapa bernilai negatif dan yang lainnya bernilai positif, seperti yang mungkin terjadi saat menangani neraca yang tidak seimbang dalam model laporan keuangan). Perhatikan juga bahwa fungsi ABS, SIGN, IF, MAX, MIN saling terkait dalam banyak hal. Misalnya:

$$\begin{aligned} \text{ABS}(N) &= \text{IF}(N \geq 0, N, -N) \\ \text{ABS}(N) &= \text{MAX}(N, 0) - \text{MIN}(N, 0) \\ N &= \text{SIGN}(N) * \text{ABS}(N) \end{aligned}$$

Gambar 3.3 memperlihatkan klip layar yang memperlihatkan bagian file tempat fungsi IF dan MAX/MIN dibandingkan dengan ABS. Pada Gambar 3.4, kami memperlihatkan klip layar bagian

lain dari file tersebut, yang menggunakan fungsi SIGN dan ABS untuk melakukan pemeriksaan galat individual dan menjumlahkannya menjadi total global (pemeriksaan galat dalam kasus ini adalah untuk memverifikasi bahwa rumus IF memberikan angka yang sama dengan pendekatan MAX-MIN).

	A	B	C	D	E	F	G	H	I	J	K
1											
2		Number		ABS	Formulae		IF	Formulae		MAX-MIN	Formulae
3		93.270		93.270	=ABS(B3)		93.270	=IF(B3>=0,B3,-B3)		93.270	=MAX(B3,0)-MIN(B3,0)
4		62.430		62.430	=ABS(B4)		62.430	=IF(B4>=0,B4,-B4)		62.430	=MAX(B4,0)-MIN(B4,0)
5		-83.200		83.200	=ABS(B5)		83.200	=IF(B5>=0,B5,-B5)		83.200	=MAX(B5,0)-MIN(B5,0)
6		-81.000		81.000	=ABS(B6)		81.000	=IF(B6>=0,B6,-B6)		81.000	=MAX(B6,0)-MIN(B6,0)
7		76.501		76.501	=ABS(B7)		76.501	=IF(B7>=0,B7,-B7)		76.501	=MAX(B7,0)-MIN(B7,0)
8		-4.326		4.326	=ABS(B8)		4.326	=IF(B8>=0,B8,-B8)		4.326	=MAX(B8,0)-MIN(B8,0)
9		-7.700		7.700	=ABS(B9)		7.700	=IF(B9>=0,B9,-B9)		7.700	=MAX(B9,0)-MIN(B9,0)
10		0.000		0.000	=ABS(B10)		0.000	=IF(B10>=0,B10,-B10)		0.000	=MAX(B10,0)-MIN(B10,0)
11											

Gambar 3.3 Menghitung Nilai Absolut Menggunakan ABS, IF, MAX, dan MIN

	L	M	N	O	P	Q	R	S	T
		SIGN	Formulae		ABS*SIGN	Formulae		Error checking	Formulae
		1	=SIGN(B3)		93.270	=ABS(B3)*SIGN(B3)		0.000	=ABS(G3-J3)
		1	=SIGN(B4)		62.430	=ABS(B4)*SIGN(B4)		0.000	=ABS(G4-J4)
		-1	=SIGN(B5)		-83.200	=ABS(B5)*SIGN(B5)		0.000	=ABS(G5-J5)
		-1	=SIGN(B6)		-81.000	=ABS(B6)*SIGN(B6)		0.000	=ABS(G6-J6)
		1	=SIGN(B7)		76.501	=ABS(B7)*SIGN(B7)		0.000	=ABS(G7-J7)
		-1	=SIGN(B8)		-4.326	=ABS(B8)*SIGN(B8)		0.000	=ABS(G8-J8)
		-1	=SIGN(B9)		-7.700	=ABS(B9)*SIGN(B9)		0.000	=ABS(G9-J9)
		0	=SIGN(B10)		0.000	=ABS(B10)*SIGN(B10)		0.000	=ABS(G10-J10)
								Global Error Check	
								0.000	=SUM(S3:S11)

Gambar 3.4 Menggunakan Fungsi SIGN, dan Pemeriksaan Kesalahan Global Menggunakan ABS

#### Contoh: INT, ROUNDDOWN, ROUNDUP, ROUND, dan TRUNC

Fungsi INT, ROUNDDOWN, ROUNDUP, dan ROUND menyediakan berbagai cara untuk melakukan pembulatan angka:

- Fungsi INT membulatkan ke bawah ke bilangan bulat terdekat, sehingga nilai masukan negatif (seperti -4,2) akan setidaknya sama kecilnya, atau bahkan lebih kecil lagi (seperti -5). Dengan kata lain, fungsi ini membulatkan ke kiri.
- Fungsi terkait lainnya termasuk ROUNDDOWN, ROUNDUP, ROUND, dan TRUNC. Masing-masing memiliki argumen yang sesuai dengan jumlah digit yang dibulatkan.
- ROUNDDOWN membulatkan ke arah nol: angka masukan positif akan menghasilkan

angka yang sama atau lebih kecil, sedangkan angka masukan negatif akan menghasilkan angka yang setidaknya sama besarnya (yaitu tidak lebih kecil dari masukannya); dengan kata lain, pembulatan mendekati nol. Sebaliknya, ROUNDUP membulatkan menjauhi nol. TRUNC dan ROUNDDOWN adalah fungsi yang sangat mirip; namun, jumlah digit merupakan argumen yang diperlukan dalam ROUNDDOWN, tetapi bersifat opsional dalam TRUNC (dan sama dengan nol jika dihilangkan).

- ROUND membulatkan angka sesuai dengan jumlah digit yang diperlukan, sehingga arah pembulatan bergantung pada apakah angka tersebut positif atau negatif, serta nilai angka itu sendiri (misalnya -4,324 dibulatkan pada dua digit menjadi -4,32, yaitu ke atas, sedangkan -4,326 dibulatkan menjadi -4,33, yaitu ke bawah.)

Gambar 3.5 menunjukkan klip layar. Ada banyak kemungkinan penggunaan fungsi-fungsi ini dalam pemodelan keuangan. Salah satunya adalah dengan menggunakan nilai numerik untuk satu bulan (misalnya April adalah bulan ke-4) untuk menghitung kuartal yang relevan (kuartal 1 adalah Januari hingga Maret, kuartal 2 adalah April hingga Juni, dan seterusnya).

	A	B	C	D	E	F	G
1							
2		Number, #	INT	ROUNDDOWN(#,0)	ROUNDUP(#,0)	ROUND(#,2)	TRUNC(#,0)
3		98.730	98.000	98.000	99.000	98.730	98.000
4		76.501	76.000	76.000	77.000	76.500	76.000
5		63.326	63.000	63.000	64.000	63.330	63.000
6		-4.324	-5.000	-4.000	-5.000	-4.320	-4.000
7		-4.326	-5.000	-4.000	-5.000	-4.330	-4.000
8		-7.700	-8.000	-7.000	-8.000	-7.700	-7.000

**Gambar 3.5 Contoh Fungsi INT dan Berbagai Fungsi Tipe ROUND untuk Input Positif dan Negatif**

Semua fungsi ini dapat digunakan untuk mengubah angka bulan menjadi kuartal (tabel pencarian juga dapat digunakan). Gambar 3.6 menunjukkan tangkapan layar.

	A	B	C	D	E	F	G	H	I	J
1										
2		Calculations of Quarter								
3		Month Number	INT	Formula	Using ROUNDDOWN	Formula	Using ROUNDUP	Formula	Using ROUND	Formula
4		1	1	=INT((B4-1)/3)+1	1	=ROUNDDOWN((B4-1)/3,0)+1	1	=ROUNDUP((B4-3)/3+1,0)	1	=ROUND((B4+1)/3,0)
5		2	1	=INT((B5-1)/3)+1	1	=ROUNDDOWN((B5-1)/3,0)+1	1	=ROUNDUP((B5-3)/3+1,0)	1	=ROUND((B5+1)/3,0)
6		3	1	=INT((B6-1)/3)+1	1	=ROUNDDOWN((B6-1)/3,0)+1	1	=ROUNDUP((B6-3)/3+1,0)	1	=ROUND((B6+1)/3,0)
7		4	2	=INT((B7-1)/3)+1	2	=ROUNDDOWN((B7-1)/3,0)+1	2	=ROUNDUP((B7-3)/3+1,0)	2	=ROUND((B7+1)/3,0)
8		5	2	=INT((B8-1)/3)+1	2	=ROUNDDOWN((B8-1)/3,0)+1	2	=ROUNDUP((B8-3)/3+1,0)	2	=ROUND((B8+1)/3,0)
9		6	2	=INT((B9-1)/3)+1	2	=ROUNDDOWN((B9-1)/3,0)+1	2	=ROUNDUP((B9-3)/3+1,0)	2	=ROUND((B9+1)/3,0)
10		7	3	=INT((B10-1)/3)+1	3	=ROUNDDOWN((B10-1)/3,0)+1	3	=ROUNDUP((B10-3)/3+1,0)	3	=ROUND((B10+1)/3,0)
11		8	3	=INT((B11-1)/3)+1	3	=ROUNDDOWN((B11-1)/3,0)+1	3	=ROUNDUP((B11-3)/3+1,0)	3	=ROUND((B11+1)/3,0)
12		9	3	=INT((B12-1)/3)+1	3	=ROUNDDOWN((B12-1)/3,0)+1	3	=ROUNDUP((B12-3)/3+1,0)	3	=ROUND((B12+1)/3,0)
13		10	4	=INT((B13-1)/3)+1	4	=ROUNDDOWN((B13-1)/3,0)+1	4	=ROUNDUP((B13-3)/3+1,0)	4	=ROUND((B13+1)/3,0)
14		11	4	=INT((B14-1)/3)+1	4	=ROUNDDOWN((B14-1)/3,0)+1	4	=ROUNDUP((B14-3)/3+1,0)	4	=ROUND((B14+1)/3,0)
15		12	4	=INT((B15-1)/3)+1	4	=ROUNDDOWN((B15-1)/3,0)+1	4	=ROUNDUP((B15-3)/3+1,0)	4	=ROUND((B15+1)/3,0)

**Gambar 3.6 Contoh Penggunaan Fungsi INT dan Berbagai Fungsi Tipe ROUND untuk Menghitung Kuartal**

Rumusnya masing-masing adalah:

Kuartal menggunakan  $\text{INT}(\text{INT}((\text{Bulan } 1) / 3) + 1)$   
 Kuartal menggunakan  $\text{ROUND}(\text{ROUND}((\text{Bulan } 1) / 3, 0) + 1)$   
 Kuartal menggunakan  $\text{ROUND}(\text{ROUND}((\text{Bulan } 3) / 3, 1, 0))$   
 Kuartal menggunakan  $\text{ROUND}(\text{ROUND}((\text{Bulan } 1) / 3, 0))$ .

Nomor bulan dapat ditentukan dari tanggal, menggunakan fungsi MONTH (Bab 23). Perhatikan juga bahwa:

- Periode setengah tahun (paruh pertama dan kedua) juga dapat dibuat dengan membaginya dengan 6, bukan dengan 3.
- Jika seseorang ingin mengubah definisi kuartal pertama tahun tersebut (misalnya kuartal pertama dimulai pada bulan April, seperti dalam tahun keuangan Inggris), seseorang dapat mengadaptasi rumus di atas dengan tepat.

### 3.2.3 Contoh: MROUND, CEILING.MATH, dan FLOOR.MATH

Fungsi MROUND membulatkan input ke nilai terdekat dari kelipatan yang dipilih. Misalnya, perusahaan dapat menghitung bonus tahunan untuk setiap karyawan menggunakan rumus pasti yang diperoleh dari berbagai metrik input, tetapi menentukan nilai akhir yang akan dibayarkan dengan membulatkannya ke angka Rp.100.000 terdekat (misalnya, sehingga perhitungan awal pasti hak bonus sebesar, katakanlah, Rp.136.523 akan menjadi Rp.140.000).

Perhatikan bahwa, pada saat penulisan, fungsi tersebut hanya berfungsi jika angka input dan kelipatan memiliki tanda yang sama. Kita dapat membuat solusi untuk ini, menggunakan ABS untuk mengubah angka dan kelipatan menjadi angka positif, dan menggunakan fungsi SIGN untuk mengembalikan hasil yang memiliki tanda yang sama dengan input numerik asli; lihat klip layar pada Gambar 3.7. Fungsi CEILING.MATH (ekstensi dari fungsi CEILING lama) membulatkan angka ke atas hingga kelipatan terdekat yang dipilih (pembulatan ke atas ini berbeda dengan MROUND, yang membulatkan ke arah mana pun). Fungsi ini memiliki parameter opsional tambahan yang memungkinkan pembulatan menjauhi nol (sehingga untuk angka negatif, proses pembulatan ke atas sebenarnya membuat hasilnya tidak lebih besar dari angka asli). Serupa dengan itu, fungsi FLOOR.MATH (dan fungsi FLOOR lamanya) membulatkan angka ke bawah hingga kelipatan terdekat yang dipilih.

	A	B	C	D	E	F	G	H	I
1									
2		Number, #	Multiple		MROUND(#, 5)	Formulae		Adaptation	Formulae
3		93.270	5		95	=MROUND(B3, C3)		95	=MROUND(ABS(B3), ABS(C3))*SIGN(B3)
4		62.430	5		60	=MROUND(B4, C4)		60	=MROUND(ABS(B4), ABS(C4))*SIGN(B4)
5		-83.200	-5		-85	=MROUND(B5, C5)		-85	=MROUND(ABS(B5), ABS(C5))*SIGN(B5)
6		-81.000	-5		-80	=MROUND(B6, C6)		-80	=MROUND(ABS(B6), ABS(C6))*SIGN(B6)
7		76.501	-5		#NUM!	=MROUND(B7, C7)		75	=MROUND(ABS(B7), ABS(C7))*SIGN(B7)
8		-4.326	5		#NUM!	=MROUND(B8, C8)		-5	=MROUND(ABS(B8), ABS(C8))*SIGN(B8)
9		-7.700	5		#NUM!	=MROUND(B9, C9)		-10	=MROUND(ABS(B9), ABS(C9))*SIGN(B9)

**Gambar 19.7** Menggunakan MROUND, ABS, dan SIGN untuk Pembulatan

	A	B	C	D	E	F	G	H	I
1									
2		Number, #	Multiple		MROUND(#, 5)	Adapted MROUND		CEILING.MATH(#,multiple,0)	CEILING.MATH(#,multiple,1)
3		93.270	5		95		95	95	95
4		62.430	5		60		60	65	65
5		-83.200	-5		-85		-85	-80	-85
6		-81.000	-5		-80		-80	-80	-85
7		76.501	-5		#NUM!		75	80	80
8		-4.326	5		#NUM!		-5	0	-5
9		-7.700	5		#NUM!		-10	-5	-10
10									

Gambar 19.8 Contoh Fungsi CELING.MATH

### 3.2.4 Contoh: MOD

Fungsi MOD menyediakan modulus suatu angka dengan pembagi tertentu (yaitu, sisa ketika pembagi dikurangi berulang kali dari angka tersebut hingga tidak ada lagi pengurangan penuh yang mungkin dilakukan). Misalnya, dengan menggunakan pembagi lima, modulus angka 1 adalah 1, angka 2 adalah 2, sedangkan modulus angka 5 adalah 0, angka 6 adalah 1, angka 7 adalah 2, dan angka 10 adalah nol, dan seterusnya.

Salah satu penerapannya adalah alokasi item ke dalam kelompok, di mana seseorang menelusuri kelompok yang tersedia secara berurutan (lebih khusus lagi, ini dapat berkaitan dengan alokasi pasien ke uji klinis, pelanggan ke pusat layanan, dan seterusnya). Berkas Ch19.7.MOD.xlsx menunjukkan contoh di mana terdapat lima kelompok (jumlah kelompok bertindak sebagai pembagi) dan item dialokasikan ke kelompok-kelompok ini berdasarkan nomor kedatangan yang diurutkan (atau waktu).

	A	B	C	D
1				
2		# Groups to Allocate To		5
3				
4		Item Number	Group Allocation	Formula
5			1	=MOD(B5-1,C\$2)+1
6			2	=MOD(B6-1,C\$2)+1
7			3	=MOD(B7-1,C\$2)+1
8			4	=MOD(B8-1,C\$2)+1
9			5	=MOD(B9-1,C\$2)+1
10			1	=MOD(B10-1,C\$2)+1
11			2	=MOD(B11-1,C\$2)+1
12			3	=MOD(B12-1,C\$2)+1
13			4	=MOD(B13-1,C\$2)+1
14			10	=MOD(B14-1,C\$2)+1
15			11	=MOD(B15-1,C\$2)+1
16			12	=MOD(B16-1,C\$2)+1
17			13	=MOD(B17-1,C\$2)+1
18			14	=MOD(B18-1,C\$2)+1
19			15	=MOD(B19-1,C\$2)+1
20			16	=MOD(B20-1,C\$2)+1
21			17	=MOD(B21-1,C\$2)+1
22			18	=MOD(B22-1,C\$2)+1

Gambar 3.9 Menggunakan MOD untuk Menetapkan Item ke Kelompok

Karena fungsi MOD mengembalikan nol saat angka tersebut merupakan kelipatan pembagi, rumus telah disesuaikan untuk mencerminkan bahwa angka kelompok adalah 1,2,3,4,5 (yaitu tidak ada angka kelompok 0) dengan pengurangan dan penambahan angka 1 pada langkah perhitungan yang sesuai; cuplikan layar ditunjukkan pada Gambar 3.9.

### 3.2.5 Contoh: SQRT dan POWER

Di Excel, seseorang dapat menghitung pangkat dan akar angka baik menggunakan operasi aritmatika langsung atau menggunakan fungsi POWER. Jika ingin menghitung akar kuadrat, seseorang juga dapat menggunakan fungsi SQRT. Salah satu keuntungan (yang dapat diperdebatkan) dari fungsi POWER dibandingkan pendekatan aritmatika umum adalah bahwa rumus tersebut cukup mudah dibaca dalam beberapa kasus; cuplikan layar ditunjukkan pada Gambar 3.10.

	A	B	C	D
1				
2		<b>Example</b>	<b>Formulae</b>	
3		729	=POWER(9,3)	
4		9	=729^(1/3)	
5		9	=POWER(729,1/3)	
6		1.414	=2^(1/2)	
7		1.414	=POWER(2,0.5)	
8		1.414	=SQRT(2)	

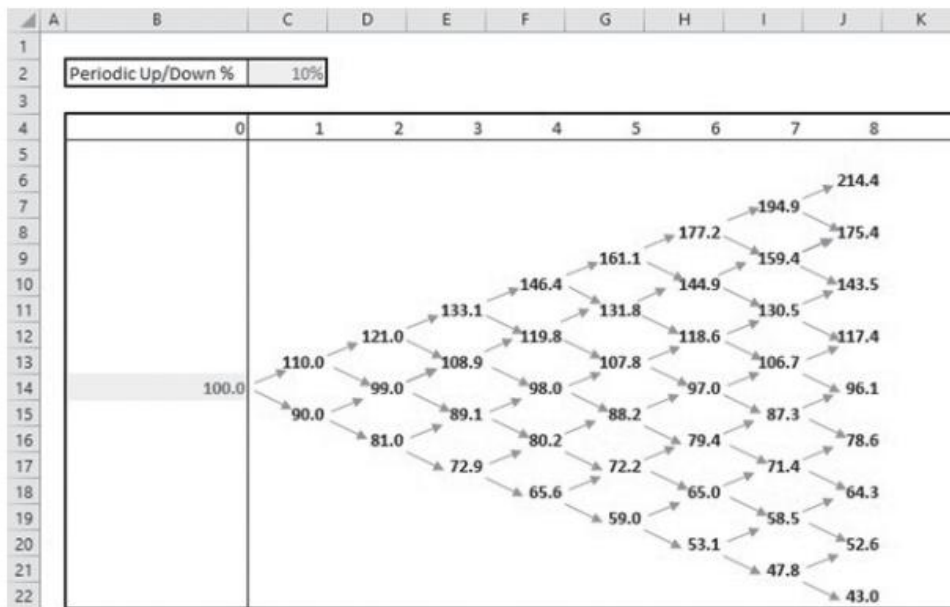
**Gambar 3.10 Contoh Fungsi POWER dan SQRT serta Perbandingan dengan Aritmatika Langsung**

### 3.2.6 Contoh: FACT dan COMBIN

Fungsi COMBIN dan FACT berguna dalam beberapa konteks probabilistik dan statistik. Misalnya, jika nilai aset, atau harga komoditas, dimodelkan sebagai proses yang tidak pasti di mana harga berubah setiap periode baik dalam persentase pergerakan naik atau turun (masing-masing dengan probabilitas tertentu), seseorang dapat menggunakan fungsi ini untuk menghitung jumlah kemungkinan cara mencapai setiap titik di pohon (serta kemungkinan hasil tersebut).

Gambar 3.11 menunjukkan klip layar yang diambil dari lembar kerja "RecombiningTreeSimple", di mana diasumsikan ada kemungkinan pergerakan naik atau turun sebesar 10%. Sebagai referensi tambahan, dalam berkas yang sama, lembar kerja "RecombiningTreeLN" (tidak ditampilkan dalam klip layar) berisi pohon serupa di mana perubahan harga 10% adalah perubahan logaritmik, sehingga nilai naik dan turun diskalakan oleh EXP(10%) dan EXP(-10%) di setiap periode.





**Gambar 3.11** Pohon Rekombinasi Menggunakan Perubahan Harga Periodik Absolut Sederhana Sebesar 10%

Gambar 3.12 menunjukkan klip layar yang diambil dari berkas di mana fungsi COMBIN digunakan untuk menghitung jumlah cara untuk mencapai (pada akhir periode 8) masing-masing dari sembilan kemungkinan hasil. Rumus untuk ini juga dapat diturunkan menggunakan fungsi FACT (yang menghitung faktorial suatu angka, yaitu hasil perkalian semua bilangan bulat dari 1 hingga angka ini).

8	# Up	# Down	# Paths	Formula	Using FACT	Formula
214.4	8	0	1	=COMBIN(8,L6)	1	=FACT(8)/(FACT(L6)*FACT(M6))
175.4	7	1	8	=COMBIN(8,L8)	8	=FACT(8)/(FACT(L8)*FACT(M8))
143.5	6	2	28	=COMBIN(8,L10)	28	=FACT(8)/(FACT(L10)*FACT(M10))
117.4	5	3	56	=COMBIN(8,L12)	56	=FACT(8)/(FACT(L12)*FACT(M12))
96.1	4	4	70	=COMBIN(8,L14)	70	=FACT(8)/(FACT(L14)*FACT(M14))
78.6	3	5	56	=COMBIN(8,L16)	56	=FACT(8)/(FACT(L16)*FACT(M16))
64.3	2	6	28	=COMBIN(8,L18)	28	=FACT(8)/(FACT(L18)*FACT(M18))
52.6	1	7	8	=COMBIN(8,L20)	8	=FACT(8)/(FACT(L20)*FACT(M20))
43.0	0	8	1	=COMBIN(8,L22)	1	=FACT(8)/(FACT(L22)*FACT(M22))

**Gambar 3.12** Jumlah Kemungkinan untuk Mencapai Setiap Hasil di Pohon Binomial

Frekuensi aktual (probabilitas atau kemungkinan) dari setiap hasil akan dibentuk dengan

mengalikan jumlah total hasil yang diwakili oleh setiap keadaan dengan probabilitas setiap jenis hasil. Jika  $p$  adalah probabilitas pergerakan ke atas, maka probabilitas terjadinya jalur tertentu yang mengarah ke  $i$  hasil adalah  $p^i(1 - p)^{8-i}$ , dengan jumlah jalur tersebut diberikan oleh fungsi COMBIN.

Jadi, probabilitas  $i$  pergerakan ke atas secara total adalah:

$$p(i) = p^i(1 - p)^{8-i} \text{COMBIN}(8, i)$$

(dengan adaptasi yang diperlukan untuk pohon dengan periode selain delapan periode yang cukup jelas). Hal ini ditunjukkan pada Gambar 3.13 untuk asumsi pergerakan probabilitas ke atas sebesar 60%.

**3.2.7 Contoh: RAND()**

Fungsi RAND() menghasilkan sampel acak antara 0 dan 1 dari distribusi kontinu yang seragam. Fungsi ini akan mengambil sampel ulang setiap kali F9 ditekan, yaitu setiap kali buku kerja dihitung ulang. (Perhatikan bahwa RANDBETWEEN hanya menyediakan sampel nilai integer, bukan rentang kontinu.) Dengan memperlakukan nilai acak seolah-olah merupakan probabilitas, seseorang dapat memperoleh sampel acak dari distribusi lain. Untuk rentang kontinu seragam:

$$\text{Sampel Min (maks Min)} * \text{RAND()}$$

Untuk terjadinya risiko peristiwa (proses Bernoulli):  
Kejadian atau Tidak JIKA(RAND() Prob,1,0)

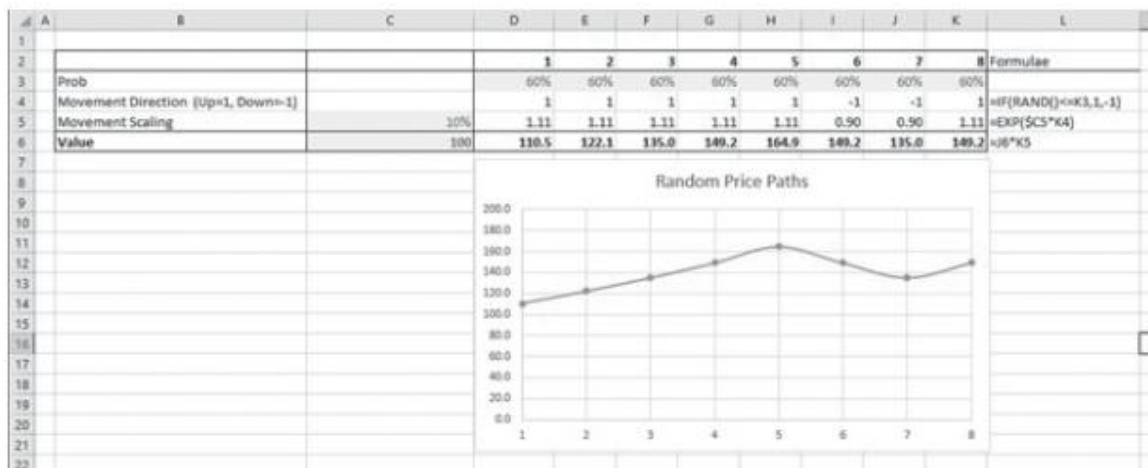
J	K	L	M	N	O	P	Q	R	S	T	U
										60%	p
										40%	=1-T2
										Direct Calc	Formula
8	# Up	# Down	# Paths	Formula	Using FACT	Formula					
214.4	8	0	1	=COMBIN(8,L6)	1	=FACT(8)/(FACT(L6)*FACT(M6))				1.68%	=(ST\$2^L6)*(ST\$3^M6)*N6
175.4	7	1	8	=COMBIN(8,L8)	8	=FACT(8)/(FACT(L8)*FACT(M8))				8.96%	=(ST\$2^L8)*(ST\$3^M8)*N8
143.5	6	2	28	=COMBIN(8,L10)	28	=FACT(8)/(FACT(L10)*FACT(M10))				20.90%	=(ST\$2^L10)*(ST\$3^M10)*N10
117.4	5	3	56	=COMBIN(8,L12)	56	=FACT(8)/(FACT(L12)*FACT(M12))				27.87%	=(ST\$2^L12)*(ST\$3^M12)*N12
96.1	4	4	70	=COMBIN(8,L14)	70	=FACT(8)/(FACT(L14)*FACT(M14))				23.22%	=(ST\$2^L14)*(ST\$3^M14)*N14
78.6	3	5	56	=COMBIN(8,L16)	56	=FACT(8)/(FACT(L16)*FACT(M16))				12.39%	=(ST\$2^L16)*(ST\$3^M16)*N16
64.3	2	6	28	=COMBIN(8,L18)	28	=FACT(8)/(FACT(L18)*FACT(M18))				4.13%	=(ST\$2^L18)*(ST\$3^M18)*N18
52.6	1	7	8	=COMBIN(8,L20)	8	=FACT(8)/(FACT(L20)*FACT(M20))				0.79%	=(ST\$2^L20)*(ST\$3^M20)*N20
43.0	0	8	1	=COMBIN(8,L22)	1	=FACT(8)/(FACT(L22)*FACT(M22))				0.07%	=(ST\$2^L22)*(ST\$3^M22)*N22
										100.00%	

Gambar 3.13 Probabilitas Akhir Setiap Keadaan Akhir

Sampel dari banyak distribusi lain juga dapat dibuat dengan menggunakan fungsi RAND() sebagai input ke dalam inversi distribusi kumulatif untuk fungsi tersebut (beberapa contoh pengambilan sampel lebih lanjut ditunjukkan kemudian dalam buku ini, dan Risiko Bisnis dan Pemodelan Simulasi dalam Praktik penulis memberikan lebih banyak detail tentang hal ini).

fungsi RAND() digunakan dalam setiap periode model untuk menentukan apakah harga aset bergerak naik atau turun, dengan asumsi probabilitas 60% dari pergerakan ke atas. Persentase perubahan harga yang mungkin (10%) setiap periode diinterpretasikan di sini sebagai angka logaritmik, dengan harga setiap periode dihitung dari harga periode sebelumnya tergantung pada hasil fungsi RAND() dalam periode tersebut.

Oleh karena itu, lintasan yang ditunjukkan dalam model ini pada dasarnya sesuai dengan lintasan individual yang ada di pohon rekombinasi pada contoh sebelumnya; Gambar 3.14 menunjukkan klip layar dari berkas tersebut. Dalam praktiknya, penentuan, dan perekaman, semua skenario yang mungkin biasanya memerlukan makro VBA, untuk menghitung ulang buku kerja secara berulang dan menyimpan hasilnya (lihat Bab 2).



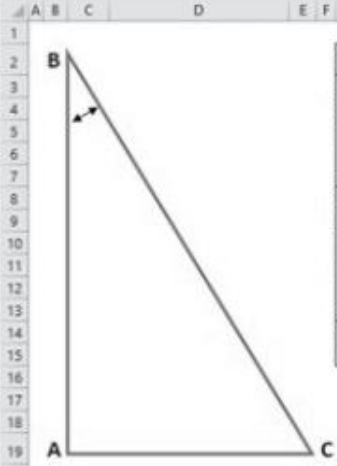
**Gambar 3.14** Kemungkinan Jalur Harga Acak Menggunakan RAND()

### 3.2.8 Contoh: SINE, ASIN, DEGREES, dan PI()

Ada serangkaian fungsi trigonometri dan fungsi serupa yang terkadang berguna dalam pemodelan keuangan. Misalnya, ketika mempertimbangkan pengeluaran modal yang diperlukan untuk membangun stasiun relai telepon seluler dengan jangkauan yang memadai di area tertentu, seseorang mungkin perlu dapat menghitung sudut yang berkaitan dengan perambatan radiasi elektromagnetik antara stasiun dan pusat populasi. Dengan demikian, fungsi yang menghitung panjang sisi segitiga siku-siku jika diketahui sudut tertentu lainnya (misalnya sinus, kosinus, atau tangen) dapat berguna, seperti halnya fungsi yang menghitung sudut segitiga jika panjang sisinya diketahui (misalnya fungsi invers).

Gambar 3.15 menunjukkan contoh fungsi SINE, ASIN, COS, ACOS, TAN, dan ATAN untuk melakukan perhitungan tersebut. Gambar 3.15 menunjukkan cuplikan layar berkas tersebut.

Kita dapat melihat bahwa fungsi ASIN digunakan untuk menghitung sudut ABC kira-kira sebesar 26,57 derajat (misalnya, jika segitiga siku-siku memiliki panjang sisi 10 dan 20, kedua sudut yang tidak siku-siku kira-kira sebesar 26,57 dan 63,43 derajat).



Find Angle ABC, given:		Input/Result	Formulae	Comments
AB		20		
AC		10		
BC (Hypotenuse)		22.36	=SQRT(H3^2+H4^2)	Pythagoras' theorem!
Angle Calculations		Result	Formulae	
Ratio (AC/BC)		0.45	=H4/H5	SINE of angle equals this, by definition
In Radians		0.46	=ASIN(H8)	
In Degrees		26.57	=H9*180/PI()	Converts radians to degrees
In Degrees		26.57	=DEGREES(H9)	Converts radians to degrees
Using Other Functions		Result	Formulae	
(Inverse) Cosine		26.57	=DEGREES(ACOS(H3/H5))	COSINE is opposite/hypotenuse
(Inverse) Tangent		26.57	=DEGREES(ATAN(H4/H3))	COSINE is opposite/adjacent

Gambar 3.15 Menggunakan ASIN untuk Menemukan Sudut dalam Segitiga

Secara default, Excel bekerja dalam radian; konversi ke derajat dapat dilakukan menggunakan fungsi DEGREES, atau langsung dengan mengalikannya dengan 180. Kita dapat menggunakan fungsi PI() (akurat hingga sekitar 15 tempat desimal) daripada harus menyetik  $\pi$  (3.141592653589793).

	A	B	C	D
1				
2		<b>Roman Number</b>	<b>Result</b>	<b>Formulae</b>
3		MCMLXXXIV	1984	=ARABIC(B3)
4				
5		<b>Arabic Number</b>	<b>Result</b>	<b>Formulae</b>
6		1984	MCMLXXXIV	=ROMAN(B6)

Gambar 3.17 Konversi Antara Ekuivalen Arab dan Romawi

Fungsi SQRTPI() memberikan akar kuadrat dari  $\pi$  sebagai angka yang telah dihitung sebelumnya (yang terkadang diperlukan dalam aplikasi statistik dan bidang terkait). Perhatikan juga bahwa fungsi inverslah yang penting dalam contoh ini (untuk mengonversi dari panjang ke sudut); jika seseorang mengetahui fungsi SINE, tetapi tidak mengetahui ASIN, maka prosedur inversi yang serupa dapat disiapkan menggunakan GoalSeek; namun, ini kurang efisien dibandingkan menggunakan prosedur fungsi bawaan jika ada.

### 3.2.9 Contoh: BASIS dan DESIMAL

Kadang-kadang seseorang mungkin perlu mengonversi angka dari satu basis (atau

"radix") ke basis lainnya. Ini dapat muncul dalam beberapa konteks manipulasi data, di mana data telah muncul dari sumber terkomputerisasi atau biner lainnya. Fungsi BASE dan DECIMAL memungkinkan konversi antar basis angka, dengan demikian angka nondesimal dinyatakan sebagai bidang teks. Berkas Ch19.12.BASE.DECIMAL.xlsx berisi contoh, yang ditunjukkan dalam klip layar pada Gambar 19.16. Sesekali (sekali lagi, kemungkinan besar dalam bidang manipulasi teks atau data) seseorang mungkin perlu mengonversi angka Romawi (teks) ke Arab dan sebaliknya. Berkas Ch19.13.ARABIC.ROMAN.xlsx berisi contoh penggunaan fungsi ARABIC dan ROMAN, dan ditunjukkan dalam klip layar pada Gambar 19.17.

## BAB 4

### FUNGSI KEUANGAN

#### 4.1 PENDAHULUAN

Bab ini berfokus pada penyediaan contoh-contoh dari serangkaian fungsi yang luas dalam kategori Keuangan Excel, yang berkaitan dengan:

- Perhitungan anuitas dan pertumbuhan umum atau diskonto.
- Perhitungan hipotek dan pembayaran kembali modal.
- Penilaian dan analisis investasi umum.
- Perhitungan penyusutan.
- Hasil, obligasi, sekuritas umum, dan perhitungan keuangan khusus.

#### 4.2 APLIKASI PRAKTIS

Beberapa poin yang perlu diingat:

- Banyak fungsi yang mengembalikan angka negatif secara default, terutama yang berkaitan dengan perhitungan pembayaran yang diperlukan atau arus kas keluar. Seseorang harus sangat berhati-hati dalam penggunaannya, terutama ketika fungsi tersebut tertanam dalam pernyataan logis seperti yang melibatkan rumus IF, MIN, atau MAX.
- Penggunaan pernyataan IF atau IFERROR di sekitar fungsi tersebut sering kali diperlukan. Misalnya, beberapa fungsi akan mengembalikan nilai kesalahan untuk jangka waktu yang berada di luar batas yang valid (seperti masa berlaku hipotek). Seperti yang dibahas dalam Bab 1, secara umum lebih baik untuk mengecualikan kondisi tertentu di mana nilai kesalahan diketahui terjadi tetapi valid (dengan menggunakan pernyataan IF yang diterapkan pada situasi khusus ini), daripada mengesampingkan semua kesalahan dengan menggunakan IFERROR.
- Banyak fungsi diformat dalam mata uang lokal (yaitu versi Excel yang digunakan), yang terkadang merepotkan, terutama saat mengerjakan proyek internasional; seseorang mungkin merasa lebih nyaman untuk beralih ke format umum.
- Jika fungsi mengharuskan penggunaan suku bunga (atau pertumbuhan atau tingkat pengembalian), seseorang perlu memastikan bahwa suku bunga yang digunakan konsisten dengan periode model (misalnya tahunan, triwulanan, atau bulanan).

##### 4.2.1 Contoh: FVSCHEDULE

FVSCHEDULE memberikan nilai masa depan dari suatu kuantitas setelah menerapkan serangkaian tingkat pertumbuhan padanya. Input yang diperlukan untuk fungsi tersebut adalah:

- Nilai awal (atau "pokok").
- Jadwal pertumbuhan, pengembalian, atau tingkat bunga.

Gambar 4.1 menunjukkan contoh berbagai aplikasi fungsi tersebut, baik untuk situasi dengan tingkat pertumbuhan konstan maupun tingkat pertumbuhan yang bervariasi seiring waktu (lihat Gambar 4.1). Salah satu aplikasinya adalah untuk menghitung tingkat pertumbuhan gabungan total (dan rata-rata) untuk model atau kumpulan data multiperiodik, di mana tingkat pada setiap periode berbeda, dan diukur menggunakan metode keuangan perusahaan tradisional (lihat Bab 3).

	A	B	C	D	E	F	G	H	I	J	K	L
1												
2	Using the FVSCHEDULE Function											
3												
4				1	2	3	4	5				
5	Principal	100								Result	Formula	
6	Schedule 1		5.0%	5.0%	5.0%	5.0%	5.0%			127.6	=FVSCHEDULE(\$C\$5,D6:H6)	
7	Schedule 2		0.0%	0.0%	0.0%	0.0%	0.0%			100.0	=FVSCHEDULE(\$C\$5,D7:H7)	
8	Schedule 3		2.0%	3.0%	4.0%	5.0%	6.0%			121.6	=FVSCHEDULE(\$C\$5,D8:H8)	
9												
10	EXPLICIT CALCS: For Schedule 3											
11												
12	Cumulative Rate		102.0%	105.1%	109.3%	114.7%	121.6%					
13	Periodic Value		102.0	105.1	109.3	114.7	121.6					

**Gambar 4.1 Contoh Fungsi FVSCHEDULE**

Dengan menetapkan nilai asli (pokok) ke 1, fungsi tersebut mengembalikan hasil perkalian "1 ditambah tingkat pertumbuhan" di setiap periode, sehingga pengurangan 1 dari hasil perkalian ini menghasilkan pertumbuhan total, dan pengambilan akar yang sesuai menghasilkan angka gabungan tahunan rata-rata (misalnya akar kuadrat untuk rata-rata tahunan selama periode dua tahun). Konteksnya adalah bahwa seseorang diberikan serangkaian angka pengembalian tradisional atau tingkat pertumbuhan dan ingin menghitung pengembalian rata-rata (atau tingkat pertumbuhan) antara dua periode secara langsung dari data ini (tanpa harus secara eksplisit menghitung nilai aset aktual di setiap periode, atau tingkat pertumbuhan gabungan kumulatif eksplisit untuk setiap periode).

Gambar 4.2 menunjukkan cuplikan layar dari perhitungan pengembalian total antara periode 5 dan 15 secara langsung dari data pengembalian. Di sisi kanan berkas (di sebelah kanan dari Kolom I, tidak ditampilkan dalam cuplikan layar) terdapat perhitungan eksplisit yang seharusnya diperlukan.

	A	B	C	D	E	F	G	H
1								
2		Period No.				Using FVCHEDULE to Calculate Return Between Periods 5 and 15		
3		0	Returns		Total	Average		
4		1	5.55%		12.41%	1.18%		
5		2	-7.83%		=FVSCHEDULE(1,D9:D18)-1	=(FVSCHEDULE(1,D9:D18))^(1/10)-1		
6		3	-2.39%					
7		4	-0.99%					
8		5	2.11%					
9		6	0.80%					
10		7	0.16%					
11		8	1.48%					
12		9	1.16%					
13		10	1.24%					
14		11	1.44%					
15		12	2.54%					
16		13	1.52%					
17		14	-0.85%					
18		15	2.32%					

**Gambar 4.2 Penggunaan FVSCHEDULE untuk Menghitung Tingkat Pertumbuhan Total dan Rata-rata**

#### 4.2.2 Contoh: FV dan PV

Fungsi FV menghitung nilai total dari serangkaian pembayaran tunai yang tumbuh secara konstan selama periode waktu terbatas, berdasarkan pembayaran awal. Fungsi ini juga memiliki parameter opsional yang memungkinkan nilai sekarang dari pembayaran sekaligus di masa mendatang untuk disertakan, dan untuk memperlakukan arus kas sebagai sesuatu yang terjadi di awal (bukan di akhir) setiap periode. Fungsi PV serupa, tetapi mendiskontokan pembayaran berkala (bukan menumbuhkannya).

	A	B	C	D	E	F	G	H	I	J	K
1											
2		Using the FV Function									
3											
4		Ex 1	Ex 2		Ex 3			Ex 4	Ex 5		
5		Rate	5.0%	5.0%	5.0%	5.0%		5.0%	5.0%		
6		Nper	1	10	10	10		1	10		
7		Pmt	-100	-100	-100	-100		-100	-100		
8		[pv]			500			0	0		
9		[type]						1	1		
10		Result	100	1258		441		105	1321		
11		Formula	=FV(C5,C6,C7)	=FV(D5,D6,D7)		=FV(F5,F6,F7,F8)		=FV(I5,I6,I7,I8,I9)	=FV(K5,K6,K7,K8,K9)		
12											
13		EXPLICIT CALCS									
14											
15		Inflated	Cumulative		Future Payment	Inflated Future Payment	Total	Discounted/Inflated	Cumulative		
16		1	100	100				105	105		
17		2	205	205				110	215		
18		3	313	313				116	331		
19		4	431	431				122	453		
20		5	553	553				128	580		
21		6	680	680				134	714		
22		7	814	814				141	855		
23		8	955	955				148	1003		
24		9	1103	1103				155	1158		
25		10	1258	1258				163	1321		
26			=C5*(1+C5)^(\$B25-1)	=SUM(C5:K25)	=F8	=F25*(1+C5)^\$B25	=D25+G25	=C5*(1+C5)^\$B25	=SUM(J5:K25)		

**Gambar 4.3 Contoh Fungsi FV**



Gambar 4.3 dan gambar 4.4 menunjukkan contoh (setiap fungsi ditampilkan pada lembar kerja terpisah, dengan nama yang sesuai); lihat juga cuplikan layar pada Gambar 4.3 dan Gambar 4.4. File tersebut juga menunjukkan hasil yang setara ketika perhitungan dilakukan secara eksplisit di Excel.

The screenshot shows an Excel spreadsheet with the following content:

	B	C	D	E	F	G	H	I	J	K
1										
2	Using the PV Function									
3										
4		Ex 1	Ex 2		Ex 3				Ex 4	Ex 5
5	Rate		3.0%	5.0%		5.0%			5.0%	5.0%
6	Nper		1	10		10			1	10
7	Pmt		-100	-100		-100			-100	-100
8	[Fv]					500			0	0
9	[Type]								1	1
10	Result		95	772		465			100	811
11	Formula		=PV(C5,C6,C7)	=PV(D5,D6,D7)		=PV(F5,F6,F7,F8)			=PV(J5,J6,J7,J8,J9)	=PV(K5,K6,K7,K8,K9)
12										
13	EXPLICIT CALCS									
14										
15		Discounted	Cumulative		Future Payment	Discounted Future Pay	Total		Discounted	Cumulative
16	1		95	95					100	100
17	2		91	186					95	195
18	3		86	272					91	286
19	4		82	355					86	372
20	5		78	433					82	455
21	6		75	508					78	533
22	7		71	579					75	608
23	8		68	646					71	679
24	9		64	711					68	746
25	10		61	772	-500	-307	465		64	811
26			=C57/(1+C55)^\$B25 =SUM(C516:C25)		=F8	=F25/(1+\$C55)^\$B25 =D25+G25			=C57/(1+C55)^(\$B25-1) =SUM(J516:J25)	

Gambar 4.4 Contoh Fungsi PV

Pembaca yang cenderung matematis akan melihat bahwa nilai-nilai tersebut dapat diperoleh dengan manipulasi matematis langsung, misalnya dengan:

$$S = 1 + (1 + g) + \dots + (1 + g)^{N-1}$$

Dengan mengalikan setiap sisi dengan  $(1+g)$ :

$$S(1 + g) = (1 + g) + (1 + g)^2 + (1 + g)^N$$

Dengan mengurangi persamaan pertama dari persamaan kedua, dan membaginya dengan  $g$ , akan diperoleh:

$$S = \frac{(1 + g)^N - 1}{g}$$

Mirip dengan:

$$T = 1 + \frac{1}{1 + d} + \dots + \frac{1}{(1 + d)^N}$$

seseorang memiliki

$$T = \frac{(1 + d)^N - 1}{d(1 + d)^N}$$

Rumus-rumus ini diimplementasikan dalam berkas Excel (Baris 29, tidak ditampilkan dalam cuplikan layar).

#### 4.2..3 Contoh: PMT, IPMT, PPMT, CUMIPMT, CUMPRINC, dan NPER

Beberapa fungsi terkait dengan perhitungan pembayaran pinjaman dan hipotek:

- PMT menghitung tingkat pembayaran tetap yang harus dilakukan pada setiap periode sehingga pinjaman (dengan suku bunga tetap) dilunasi pada akhir masa pinjaman. Angka tunggal ini secara implisit terdiri dari komponen pembayaran bunga dan satu komponen pembayaran pokok. Karena pokok dilunasi secara bertahap, jumlah bunga pada setiap periode berkurang, sehingga porsi pembayaran pokok meningkat dari satu periode ke periode berikutnya.
- IPMT dan PPMT menghitung komponen bunga dan pokok secara eksplisit; karena komponen ini berubah pada setiap periode waktu, fungsi tersebut memiliki parameter tambahan (dibandingkan dengan PMT) untuk menunjukkan periode mana yang dirujuk.
- CUMIPMT dan CUMPRINC menghitung bunga kumulatif dan pokok yang dibayarkan antara dua periode waktu tertentu.
- NPER menghitung jumlah periode yang diperlukan untuk melunasi pinjaman dengan jumlah pembayaran yang konstan.

Gambar 4.5 menunjukkan contohnya (lihat lembar kerja “PMT” untuk semua fungsi, kecuali NPER, yang ditunjukkan dalam lembar kerja “NPER”). Berkas tersebut juga berisi perhitungan terkait yang akan diperlukan jika dilakukan secara eksplisit di Excel, yaitu tanpa menggunakan fungsi tersebut. Perhatikan bahwa untuk perhitungan eksplisit, jumlah pembayaran diasumsikan sebagai angka yang dikodekan secara kaku yang sama dengan yang dihasilkan oleh fungsi PMT; jika fungsi tersebut tidak tersedia, angka ini dapat ditentukan melalui eksperimen manual atau lebih efisien melalui Goal-Seek atau Solver Excel. Gambar 4.5 dan Gambar 4.6 menunjukkan cuplikan layar dari contoh tersebut.

	A	B	C	D	E	F	G
1							
2		Periodic interest rate	5.0%				
3		Periods to end of term	15			Needs to be <=15 if using the	
4		Principal amount (\$)	100,000				
5		FV	0			Optional argument (not used i	
6		Type	0			Optional for some of the functi	
7							
8		For CUM functions					
9		First period	2				
10		Last period	3				
11							
12		Using PMT, CUMIPMT, CUMPRINC, IPMT and PPMT					
13							
14		Total Constant Payment using PMT	-9,634	=PMT(C2,C3,C4)			
15							
16		Period by period Split into interst and principal		1	2	3	4
17		Interest part of total, using IPMT	-44,513	-5,000	-4,768	-4,525	-4,270
18		Principal part of total, using PPMT	-100,000	-4,634	-4,866	-5,109	-5,365
19							
20		Cumulative interest between periods using CUMIPMT	-9,293	=CUMIPMT(C2,C3,C4,C9,C10,C6)			
21		Cumulative principal between periods using CUMPRINC	-9,975	=CUMPRINC(C2,C3,C4,C9,C10,C6)			
22							
23		Explicit Calculation					
24							
25		Periodic payment assumed	9,634				Set so that ending balance is :
26		Ending balance at end of term	0				Cross-check, or for use with G
27							
28		Direct calculation of interest and principal by period		1	2	3	4
29		Starting balance		100,000	95,366	90,500	85,391
30		(+) Interest (=balance * interest rate)	9,293	5,000	4,768	4,525	4,270
31		(-) Principal repayment (=assumed payment-interest)	9,975	4,634	4,866	5,109	5,365
32		Ending balance		95,366	90,500	85,391	80,026
33							

Gambar 4.5 Contoh Penggunaan Fungsi PMT, IPMT, PPMT, CUMIPMT dan CUMPRINC

	A	B	C	D
1				
2		Rate	5.0%	
3		PMT	-9634	
4		PV	100000	
5		FV	0	
6		Type	0	
7		NPER	15.0	=NPER(C2,C3,C4,C5,C6)

Gambar 4.6 Contoh Fungsi NPER

#### 4.2.4 Contoh: NPV dan IRR untuk Keputusan Beli atau Sewa

Fungsi NPV dan IRR digunakan secara luas dalam evaluasi dan analisis investasi umum. Untuk arus kas yang berjarak sama, NPV menghitung nilai sekarang bersihnya pada tingkat diskonto tertentu, dan IRR menghitung tingkat pengembalian internalnya (yaitu tingkat diskonto yang akan membuat nilai sekarang bersih sama dengan nol). Penggunaan fungsi NPV untuk menghitung nilai diskonto dari serangkaian arus kas pada dasarnya mudah (asalkan

seseorang terbiasa dengan konsep diskonto).

Kesalahan yang sering terjadi adalah mengabaikan bahwa fungsi tersebut secara implisit mengasumsikan bahwa nilai dalam sel pertama rentang didiskontokan untuk satu periode, nilai kedua didiskontokan untuk dua periode, dan seterusnya. Ini setara dengan mengasumsikan bahwa semua arus kas terjadi pada akhir periodenya. Beberapa penyesuaian mungkin diperlukan jika waktunya diperkirakan berbeda: misalnya, investasi yang dilakukan pada awal proyek tidak boleh didiskontokan, dan karenanya harus dikecualikan sebagai input ke fungsi NPV, tetapi ditambahkan ke (atau dikurangkan dari) hasilnya.

Gambar 4.7 memberikan contoh dalam konteks keputusan untuk menyewa atau membeli aset. File ini menunjukkan perhitungan arus kas dalam setiap kasus dan arus kas diferensial antara kedua opsi tersebut. Fungsi IRR dan NPV digunakan, dan perhitungan nilai sekarang bersih dari arus kas diferensial (didiskontokan pada tingkat bunga setelah pajak) dibuat, dengan cara arus kas keluar awal yang terkait dengan pembelian aset tidak didiskontokan. File ini juga menunjukkan bahwa nilai sekarang bersih adalah nol jika IRR digunakan sebagai tingkat diskonto (lihat Gambar 4.7).

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1														
2			1,000,000											
3			10											
4			100,000											
5			6.0%											
6			30.0%											
7			4.2%											
8			125,000											
														(Could also use Goal Seek on this figure to set NPV to zero to find maximum acceptable to pay)
9														
10				0	1	2	3	4	5	6	7	8	9	10
11														
12			-125,000											
13			-87,500		-87,500	-87,500	-87,500	-87,500	-87,500	-87,500	-87,500	-87,500	-87,500	-87,500
14														
15			-1,000,000	-1,000,000	0	0	0	0	0	0	0	0	0	0
16			30,000		30,000	30,000	30,000	30,000	30,000	30,000	30,000	30,000	30,000	30,000
17			-1,000,000		30,000	30,000	30,000	30,000	30,000	30,000	30,000	30,000	30,000	30,000
18			-1,000,000		117,500	117,500	117,500	117,500	117,500	117,500	117,500	117,500	117,500	117,500
19														
20				3.05%	=IRR(D18:N18,10%)									
21			4.20%	-56388	=D18+NPV(C21,E18:N18)									
22														
23			3.05%	0	=D18+NPV(C23,E18:N18)									

**Gambar 4.7 Penggunaan NPV dan IRR untuk Keputusan Beli versus Sewa**

Fungsi IRR juga dapat digunakan untuk menghitung imbal hasil obligasi. Penggunaan IRR sebagai ukuran kinerja proyek memiliki beberapa kelemahan, termasuk:

- IRR hanya ada jika terdapat setidaknya satu perubahan tanda dalam arus kas. Misalnya, nilai diskonto dari arus kas yang murni positif akan selalu positif, betapa pun besarnya tingkat diskonto.
- Jika arus kas berubah tanda beberapa kali selama periode perkiraan, terdapat banyak kemungkinan nilai IRR sebanyak perubahan tanda arus kas. Jadi, untuk proyek dengan investasi awal, periode produksi kas, dan biaya penghentian operasi berikutnya, terdapat setidaknya dua kemungkinan nilai IRR. Fungsi tersebut akan mengembalikan satu nilai, yang sesuai dengan salah satu dari keduanya, tetapi seseorang tidak dapat secara apriori mengetahui nilai mana dari keduanya yang dikembalikan dan apakah itu

merupakan ukuran yang tepat.

- Karena merupakan ukuran persentase, maka tidak memungkinkan untuk membandingkan proyek-proyek yang besarnya berbeda dengan mudah.

Tidak jelas bagaimana risiko dapat diperhitungkan dalam keputusan proyek. Dua proyek mungkin memiliki IRR yang sama tetapi memiliki profil risiko yang sangat berbeda. Sebaliknya, pendekatan NPV secara lebih eksplisit memilih tingkat diskonto yang sesuai dengan risiko arus kas yang dipertimbangkan. Perilaku IRR dalam kaitannya dengan efek waktu dapat menyebabkan keputusan yang tidak efektif atau tidak tepat. Gambar 4.8 menunjukkan efek dari upaya menghitung NPV dan IRR untuk berbagai rangkaian arus kas, termasuk yang hanya positif (lihat Gambar 4.8). Profil arus kas pertama murni positif, sehingga IRR tidak ada sama sekali.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
1																	
2			2018	2019	2020	2021	2022	2023	2024	2025	2026	2027	2028	2029	2030		Formulae in column F
3																	
4																	
5																	
6																	
7																	
8																	
9																	
10																	

**Gambar 4.8 Hasil Fungsi IRR untuk Beberapa Profil Arus Kas Non-standar**

Profil kedua memiliki investasi di muka dan biaya penghentian operasional berikutnya. Perhatikan bahwa IRR arus kas masa depan ada dalam beberapa tahun, tetapi tidak selalu memberikan indikator yang baik tentang perolehan nilai masa depan. Misalnya, di Kolom G (tahun 2022), NPV arus kas masa depan positif, sedangkan IRR negatif. Gambar 4.9 menunjukkan dampak pada IRR dan NPV dari berbagai skenario untuk menunda arus kas masuk proyek setelah investasi dilakukan (lihat Gambar 4.9):

- Dalam contoh pertama, satu investasi dilakukan dan jumlah yang sama dikembalikan baik satu periode kemudian atau 10 periode kemudian. Perhatikan bahwa IRR adalah nol dalam kedua kasus, sedangkan NPV lebih rendah dalam kasus penundaan; ini memberikan beberapa intuisi tentang sensitivitas potensial IRR yang lebih rendah terhadap penundaan proyek (terutama untuk proyek yang IRR-nya rendah sejak awal).
- Dalam contoh kedua, proyek ditunda satu periode setelah investasi awal dilakukan; Perubahan pada NPV secara proporsional jauh lebih besar daripada perubahan pada IRR; sekali lagi, efeknya umumnya paling kuat untuk proyek-proyek yang NPV-nya hanya sedikit positif.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1														
2		NPV @10%	IRR	0	1	2	3	4	5	6	7	8	9	10
3														
4		-41	0.0%	-500	500	0	0	0	0	0	0	0	0	0
5		-279	0.0%	-500	0	0	0	0	0	0	0	0	0	500
6														
7		69	13.7%	-500	100	100	100	100	100	100	100	100	100	0
8		21	11.0%	-500	0	100	100	100	100	100	100	100	100	100
9														

**Gambar 4.9 Perbandingan NPV dan IRR Ketika Ada Kemungkinan Arus Kas Masuk Tertunda**

Poin utama dari contoh ini adalah, ketika hanya menggunakan IRR, seseorang dapat meremehkan (dampak) risiko penundaan pada proyek, terutama yang berada di ambang positif. Hal ini dapat menyebabkan para pengambil keputusan percaya bahwa menunda proyek tidak akan berdampak besar, dan karenanya tidak memberinya prioritas atau perhatian yang cukup.

Hal ini didasarkan pada gagasan bahwa ada serangkaian arus kas dari waktu ke waktu, dan setiap saat hak untuk memperoleh arus kas masa depan dapat diabaikan dengan imbalan nilai sekarang dari arus kas masa depan tersebut. Grafik menunjukkan IRR yang dihasilkan yang akan muncul, tergantung pada tahun di mana proyek dijual (lihat Gambar 4.10). Penggunaan IRR sebagai ukuran kinerja dapat (bisa dibayangkan) mendorong pemikiran jangka pendek dengan memberi insentif pada penjualan awal proyek tersebut. Variasi pada fungsi NPV dan IRR yang terkadang berguna meliputi:

- XNPV dan XIRR serupa dengan NPV dan IRR tetapi di mana waktu arus kas ditentukan (yaitu, umumnya, arus kas tidak berjarak sama).
- Fungsi MIRR dapat digunakan untuk menghitung tingkat pengembalian internal jika arus kas positif dan negatif dibiayai pada tingkat yang berbeda.

#### 4.2.5 Contoh: SLN, DDB, dan VDB

Excel memiliki berbagai fungsi untuk membantu dalam perhitungan penyusutan, termasuk:

- DB dan DDB menghitung penyusutan aset untuk periode tertentu dengan menggunakan metode saldo menurun tetap dan menurun ganda.
- SLN menghitung penyusutan aset secara garis lurus untuk satu periode.
- VDB menghitung penyusutan aset untuk periode tertentu atau sebagian dengan menggunakan pilihan metode saldo menurun.
- SYD menghitung penyusutan jumlah digit tahun suatu aset untuk periode tertentu.

Gambar 4.11 berisi contoh penggunaan fungsi SLN, DDB, dan VDB (lihat Gambar 4.11).

A	B	C	D	E	F	G	H	I
1	Use of VDB function to calculate depreciation for a single year							
2								
3		SLN		DDB: Ex 1	DDB: Ex 2		VDB	
4	Asset Cost	10,000	10,000	10,000	10,000	10,000	10,000	10,000
5	% value at end of life	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
6	Life of assets	10	10	10	10	10	10	10
7	Start period (or period for DDB)	NA	1	1	1	1	1	5
8	End period	NA	NA	NA	NA	NA	NA	6
9	Factor for VDB method	NA	1	2	2	2	2	2
10	No. Switch	NA	NA	NA	NA	NA	NA	0
11								
12	Function Results	1,000	1,000	2,000	2,000	2,000	2,000	655
13		=SLN(D4,D4*D5,D6)	=DDB(F4,F4*F5,F6,F7,F9)	=DDB(G4,G4*G5,G6,G7,G9)	=DDB(G4,G4*G5,G6,G7,G9)	=DDB(G4,G4*G5,G6,G7,G9)	=DDB(G4,G4*G5,G6,G7,G9)	=VDB(I4,I4*I5,I6,I7,I8,I9,I10)

Gambar 4.11 Contoh Fungsi SLN, DDB, dan VBB

Tentu saja, dalam praktiknya, fungsi-fungsi tersebut mungkin diperlukan untuk menghitung profil penyusutan berdasarkan kumpulan input data belanja modal (CapEx), atau jika angka CapEx dihitung dari asumsi lain. Penyusutan dihasilkan dari perhitungan "tipe segitiga", seperti yang dibahas sebelumnya dalam teks (juga menggunakan fungsi TRANSPOSE untuk mengubah tanggal asli dan rentang CapEx) dan fungsi VDB disematkan dalam pernyataan pemeriksaan kesalahan (menggunakan fungsi IF dan AND untuk memastikan bahwa kesalahan tidak ditampilkan pada tahun di mana CapEx sebelumnya telah disusutkan sepenuhnya) (lihat Gambar 4.12).

A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	Use of VDB function to calculate depreciation schedule from a given capex schedule												
2													
3	Depreciation policy												
4	Life of assets	7											
5	% value at end of life	10.0%											
6	Factor for VDB method	2	E.g. 1=st. line when no residual value, 2=double declining balance etc.										
7													
8			2018	2019	2020	2021	2022	2023	2024	2025	2026	2027	2028
9													
10	Capex (€k)		8,000	10,000	12,000	15,000	12,000	10,000	8,000	6,000	5,000	5,000	5,000
11													
12		Capex	2018	2019	2020	2021	2022	2023	2024	2025	2026	2027	2028
13	2018	8,000	2,286	1,633	1,166	833	595	425	262	0	0	0	0
14	2019	10,000	0	2,857	2,041	1,458	1,041	744	531	328	0	0	0
15	2020	12,000	0	0	3,429	2,449	1,749	1,249	892	637	394	0	0
16	2021	15,000	0	0	0	4,286	3,061	2,187	1,562	1,116	797	492	0
17	2022	12,000	0	0	0	0	3,429	2,449	1,749	1,249	892	637	394
18	2023	10,000	0	0	0	0	0	2,857	2,041	1,458	1,041	744	531
19	2024	8,000	0	0	0	0	0	0	2,286	1,633	1,166	833	595
20	2025	6,000	0	0	0	0	0	0	0	1,714	1,224	875	625
21	2026	5,000	0	0	0	0	0	0	0	0	1,429	1,020	729
22	2027	5,000	0	0	0	0	0	0	0	0	0	1,429	1,020
23													
24	Depreciation total		2,286	4,490	6,636	9,025	9,875	9,911	9,324	8,135	6,944	6,030	3,894

Gambar 4.12 Perhitungan Profil Depresiasi Penuh Berdasarkan Profil CapEx

#### 4.2.6 Contoh: YIELD

Fungsi YIELD menghitung yield pada sekuritas yang membayar bunga berkala. Fungsi

ini mirip dalam banyak hal dengan IRR, kecuali bahwa tanggal jatuh tempo dan penyelesaian secara eksplisit diperlukan sebagai argumen, dan tidak ada profil arus kas eksplisit (periode demi periode) yang perlu dibuat (tidak seperti IRR). Aplikasi ini khusus untuk situasi jenis obligasi, sehingga ada satu pembelian atau peristiwa arus kas keluar yang diikuti oleh arus kas masuk, sedangkan IRR dapat diterapkan ketika arus kas keluar awal semuanya negatif selama beberapa periode. Gambar 4.13 menunjukkan contoh penggunaan fungsi YIELD, serta perhitungan eksplisit yield menggunakan IRR untuk situasi yang sesuai (lihat Gambar 4.13). Fungsi lain yang berkaitan dengan yield meliputi YIELDDISC, YIELDMAT, TBILLYIELD, ODDFYIELD, dan ODDLYIELD.

#### 4.2.7 Contoh: Durasi Arus Kas

Saat mempertimbangkan serangkaian arus kas, seseorang dapat mengembangkan beberapa kemungkinan ukuran waktu rata-ratanya:

- Arus kas nominal (non-diskon) rata-rata tertimbang waktu.
- Arus kas diskon rata-rata tertimbang waktu (dengan beberapa asumsi tingkat diskonto).
- Ukuran modifikasi lain yang dapat digunakan untuk tujuan standardisasi.

Nilai tertimbang waktu non-diskon dapat dianggap sebagai "durasi" arus kas, dengan rata-rata tertimbang waktu dari arus kas yang didiskon sering dikenal sebagai "durasi Macaulay". Saat durasi Macaulay dimodifikasi dengan diskon lebih lanjut, ini dikenal sebagai "durasi yang dimodifikasi". Gambar 4.14 menunjukkan contoh perhitungan ukuran tersebut untuk profil arus kas umum (lihat Gambar 4.14).

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1															
2		Calculation of bond yield from market price using YIELD function													
3															
4		Bond price	95												
5		Face value (\$)	100												
6		Annual coupon rate (%)	4.0%												
7		Settlement	01/01/2018												
8		Maturity	01/01/2028												
9		Frequency	1												
10															
11		Yield to maturity using YIELD	4.636%												
12															
13		Calculation of bond yield from market price using IRR function													
14															
15		Bond price	95												
16		Face value (\$)	100												
17		Annual coupon rate (%)	4.0%												
18															
19		Period Number		0	1	2	3	4	5	6	7	8	9	10	Total
20		Cash flow		-95.0	4.0	4.0	4.0	4.0	4.0	4.0	4.0	4.0	4.0	104.0	45.0
21															
22		Yield to maturity using IRR	4.636%												

Gambar 4.13 Penggunaan Fungsi YIELD dan Perbandingannya dengan IRR



#### 4.2.8 Contoh: DURATION dan MDURATION

Dalam konteks khusus perhitungan obligasi, dengan informasi yang relevan tentang obligasi (harga, kupon, tanggal penyelesaian dan jatuh tempo, dll.), seseorang dapat menghitung angka durasi secara langsung (tanpa harus melakukan prakiraan arus kas eksplisit). Perhatikan bahwa fungsi DURATION menghitung durasi Macaulay (yaitu berdasarkan arus kas yang didiskontokan pada tingkat imbal hasil), dan MDURATION menghitung modifikasi durasi Macaulay ini. Dengan demikian, angka "mentah" atau tidak didiskontokan (yaitu "durasi") tidak tersedia melalui fungsi Excel tertentu, meskipun dapat dihitung dengan menggunakan SUMPRODUCT.

		1	2	3	4	5	6	7	8	9	10	Total	Time Average
General Cash Flows													
Cash Flow		10.0	10.0	10.0	10.0	10.0	10.0	10.0	10.0	10.0	10.0	100.0	5.50
Periodic Discount Factor	8.0%	8.0%	8.0%	8.0%	8.0%	8.0%	8.0%	8.0%	8.0%	8.0%	8.0%		=SUMPRODUCT(D54:M54,D5:M5)/N5
Cumulative Discount Factor		92.6%	85.7%	79.4%	73.5%	68.1%	63.0%	58.3%	54.0%	50.0%	46.3%		
Discounted Cash Flow		9.26	8.57	7.94	7.35	6.81	6.30	5.83	5.40	5.00	4.63	67.1	4.87
													=SUMPRODUCT(D54:M54,D8:M8)/N8
													4.51
													=O8/(1+C6)

Gambar 4.14 Perhitungan Durasi, Durasi Macaulay dan Durasi Modifikasi Arus Kas

lihat Gambar 4.15 menunjukkan sebuah contoh. Perhatikan bahwa fungsi YIELD diperlukan sebagai input untuk perhitungan (untuk menemukan tingkat diskonto). Perhitungan yang setara juga ditampilkan secara langsung di Excel seolah-olah arus kas eksplisit telah diramalkan.

		0	1	2	3	4	5	6	7	8	9	10	Total	Time Average
Bond Functions (For \$100 face value/principal)														
Price	85.0													
Coupon	5.0%													
Settlement	1-Jan-10													
Maturity	1-Jan-20													
Frequency	1													
Basis	1													
YIELD	7.13%	=YIELD(C7,C3,-C18,100,C8,C9)												
[Macaulay] DURATION	7.80	=DURATION(C6,C7,C5,C11,C8,C9)												
[Modified [Macaulay] MDURATION	7.38	=MDURATION(C6,C7,C3,C11,C8,C9)												
Using Explicit Cash Flows (For \$100 face value/principal)														
Cash Flows	-85	5.0	5.0	5.0	5.0	5.0	5.0	5.0	5.0	5.0	5.0	105.0	150.0	8.50
Periodic Discount Factor FROM YIELD	7.13%	7.2%	7.2%	7.2%	7.2%	7.2%	7.2%	7.2%	7.2%	7.2%	7.2%	7.2%	7.2%	
Cumulative Discount Factor		95.5%	87.1%	81.5%	75.9%	70.8%	66.1%	61.7%	57.5%	53.7%	50.1%			
Discounted Cash Flow		4.67	4.35	4.06	3.78	3.54	3.30	3.08	2.88	2.69	2.53	85.0	7.92	
													7.38	
													7.38	
													=O22/(1+C20)	

Gambar 4.15 Penggunaan DURATION dan MDURATION dan Perbandingan dengan Perhitungan Eksplisit

#### 4.2.9 Contoh: PDURATION dan RRI

Fungsi PDURATION tidak terkait langsung dengan perhitungan obligasi, maupun dengan fungsi durasi lain yang dibahas di atas; sebaliknya, fungsi ini menghitung jumlah periode yang diperlukan agar investasi mencapai nilai tertentu, dengan asumsi tingkat pertumbuhan periodik. Dengan kata lain, fungsi ini menemukan  $n$  sehingga:

$$FV = PV(1 + g)^n$$

Tentu saja, alih-alih menggunakan fungsi, seseorang dapat memecahkan persamaan ini secara langsung (untuk menemukan n):

$$n = \frac{LN\left(\frac{FV}{PV}\right)}{LN(1 + g)}$$

Demikian pula, fungsi RRI menyediakan nilai yang dapat ditemukan dengan memecahkan g:

$$g = EXP\left(LN\left(\frac{FV}{PV}\right) / n\right) - 1$$

RRI dapat dianggap sebagai tingkat pengembalian tersirat jika g dianggap sebagai tingkat pengembalian dan bukan tingkat pertumbuhan. Gambar 4.16 menyediakan contoh fungsi-fungsi ini (lihat Gambar 20.16).

	A	B	C	D	E	F	G	H	I
1									
2		Rate	10%			NPER	7.27		
3		PV	100			PV	100		
4		FV	200			FV	200		
5		PDURATION	7.27	=PDURATION(C2,C3,C4)		RRI	10.0%	=RRI(G2,G3,G4)	
6		Direct calculation	7.27	=LN(C4/C3)/LN(1+C2)		Direct calculation	10.0%	=EXP(LN(G4/G3)/G2)-1	

**Gambar 4.16 Contoh PDURATION dan Ekuivalennya yang Dihitung Secara Langsung**

#### 4.3 FUNGSI KEUANGAN LAINNYA

Fungsi lain dalam kategori Keuangan Excel sebagian besar terkait dengan suku bunga dan obligasi:

- ACCRINT dan ACCRINTM menghitung bunga yang masih harus dibayar untuk sekuritas yang membayar bunga secara berkala atau pada saat jatuh tempo.
- COUPDAYBS, COUPDAYS, COUPDAYSNC, COUPNCD, COUPNUM, dan
- COUPPCD melakukan berbagai perhitungan tentang tanggal, angka, dan jumlah kupon yang dibayarkan.
- DISC menghitung tingkat diskonto untuk sekuritas.
- DOLLARDE dan DOLLARFR mengonversi harga dolar yang dinyatakan sebagai pecahan ke harga yang dinyatakan sebagai angka desimal, dan sebaliknya.
- EFFECT dan NOMINAL menghitung suku bunga efektif dan nominal saat bunga dapat dimajemukkan beberapa kali per periode.
- INTRATE menghitung suku bunga untuk sekuritas yang diinvestasikan sepenuhnya.

- RATE menghitung suku bunga per periode anuitas.
- ODDFPRICE (ODDLPRICE) menghitung harga per nilai nominal \$100 dari sekuritas dengan periode pertama (terakhir) ganjil.
- PRICE menghitung nilai sekuritas yang membayar bunga berkala.
- PRICEDISC menghitung nilai sekuritas yang didiskon.
- PRICEMAT menghitung nilai sekuritas yang membayar bunga saat jatuh tempo.
- RECEIVED menghitung jumlah yang diterima saat jatuh tempo untuk sekuritas yang diinvestasikan penuh.
- TBILLEQ menghitung imbal hasil setara obligasi untuk surat utang negara.
- TBILLPRICE menghitung nilai surat utang negara.

Pembaca dapat mempelajarinya lebih mendalam jika relevansinya potensial.

## BAB 5

### FUNGSI STATISTIK

#### 5.1 PENDAHULUAN

Bab ini menyediakan contoh-contoh dari banyak fungsi dalam kategori Statistik Excel. Fungsi-fungsi ini sering kali diperlukan untuk melakukan analisis data, menyelidiki hubungan antara variabel, dan untuk memperkirakan atau mengkalibrasi masukan atau hasil model. Sebagian besar perhitungan menyangkut:

- Posisi, peringkat, sebaran, dan bentuk kumpulan data.
- Distribusi probabilitas (perhitungan X-ke-P dan P-ke-X).
- Korelasi dan analisis regresi.
- Peramalan dan perhitungan statistik lainnya.

Sejak Excel 2010, kategori ini telah mengalami perubahan signifikan, yang diperkenalkan karena beberapa alasan:

- Untuk memberikan definisi yang lebih jelas, misalnya apakah perhitungan didasarkan pada sampel data atau pada statistik populasi penuh.
- Untuk memungkinkan kemungkinan fungsi tambahan dan opsi pengguna, seperti apakah distribusi probabilitas dinyatakan dalam bentuk kumulatif atau kepadatannya.
- Untuk membuat jenis perhitungan baru yang sebelumnya mungkin tidak ada, seperti yang ada dalam rangkaian fungsi terkait FORECAST.

Untuk membuat kompatibilitas mundur dengan versi Excel sebelumnya, banyak perubahan telah diperkenalkan dengan memberi nama fungsi baru dengan ".", seperti MODE.SNGL, atau RANK.EQ atau STDEV.S. Perlu dicatat bahwa kategori fungsi Excel dalam beberapa kasus merupakan konstruksi yang sedikit arbitrer, dengan hasil bahwa beberapa fungsi tidak selalu berada dalam kategori yang diharapkan. Secara khusus, perbedaan antara fungsi Statistik dan Math&Trig tidak sepenuhnya jelas.

Misalnya, SUMIFS, COMBIN, dan RAND diklasifikasikan dalam kategori Math&Trig, sedangkan AVERAGEIFS, PERMUT, NORM.DIST, NORM. IN, dan PROBE berada dalam kategori Statistik. Perhatikan bahwa bab ini tidak memberikan pembahasan spesifik lebih lanjut tentang fungsi yang telah dibahas sebelumnya (seperti AVERAGE, AVERAGEA, AVERAGEIF, AVERAGEIFS, MAXIFS, MINIFS, COUNT, COUNTA, COUNTBLANK, COUNTIF, COUNTIFS, MAX, MAXA, MIN, MINA).

#### 5.2 APLIKASI PRAKTIS: POSISI, PERINGKAT DAN NILAI PUSAT

Secara umum, ketika disajikan dengan sekumpulan data, seseorang mungkin ingin menghitung informasi ringkasan, seperti:

- Nilai rata-rata, minimum dan maksimum.
- Nilai yang paling mungkin (modus atau yang paling sering muncul).

- Titik median, yaitu di mana 50% item lebih tinggi/lebih rendah dari nilai ini.
- Nilai yang dilampaui dalam 10% kasus, atau di mana 10% item dalam kumpulan data berada di bawah nilai ini.
- Ukuran penyebaran, rentang atau sebaran data. Fungsi utama yang berkaitan dengan informasi tersebut adalah:
- MODE.SNGL menghitung nilai yang paling umum (paling sering) dalam kumpulan data, selama ada nilai unik. MODE.MULT menghitung larik vertikal dari yang paling sering muncul dalam rentang data, termasuk nilai berulang. MODE adalah versi lama dari fungsi MODE.SNGL.
- GEOMEAN dan HARMEAN menghitung rata-rata geometrik dan harmonik dari sekumpulan data, dan TRIMMEAN menghitung rata-rata bagian dalam sekumpulan data.
- LARGE dan SMALL menghitung nilai terbesar atau terkecil yang ditentukan dalam sekumpulan data (misalnya pertama, kedua, ketiga terbesar atau terkecil, dst.)
- RANK.EQ (RANK dalam versi Excel sebelumnya) menghitung peringkat angka dalam daftar angka, dengan RANK.AVG memberikan peringkat rata-rata dalam kasus nilai yang seri.
- PERCENTILE.EXC dan PERCENTILE.INC (dan PERCENTILE yang lama) menghitung persentil tertentu dari nilai dalam rentang, masing-masing eksklusif dan inklusif. Fungsi QUARTILE.EXC dan QUARTILE.INC (QUARTILE pada versi sebelumnya) menghitung persentil untuk kelipatan persentase ke-25 tertentu. MEDIAN menghitung median dari angka yang diberikan, yaitu persentil ke-50.
- PERCENTRANK.INC (PERCENTRANK pada versi sebelumnya) dan PERCENTRANK.EXC menghitung peringkat persentase nilai dalam kumpulan data, masing-masing inklusif dan eksklusif.

#### **Contoh: Menghitung Rata-rata dan Modus**

Rata-rata dari sekumpulan data dapat dihitung dengan menjumlahkan nilai-nilai individual dan membaginya dengan jumlah item. Di Excel, rata-rata dapat dihitung secara langsung menggunakan fungsi AVERAGE atau dengan membagi hasil menggunakan SUM dan COUNT. Sehubungan dengan rata-rata, perhatikan bahwa:

- Rata-rata yang dihitung dari kumpulan penuh (mentah) titik data individual juga disebut "rata-rata tertimbang". Hal ini karena setiap nilai akan secara otomatis dimasukkan dalam perhitungan secara total sesuai dengan frekuensi kemunculannya dalam set data.
- Dalam matematika, rata-rata (tertimbang) juga disebut "mean" atau "Expected Value (EV)". Pada saat penulisan ini, tidak ada fungsi "MEAN" atau "WEIGHTEDAVERAGE" di Excel. Namun, dalam kasus di mana set data lengkap dari masing-masing item (termasuk nilai berulang) tidak disediakan, melainkan nilai dan frekuensi setiap nilai yang diberikan, fungsi SUMPRODUCT dapat digunakan untuk menghitung rata-rata (tertimbang).

- Untuk menghitung frekuensi setiap item dalam satu set data, seseorang dapat menggunakan fungsi tipe COUNTIFS (lihat Bab 1), atau fungsi array FREQUENCY (dibahas dalam Bab 2). Aplikasi penting adalah menghitung frekuensi titik-titik dalam satu set data yang berada dalam rentang ("bucket" atau "bin").
- Perhitungan kondisional lainnya mungkin juga diperlukan: misalnya, AVERAGEIFS, MINIFS, dan MAXIFS (dan fungsi array yang terkait dengannya) dapat digunakan untuk menghitung angka yang sesuai untuk subset data menurut kriteria yang ditentukan. Ini dibahas dalam Bab 1 dan Bab 2, jadi tidak dibahas lebih lanjut.

Nilai yang paling mungkin (atau paling sering muncul) dalam satu set data dikenal sebagai modus (atau nilai modal). Ini dapat dianggap sebagai "estimasi terbaik", atau memang sebagai nilai yang mungkin "diharapkan" terjadi; mengharapkan hal lain sama saja dengan mengharapkan sesuatu yang kurang mungkin. Oleh karena itu, sering kali nilai itulah yang akan dipilih sebagai nilai kasus dasar untuk input model, khususnya jika nilai tersebut ditetapkan menggunakan penilaian atau estimasi ahli.

Nilai tersebut tidak sama dengan definisi matematis dari "Nilai yang Diharapkan", yang merupakan rata-rata (atau mean) (tertimbang); secara umum, kedua "harapan" ini akan berbeda, kecuali jika kumpulan data tersebut simetris. Nilai tersebut hanya ada jika satu nilai muncul lebih sering daripada yang lain. Jadi, untuk sampel data sederhana dari suatu proses (di mana setiap pengukuran hanya terjadi satu kali), mode tidak akan ada, dan fungsi akan mengembalikan pesan kesalahan.

Kita dapat melihat bahwa fungsi MODE.MULT akan melaporkan beberapa nilai ketika ada beberapa yang memiliki kemungkinan yang sama (tiga item terakhir dalam set data merupakan pengulangan nilai lain, dan ada tiga mode; kolom terakhir mengembalikan #N/A karena fungsi telah dimasukkan untuk mengembalikan hingga empat nilai).

	A	B	C	D	E
1					
2		<b>Data 1</b>			
3		1.59	1.59	=MODE(B3:B15)	
4		-1.78	1.59	=MODE.SNGL(B3:B15)	
5		1.12	1.59	=MODE.MULT(B3:B15)	
6		1.56	-1.78		
7		0.59	1.56		
8		-0.34			
9		1.03			
10		1.06			
11		1.05			
12		0.8			
13		1.56			
14		1.59			
15		-1.78			
16					

**Gambar 5.1 Penggunaan Fungsi Tipe MODE**

Berkas yang sama juga memperlihatkan bahwa fungsi-fungsi ini akan mengembalikan #N/A ketika setiap nilai dalam kumpulan data hanya muncul satu kali (lihat Gambar 5.2).

E	F	G	H
	<b>Data 2</b>		
	1.59	#N/A	=MODE(F3:F102)
	-1.78	#N/A	=MODE.SNGL(F3:F102)
	1.12	#N/A	{=MODE.MULT(F3:F102)}
	1.56		
	0.59		
	-0.34		
	1.03		
	1.06		
	1.05		
	0.80		
	0.61		
	0.87		
	-0.22		
	2.40		
	0.46		
	2.10		
	0.40		

**Gambar 5.2 Penggunaan Fungsi Tipe MODE dalam Situasi yang Hanya Berisi Nilai Unik**

Perlu dicatat bahwa meskipun tidak ada fungsi “MEAN” sederhana di Excel, fungsi TRIMMEAN, GEOMEAN, dan HARMEAN memang ada:

- TRIMMEAN menghitung rata-rata setelah mengecualikan sejumlah item terbesar dan terkecil tertentu (angka yang dikecualikan sama pada bagian atas dan bawah, sehingga fungsi menghitung jumlah yang diperlukan untuk dikecualikan berdasarkan nilai persentase masukan pengguna; oleh karena itu, seseorang perlu berhati-hati ketika menggunakan fungsi untuk memastikan bahwa jumlah pengecualian sesuai dengan angka yang diinginkan).
- GEOMEAN menghitung rata-rata geometrik dari sekumpulan n angka positif; ini adalah akar ke-n dari hasil perkalian angka-angka ini.
- HARMEAN menghitung rata-rata harmonik dari sekumpulan angka positif, yang merupakan kebalikan dari rata-rata kebalikan dari nilai-nilai individual.

#### **Contoh: Pengurutan Data Dinamis Menggunakan LARGE**

Meskipun pengurutan data dapat dilakukan menggunakan alat-alat pada tab Data Excel, ini adalah proses yang harus dilakukan secara manual (atau menggunakan makro VBA yang menjalankan prosedur pengurutan). Alternatifnya adalah menggunakan fungsi LARGE untuk menghitung data dalam urutan yang diurutkan. Ini sangat berguna dalam model yang perlu bersifat dinamis (baik jika data baru akan dimasukkan secara berkala, atau jika analisis sensitivitas keluaran perlu dijalankan).

Tentu saja, fungsi ini juga dapat digunakan untuk menemukan hanya item data terbesar, atau kedua atau ketiga terbesar, dan tidak ada persyaratan untuk harus menempatkan semua set data asli secara berurutan. Fungsi SMALL bersifat analog,

mengembalikan nilai terkecil terlebih dahulu. (Tentu saja, jika yang Anda minati hanya nilai terbesar atau terkecil, Anda cukup menggunakan MAX atau MIN.) gambar 5.4 memberikan contoh (lihat Gambar 5.4). Fungsi LARGE dan SMALL digunakan untuk mencantumkan 11 item terbesar dan terkecil dalam kumpulan data yang berisi 100 item.

	A	B	C	D
1				
2		<b>Data 1</b>		
3		1.59	0.668	=AVERAGE(B3:B12)
4		-1.78	0.668	=TRIMMEAN(B3:B12,0%)
5		1.12	0.668	=TRIMMEAN(B3:B12,10%)
6		1.56	0.859	=TRIMMEAN(B3:B12,30%)
7		0.59		
8		-0.34	1.010	=GEOMEAN(B5:B7)
9		1.03	1.010	=PRODUCT(B5:B7)^(1/COUNT(B5:B7))
10		1.06		
11		1.05	0.931	=HARMEAN(B5:B7)
12		0.80	0.931	=1/((1/3)*(1/B5+1/B6+1/B7))

Gambar 21.3 Contoh TRIMMEAN, GEOMEAN dan HARMEAN

	A	B	C	D	E	F	G	H
1								
2		<b>Data 1</b>		<b>Largest 11</b>			<b>Smallest 11</b>	
3	1	1.59		3.30	=LARGE(\$B\$3:\$B\$102,\$A3)		-1.78	=SMALL(\$B\$3:\$B\$102,\$A3)
4	2	-1.78		2.78	=LARGE(\$B\$3:\$B\$102,\$A4)		-1.62	=SMALL(\$B\$3:\$B\$102,\$A4)
5	3	1.12		2.66	=LARGE(\$B\$3:\$B\$102,\$A5)		-1.20	=SMALL(\$B\$3:\$B\$102,\$A5)
6	4	1.56		2.41	=LARGE(\$B\$3:\$B\$102,\$A6)		-0.77	=SMALL(\$B\$3:\$B\$102,\$A6)
7	5	0.59		2.40	=LARGE(\$B\$3:\$B\$102,\$A7)		-0.57	=SMALL(\$B\$3:\$B\$102,\$A7)
8	6	-0.34		2.39	=LARGE(\$B\$3:\$B\$102,\$A8)		-0.54	=SMALL(\$B\$3:\$B\$102,\$A8)
9	7	1.03		2.39	=LARGE(\$B\$3:\$B\$102,\$A9)		-0.51	=SMALL(\$B\$3:\$B\$102,\$A9)
10	8	1.06		2.30	=LARGE(\$B\$3:\$B\$102,\$A10)		-0.47	=SMALL(\$B\$3:\$B\$102,\$A10)
11	9	1.05		2.25	=LARGE(\$B\$3:\$B\$102,\$A11)		-0.46	=SMALL(\$B\$3:\$B\$102,\$A11)
12	10	0.80		2.21	=LARGE(\$B\$3:\$B\$102,\$A12)		-0.36	=SMALL(\$B\$3:\$B\$102,\$A12)
13	11	0.61		2.20	=LARGE(\$B\$3:\$B\$102,\$A13)		-0.34	=SMALL(\$B\$3:\$B\$102,\$A13)
14	12	0.87		2.10				
15	13	-0.22		2.10				
16	14	2.40		2.09				
17	15	0.46		2.08				
18	16	2.10		2.07				

Gambar 5.4 Menggunakan LARGE dan SMALL untuk Mengurutkan dan Memanipulasi Data

#### Contoh: RANK.EQ

Peringkat item dalam set datanya adalah posisi terurutnya; yaitu, angka terbesar memiliki peringkat 1 (atau terkecil, tergantung pada definisi yang disukai), angka terbesar kedua (atau terkecil) memiliki peringkat 2, dan seterusnya. Peringkat tersebut terkait erat dengan fungsi LARGE dan SMALL, pada dasarnya sebagai proses terbalik (misalnya peringkat



angka adalah posisi terurutnya seolah-olah set data diurutkan, sedangkan fungsi LARGE mengembalikan angka sebenarnya untuk posisi terurut tertentu).

Peringkat item terkadang diperlukan dalam pemodelan keuangan, seperti untuk pelaporan tentang seberapa penting suatu item. Hal ini juga diperlukan dalam perhitungan koefisien korelasi peringkat (Spearman) saat menganalisis hubungan antara beberapa set data, seperti yang dibahas nanti dalam bab ini. Gambar 5.5 menunjukkan contoh fungsi RANK.EQ yang diterapkan pada kumpulan data berisi 100 nilai unik, yang menunjukkan kasus saat parameter opsional dihilangkan dan saat parameter tersebut disertakan (untuk memberikan urutan peringkat menurun atau menaik) (lihat Gambar 5.5). Fungsi lama RANK juga ditampilkan dalam file (tetapi tidak dalam tangkapan layar).

	A	B	C	D	E	F	G	H
1								
2		Data			Rank Order Descending		Rank Order Ascending	
3		1.59			29 =RANK.EQ(B3,B\$3:B\$102)		72 =RANK.EQ(B3,B\$3:B\$102,1)	
4		-1.78			100 =RANK.EQ(B4,B\$3:B\$102)		1 =RANK.EQ(B4,B\$3:B\$102,1)	
5		1.12			44 =RANK.EQ(B5,B\$3:B\$102)		57 =RANK.EQ(B5,B\$3:B\$102,1)	
6		1.56			30 =RANK.EQ(B6,B\$3:B\$102)		71 =RANK.EQ(B6,B\$3:B\$102,1)	
7		0.59			68 =RANK.EQ(B7,B\$3:B\$102)		33 =RANK.EQ(B7,B\$3:B\$102,1)	
8		-0.34			90 =RANK.EQ(B8,B\$3:B\$102)		11 =RANK.EQ(B8,B\$3:B\$102,1)	
9		1.03			50 =RANK.EQ(B9,B\$3:B\$102)		51 =RANK.EQ(B9,B\$3:B\$102,1)	
10		1.06			48 =RANK.EQ(B10,B\$3:B\$102)		53 =RANK.EQ(B10,B\$3:B\$102,1)	

Gambar 5.5 Contoh Fungsi RANK.EQ

#### Contoh: RANK.AVG

Gambar 5.6 menunjukkan kumpulan data yang berisi nilai duplikat (seperti dalam sel B5 dan B16), sehingga terdapat peringkat yang seri. Fungsi RANK.EQ memberikan peringkat yang sama untuk setiap seri dan melewati posisi yang diurutkan (misalnya tidak ada item dengan peringkat 4). Nilai RANK.AVG memberikan nilai rata-rata item (yaitu 3,5, sebagai rata-rata 3 dan 4 dalam contoh ini).

#### Contoh: Menghitung Persentil

Statistik penting lainnya adalah persentil (kadang-kadang disebut persentil). Ini menunjukkan untuk setiap angka persentase yang diasumsikan nilai-x di bawah persentase hasil tersebut. Misalnya, persentil ke-10 (atau P10) adalah nilai di bawah 10% hasil, dan P90 adalah nilai di bawah 90% kejadian. Perhatikan bahwa minimum dan maksimum masing-masing adalah persentil ke-0 dan ke-100. Kasus khusus penting lainnya adalah persentil ke-50 (P50), yaitu titik di mana 50% item lebih tinggi/lebih rendah dari nilai ini. Ini dikenal sebagai median. Fungsi seperti PERCENTILE, PERCENTILE.INC, dan PERCENTILE.EXC dapat digunakan untuk menghitung persentil kumpulan data, tergantung pada versi Excel yang digunakan (dengan kasus khusus P0, P100, dan P50 juga dapat ditangkap melalui fungsi MIN, MAX, dan MEDIAN sebagai alternatif).

	A	B	C	D	E	F
1						
2		<b>Data</b>	<b>Rank Order Descending</b>		<b>Rank Order Descending</b>	
3		1.59	1 =RANK.EQ(B3,B\$3:B\$16)		1 =RANK.AVG(B3,B\$3:B\$16)	
4		-1.78	14 =RANK.EQ(B4,B\$3:B\$16)		14 =RANK.AVG(B4,B\$3:B\$16)	
5		1.12	3 =RANK.EQ(B5,B\$3:B\$16)		3.5 =RANK.AVG(B5,B\$3:B\$16)	
6		1.56	2 =RANK.EQ(B6,B\$3:B\$16)		2 =RANK.AVG(B6,B\$3:B\$16)	
7		0.59	11 =RANK.EQ(B7,B\$3:B\$16)		11 =RANK.AVG(B7,B\$3:B\$16)	
8		-0.34	13 =RANK.EQ(B8,B\$3:B\$16)		13 =RANK.AVG(B8,B\$3:B\$16)	
9		1.03	7 =RANK.EQ(B9,B\$3:B\$16)		7 =RANK.AVG(B9,B\$3:B\$16)	
10		1.06	5 =RANK.EQ(B10,B\$3:B\$16)		5 =RANK.AVG(B10,B\$3:B\$16)	
11		1.05	6 =RANK.EQ(B11,B\$3:B\$16)		6 =RANK.AVG(B11,B\$3:B\$16)	
12		0.80	9 =RANK.EQ(B12,B\$3:B\$16)		9 =RANK.AVG(B12,B\$3:B\$16)	
13		0.61	10 =RANK.EQ(B13,B\$3:B\$16)		10 =RANK.AVG(B13,B\$3:B\$16)	
14		0.87	8 =RANK.EQ(B14,B\$3:B\$16)		8 =RANK.AVG(B14,B\$3:B\$16)	
15		-0.22	12 =RANK.EQ(B15,B\$3:B\$16)		12 =RANK.AVG(B15,B\$3:B\$16)	
16		1.12	3 =RANK.EQ(B16,B\$3:B\$16)		3.5 =RANK.AVG(B16,B\$3:B\$16)	

**Gambar 5.6 RANK.EQ dan RANK.AVG dalam Kasus Item yang Terikat**

Meskipun secara teori, properti statistik ini umumnya didefinisikan dengan jelas hanya untuk kumpulan data kontinu (tak terbatas), dalam praktiknya properti ini akan diterapkan dalam banyak konteks pada data terbatas (diskrit). Dalam hal ini, seseorang perlu berhati-hati untuk menerapkannya dengan benar, serta memahami hasil yang dikembalikan: fungsi tipe PERCENTILE umumnya perlu melakukan interpolasi antar nilai untuk menemukan perkiraan terhadap angka sebenarnya (asumsi implisit adalah bahwa proses dasar yang menghasilkan data adalah proses yang berkelanjutan).

Misalnya, nilai PERCENTILE.INC({1,2,3,4,5},10%) dikembalikan sebagai 1,4, dan PERCENTILE.INC({1,2,3,4,5},20%) dikembalikan sebagai 1,8. Gambar 5.7 menunjukkan beberapa implementasi fungsi tipe PERCENTILE. Perhatikan bahwa, sementara PERCENTILE.INC dan fungsi PERCENTILE lama dapat diterapkan dengan nilai persentase apa pun (termasuk 0% atau 100%), fungsi PERCENTILE.EXC hanya akan berfungsi jika persentase input berada di antara  $1/n$  dan  $1-1/n$ , di mana  $n$  adalah jumlah titik dalam set data.

Dalam rentang ini, prosedur interpolasinya dapat memberikan estimasi yang lebih akurat dari nilai persentil yang mendasarinya; hal ini sebagian dapat dilihat dari berkas sehubungan dengan perhitungan P10 (sisi kanan berkas menunjukkan data dalam urutan yang diurutkan, menggunakan fungsi LARGE, untuk melihat nilai yang lebih akurat yang dihasilkan oleh PERCENTILE.EXC) (lihat Gambar 5.7). (Nanti dalam bab ini, kami memberikan contoh lebih lanjut tentang di mana fungsi terakhir ini juga lebih akurat, meskipun pengujian umum kami menunjukkan bahwa hal itu mungkin tidak selalu demikian.)

	A	B	C	D	E	F	G	H
1								
2		<b>Data 2</b>						
3		1.59	2.203	=PERCENTILE(B3:B102,90%)		1	3.302	=LARGE(B\$3:B\$102,\$F3)
4		-1.78	2.203	=PERCENTILE.INC(B3:B102,90%)		2	2.78	=LARGE(B\$3:B\$102,\$F4)
5		1.12	2.212	=PERCENTILE.EXC(B3:B102,90%)		3	2.66	=LARGE(B\$3:B\$102,\$F5)
6		1.56				4	2.41	=LARGE(B\$3:B\$102,\$F6)
7		0.59	3.044	=PERCENTILE(B3:B102,99.5%)		5	2.40	=LARGE(B\$3:B\$102,\$F7)
8		-0.34	3.044	=PERCENTILE.INC(B3:B102,99.5%)		6	2.39	=LARGE(B\$3:B\$102,\$F8)
9		1.03	#NUM!	=PERCENTILE.EXC(B3:B102,99.5%)		7	2.39	=LARGE(B\$3:B\$102,\$F9)
10		1.06				8	2.30	=LARGE(B\$3:B\$102,\$F10)
11		1.05	3.302	=PERCENTILE(B3:B102,100%)		9	2.25	=LARGE(B\$3:B\$102,\$F11)
12		0.80	3.302	=PERCENTILE.INC(B3:B102,100%)		10	2.213	=LARGE(B\$3:B\$102,\$F12)
13		0.61	#NUM!	=PERCENTILE.EXC(B3:B102,100%)		11	2.202	=LARGE(B\$3:B\$102,\$F13)
14		0.87				12	2.10	
15		-0.22				13	2.10	

Gambar 5.7 Penggunaan Fungsi tipe PERCENTILE dan MEDIAN

#### Contoh: Fungsi tipe PERCENTRANK

lihat Gambar 5.8 berisi contoh beberapa fungsi tipe PERCENTRANK; fungsi ini menghitung peringkat persentase suatu nilai dalam kumpulan data. Fungsi PERCENTRANK.INC (inklusif) sesuai dengan fungsi PERCENTRANK lama, sedangkan PERCENTRANK.EXC (eksklusif) menyediakan serangkaian perhitungan baru.

	A	B	C	D	E	F	G	H	I
1									
2		<b>Data</b>	<b>Rank Order Ascending</b>		<b>PERCENTRANK.INC</b>		<b>PERCENTRANK.EXC</b>		
3		1.59	72 =RANK.EQ(B3,B\$3:B\$102,1)		72% =PERCENTRANK.INC(B\$3:B\$102,B3)		71% =PERCENTRANK.EXC(B\$3:B\$102,B3)		
4		-1.78	1 =RANK.EQ(B4,B\$3:B\$102,1)		0% =PERCENTRANK.INC(B\$3:B\$102,B4)		1% =PERCENTRANK.EXC(B\$3:B\$102,B4)		
5		1.12	57 =RANK.EQ(B5,B\$3:B\$102,1)		57% =PERCENTRANK.INC(B\$3:B\$102,B5)		56% =PERCENTRANK.EXC(B\$3:B\$102,B5)		
6		1.56	71 =RANK.EQ(B6,B\$3:B\$102,1)		71% =PERCENTRANK.INC(B\$3:B\$102,B6)		70% =PERCENTRANK.EXC(B\$3:B\$102,B6)		
7		0.59	33 =RANK.EQ(B7,B\$3:B\$102,1)		32% =PERCENTRANK.INC(B\$3:B\$102,B7)		33% =PERCENTRANK.EXC(B\$3:B\$102,B7)		
8		-0.34	11 =RANK.EQ(B8,B\$3:B\$102,1)		10% =PERCENTRANK.INC(B\$3:B\$102,B8)		11% =PERCENTRANK.EXC(B\$3:B\$102,B8)		
9		1.03	51 =RANK.EQ(B9,B\$3:B\$102,1)		51% =PERCENTRANK.INC(B\$3:B\$102,B9)		50% =PERCENTRANK.EXC(B\$3:B\$102,B9)		
10		1.06	53 =RANK.EQ(B10,B\$3:B\$102,1)		53% =PERCENTRANK.INC(B\$3:B\$102,B10)		52% =PERCENTRANK.EXC(B\$3:B\$102,B10)		
11		1.05	52 =RANK.EQ(B11,B\$3:B\$102,1)		52% =PERCENTRANK.INC(B\$3:B\$102,B11)		51% =PERCENTRANK.EXC(B\$3:B\$102,B11)		
12		0.80	43 =RANK.EQ(B12,B\$3:B\$102,1)		42% =PERCENTRANK.INC(B\$3:B\$102,B12)		43% =PERCENTRANK.EXC(B\$3:B\$102,B12)		
13		0.61	35 =RANK.EQ(B13,B\$3:B\$102,1)		34% =PERCENTRANK.INC(B\$3:B\$102,B13)		35% =PERCENTRANK.EXC(B\$3:B\$102,B13)		

Gambar 5.8 Contoh Fungsi PERCENTRANK

Perhatikan bahwa, sementara pengaturan default untuk fungsi RANK.EQ, RANK.AVG dan RANK adalah untuk menunjukkan urutan peringkat menurun (yaitu ketika parameter opsional dihilangkan), untuk fungsi tipe PERCENTRANK peringkat implisit adalah menaik.

### 5.3 APLIKASI PRAKTIS: PENYEBARAN DAN BENTUK

Penyebaran dan "bentuk" keseluruhan titik juga merupakan aspek utama dari kumpulan data, yang memberikan indikasi agregat, bukan yang terkait dengan titik data

individual. Beberapa ukuran utama yang terkait dengan bentuk meliputi:

- FREKUENSI adalah fungsi array yang menghitung berapa kali titik dalam kumpulan data berada dalam setiap rangkaian rentang (atau bin) yang telah ditentukan sebelumnya, yaitu distribusi frekuensi.
- Simetri kumpulan data. Di Excel, SKEW.P (Excel 2013 dan seterusnya) menghitung kemiringan populasi data penuh, dan SKEW memperkirakan kemiringan populasi berdasarkan sampel (SKEW.S belum ada saat artikel ini ditulis).
- Sejauh mana titik data berada dalam ekor (menuju ekstrem minimum atau maksimum) versus yang lebih bersifat sentral. KURT memperkirakan kurtosis (kelebihan) populasi untuk kumpulan data yang mewakili sampel (KURT.P dan KURT.S tidak ada pada saat penulisan).

Beberapa ukuran utama penyebaran titik dalam suatu rentang meliputi:

Lebar total rentang (nilai maksimum dikurangi nilai minimum), atau perbedaan antara dua persentil (P90 dikurangi P10). Fungsi tipe MIN, MAX, PERCENTILE, dan fungsi terkait lainnya dapat digunakan dalam hal ini (lihat sebelumnya dalam bab ini).

Simpangan baku, atau ukuran simpangan baku lainnya, seperti semi-deviasi. Pada saat penulisan, ada banyak fungsi yang terkait dengan perhitungan simpangan baku dan varians kumpulan data. Akan tetapi, tidak ada fungsi untuk semi-deviasi (lihat bagian selanjutnya dalam bab ini untuk metode perhitungan, dan juga untuk implementasinya sebagai fungsi yang ditentukan pengguna menggunakan VBA):

- VAR.P (VARP dalam versi sebelumnya) menghitung varians populasi dari sekumpulan data, yaitu rata-rata kuadrat deviasi titik-titik dari mean, dengan asumsi bahwa data tersebut mewakili seluruh populasi. Demikian pula, VAR.S (VAR dalam versi sebelumnya) memperkirakan varians populasi berdasarkan asumsi bahwa kumpulan data tersebut mewakili sampel dari populasi (sehingga istilah koreksi untuk bias diperlukan untuk digunakan dalam rumus).
- STDEV.P (STDEVP dalam versi sebelumnya) dan STDEV.S (STDEV dalam versi sebelumnya) menghitung atau memperkirakan deviasi standar yang terkait dengan populasi, masing-masing berdasarkan data populasi atau sampel. Deviasi standar tersebut adalah akar kuadrat dari varians yang sesuai.
- AVEDEV menghitung rata-rata deviasi absolut titik data dari nilai rata-ratanya (deviasi rata-rata dari nilai rata-rata tentu saja nol).

Fungsi terkait lainnya meliputi:


- VARPA, STDEVPA, VARA, dan STDEVA menghitung statistik berbasis populasi dan sampel, termasuk angka, teks, dan nilai logika dalam perhitungan.
- DEVSQ menghitung jumlah kuadrat deviasi dari nilai rata-rata untuk kumpulan data.
- STANDARDIZE menghitung jumlah deviasi nilai input dari angka yang diasumsikan, dan dengan faktor deviasi standar yang diasumsikan.

Perlu dicatat bahwa dalam kebanyakan kasus, fungsi tersebut mengasumsikan bahwa kumpulan data yang diberikan adalah daftar kejadian individual dari sampel atau populasi.

Jadi, fungsi tersebut tidak memungkinkan seseorang untuk menggunakan data apa pun secara eksplisit tentang frekuensi kejadian. Pengecualian adalah penggunaan SUMPRODUCT di mana frekuensi adalah salah satu bidang input.

### Contoh: Membuat Histogram Pengembalian Menggunakan FREQUENCY

FREQUENCY menghitung jumlah titik data yang berada dalam setiap rangkaian rentang (atau bin) yang telah ditetapkan sebelumnya. Fungsi ini dapat digunakan untuk membuat data guna menghasilkan diagram batang (histogram) dari rangkaian data. Perhatikan bahwa fungsi ini adalah fungsi array yang harus dimasukkan dalam rentang vertikal (bukan horizontal).

	A	B	C	D	E	F	G
1							
2		Period	Returns	Xbins	FREQUENCY		
3		1	5.40%	-10.0%	0	{=FREQUENCY(C3:C202,E3:E23)}	
4		2	-8.16%	-9.00%	0	{=FREQUENCY(C3:C202,E3:E23)}	
5		3	-2.42%	-8.00%	1	{=FREQUENCY(C3:C202,E3:E23)}	
6		4	-1.00%	-7.00%	0	{=FREQUENCY(C3:C202,E3:E23)}	
7		5	2.09%	-6.00%	0	{=FREQUENCY(C3:C202,E3:E23)}	
8		6	0.80%	-5.00%	1	{=FREQUENCY(C3:C202,E3:E23)}	
9		7	0.16%	-4.00%	2	{=FREQUENCY(C3:C202,E3:E23)}	
10		8	1.47%	-3.00%	4	{=FREQUENCY(C3:C202,E3:E23)}	
11		9	1.16%	-2.00%	6	{=FREQUENCY(C3:C202,E3:E23)}	
12		10	1.24%	-1.00%	10	{=FREQUENCY(C3:C202,E3:E23)}	
13		11	1.43%	0.00%	31	{=FREQUENCY(C3:C202,E3:E23)}	
14		12	2.51%	1.00%	52	{=FREQUENCY(C3:C202,E3:E23)}	
15		13	1.51%	2.00%	54	{=FREQUENCY(C3:C202,E3:E23)}	
16		14	-0.86%	3.00%	24	{=FREQUENCY(C3:C202,E3:E23)}	
17		15	2.30%	4.00%	7	{=FREQUENCY(C3:C202,E3:E23)}	
18		16	1.06%	5.00%	6	{=FREQUENCY(C3:C202,E3:E23)}	
19		17	1.61%	6.00%	1	{=FREQUENCY(C3:C202,E3:E23)}	
20		18	-1.36%	7.00%	0	{=FREQUENCY(C3:C202,E3:E23)}	
21		19	-0.63%	8.00%	1	{=FREQUENCY(C3:C202,E3:E23)}	
22		20	0.57%	9.00%	0	{=FREQUENCY(C3:C202,E3:E23)}	
23		21	1.35%	10.00%	0	{=FREQUENCY(C3:C202,E3:E23)}	
24		22	2.38%	Infinity	0	{=FREQUENCY(C3:C202,E3:E23)}	
25		23	-2.01%				

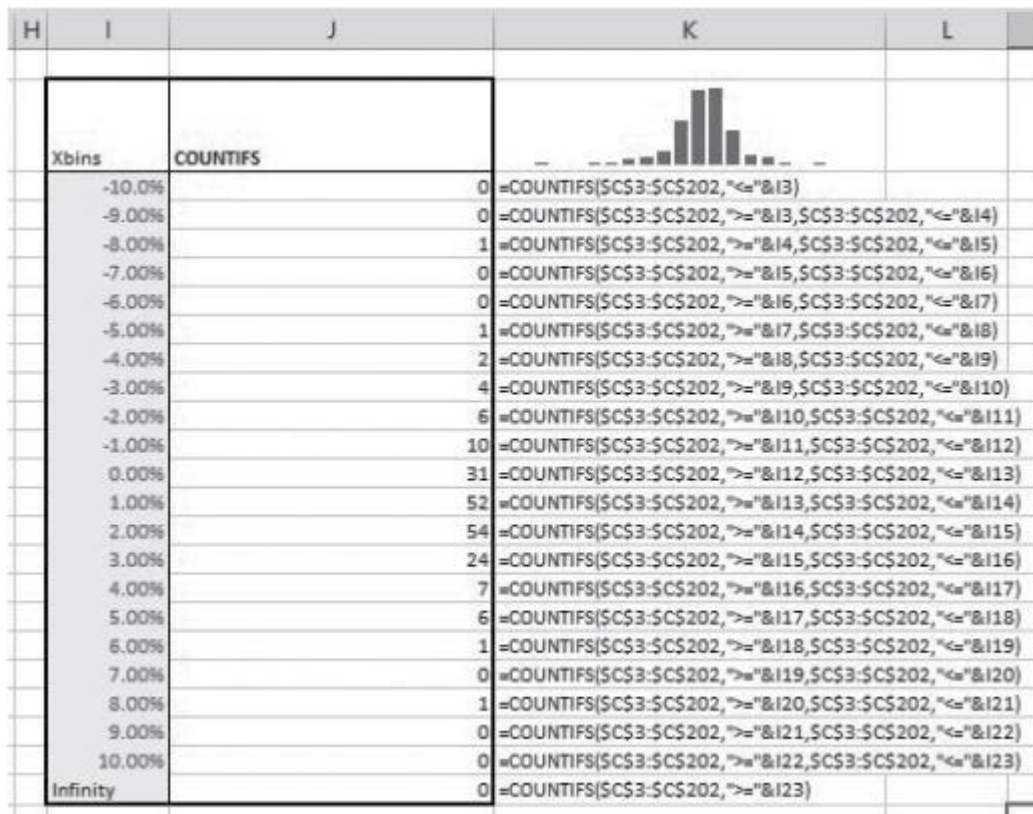
Gambar 5.9 Menggunakan Fungsi FREQUENCY

Gambar 5.9 menunjukkan contoh dalam konteks data pengembalian untuk aset yang dikutip pasar selama beberapa periode (lihat Gambar 5.9). Sel G2 juga berisi bentuk Sparkline dari bagan Kolom, yang dibuat menggunakan opsi Sparkline Excel dalam menu Sisip utama. Perhatikan beberapa poin tambahan terkait fungsi ini:

- Sebagai fungsi array, seluruh rentang yang akan memuat fungsi tersebut harus dipilih (rentang ini harus satu sel lebih besar dari rentang bin, karena mungkin ada titik data yang lebih besar dari nilai atas bin terbesar).
- Pilihan definisi (lebar) bin akan menjadi penting; jika lebarnya terlalu lebar, histogram tidak akan cukup terperinci, sedangkan jika terlalu sempit, histogram akan terlihat terfragmentasi dan mungkin multi-modus.

- Karena fungsi tersebut menghitung jumlah titik data yang terletak di antara nilai bin bawah dan atas, seseorang dapat membuat hasil yang sama menggunakan fungsi COUNTIFS untuk setiap bin (dengan dua kriteria: satu terkait dengan batas bawah bin tertentu dan yang lainnya terkait dengan batas atasnya, dengan penyesuaian yang sesuai untuk bin pertama dan terakhir, yang masing-masing tidak memiliki batas bawah atau atas). Histogram juga dapat dibuat menggunakan variasi aplikasi fungsi PERCENTRANK, tetapi sering kali dalam praktiknya manfaatnya kecil atau tidak ada sama sekali.

Gambar 5.10 menunjukkan penggunaan fungsi COUNTIFS dalam konteks yang sama.



Gambar 5.10 Penggunaan COUNTIFS sebagai Alternatif untuk FREKUENSI

#### Contoh: Varians, Simpangan Baku, dan Volatilitas

Simpangan baku ( $\sigma$ ) memberikan ukuran standar dari rentang atau sebaran. Ini adalah ukuran deviasi "rata-rata" di sekitar nilai tengah. Jika semua hal lain sama, distribusi yang memiliki simpangan baku yang lebih besar daripada distribusi lain lebih menyebar (dan memiliki lebih banyak ketidakpastian atau risiko) yang terkait dengannya. Simpangan baku dihitung sebagai akar kuadrat dari varians ( $V$ ):

$$V(x) = \sum P_i (x_i - \mu)^2$$

Dan

$$\sigma = \sqrt{V} = \sqrt{\sum P_i (x_i - \mu)^2}$$

Di Sini

$$\mu = \sum P_i x_i = E(x)$$

di mana  $p$  adalah probabilitas relatif (atau frekuensi kejadian) dari nilai- $x$  yang sesuai;  $\mu$  menunjukkan nilai rata-rata atau ekspektasi matematis (Nilai Harapan), yang juga diberikan oleh  $E$ . Kita juga dapat menulis persamaan varians sebagai:

$$V = E((x - \mu)^2)$$

Atau

$$V = E((x - E(x))^2)$$

yang dapat diperluas menjadi:

$$V = E(x^2 - 2xE(x) + E(x)^2)$$

Atau

$$V = E(x^2) - 2E(x)E(x) + E(x)^2$$

Jadi

$$V = E(x^2) - (E(x))^2$$

Persamaan terakhir ini sering kali merupakan cara yang paling efisien secara komputasi untuk menghitung varians, serta mudah digunakan untuk memperoleh rumus terkait lainnya. Persamaan ini menggambarkan varians sebagai "ekspektasi kuadrat dikurangi kuadrat ekspektasi". Ada beberapa hal lain yang perlu diperhatikan:

- Simpangan baku mengukur akar kuadrat dari rata-rata jarak kuadrat dari rata-rata, yang tidak sama dengan simpangan baku jarak (nonkuadrat); prosedur pemangkatan kuadrat yang terlibat dalam menghitung simpangan baku sedikit menekankan nilai yang lebih jauh dari rata-rata daripada simpangan mutlak. Oleh karena itu, dalam beberapa kasus simpangan mutlak (seperti yang dihitung oleh fungsi AVEDEV) mungkin lebih relevan sebagai ukuran penyebaran atau risiko.
- Simpangan baku memiliki satuan pengukuran yang sama dengan data yang mendasarinya (misalnya nilai dolar atau moneter, waktu, ruang, dll.), dan karenanya sering kali mudah ditafsirkan dalam pengertian fisik atau intuitif; varians tidak memiliki sifat ini (misalnya dalam konteks moneter, unitnya adalah dolar kuadrat).
- Selain menjadi ukuran risiko, deviasi standar adalah salah satu parameter yang menentukan distribusi Normal dan Lognormal (yang lainnya adalah rata-rata). Distribusi ini sangat penting dalam pemodelan keuangan, terutama yang berkaitan dengan pengembalian aset, harga saham, dan sebagainya.

- Dalam banyak konteks pasar keuangan, deviasi standar pengembalian aset (atau perubahan harga) dianggap sebagai ukuran inti risiko, dan sering disebut volatilitas. Ukuran lain, seperti nilai risiko, risiko sepihak, semi-deviasi, dan probabilitas gagal bayar juga penting, tergantung pada konteksnya.

Secara umum, seseorang hanya berurusan dengan sampel dari semua nilai yang mungkin (yaitu subset), sehingga seseorang mungkin perlu mempertimbangkan apakah deviasi standar sampel tersebut mewakili deviasi standar seluruh populasi tempat sampel diambil. Dengan kata lain, setiap ukuran statistik dari suatu sampel dapat dianggap sebagai estimasi dari populasi yang mendasarinya (bukan angka sebenarnya); data dalam sampel dapat dihasilkan oleh proses (populasi) mendasar lainnya yang memiliki rata-rata atau simpangan baku yang sedikit berbeda. Jadi, ada dua masalah yang mungkin perlu dipertimbangkan:

- Perhitungan estimasi titik yang tidak bias dari parameter populasi, ketika datanya adalah sampel.
- Perhitungan rentang nilai yang mungkin (interval keyakinan) untuk parameter ini (karena perhitungan berdasarkan data sampel hanya merupakan estimasi dari angka sebenarnya).

Dalam contoh ini, kami menunjukkan perhitungan estimasi varians dan simpangan baku, dan faktor koreksi terkait. Rumus untuk interval keyakinan di sekitar estimasi dasar memerlukan penggunaan distribusi invers dari distribusi Student (T) (untuk rata-rata), dan distribusi Chi-kuadrat (untuk simpangan baku), dan sebagainya dibahas kemudian dalam bab ini. Untuk perhitungan estimasi yang tidak bias:

- Rata-rata sampel adalah estimasi yang tidak bias dari populasi.
- Simpangan baku sampel adalah sedikit perkiraan yang lebih rendah dari populasi; faktor koreksi perkalian dari  $\sqrt{\frac{n}{n-1}}$  (di mana  $n$  adalah jumlah titik dalam sampel) diperlukan untuk memberikan estimasi nilai populasi yang tidak bias. Fungsi STDEV.S (atau STDEV lama) menghitung estimasi deviasi standar populasi dari data sampel, dan memiliki faktor koreksi yang dibangun dalam perhitungan. Di sisi lain, STDEV.P (atau STDEVP) menghitung deviasi standar dengan asumsi bahwa data tersebut adalah populasi penuh (sehingga tidak diperlukan istilah koreksi).
- Pendekatan yang sama tentu saja diperlukan saat mempertimbangkan varians (menggunakan VAR.S atau VAR.P), kecuali bahwa faktor koreksi adalah  $\frac{n}{n-1}$  alih-alih akar kuadratnya.
- Ukuran statistik lainnya (skewness dan kurtosis, lihat nanti) juga memerlukan faktor koreksi untuk diterapkan guna membuat estimasi yang tidak bias.



	A	B	C	D	E	F	G
1							
2		Month	Log Changes				
3			S&P	Item		Value	Formula
4		1	-1.4%	Variance		0.0013	=VAR.S(C4:C63)
5		2	-2.2%	Sqrt of Variance		3.58%	=SQRT(F4)
6		3	1.4%	Standard deviation		3.58%	=STDEV.S(C4:C63)
7		4	1.3%				

Gambar 5.11 Varians dan Deviasi Standar Pengembalian Berdasarkan Sampel Data

	A	B	C	D	E	F	G
1							
2		Month	Log Changes				
3			S&P	Item		Value	Formula
4		1	-1.4%	Variance		0.0013	=VAR.S(C4:C63)
5		2	-2.2%	Sqrt of Variance		3.58%	=SQRT(F4)
6		3	1.4%	Standard deviation		3.58%	=STDEV.S(C4:C63)
7		4	1.3%				
8		5	1.6%	Based on Sample Representing the Whole Population:			
9		6	3.1%	Variance		0.0013	=VAR.P(C4:C63)
10		7	2.4%	Standard deviation		3.55%	=STDEV.P(C4:C63)
11		8	2.1%	Ratios:			
12		9	0.5%	Variance		1.0169	=F4/F9
13		10	0.0%	Standard deviation		1.0084	=F6/F10
14		11	-3.1%	Cross-check			
15		12	1.2%	Sample Size, N		60	=COUNT(C4:C63)
16		13	1.1%	N/(N-1)		1.0169	=F15/(F15-1)
17		14	0.0%	Sqrt(N/(N-1))		1.0084	=SQRT(F16)

Gambar 5.12 Rekonsiliasi Statistik Sampel dan Populasi

Pada Gambar 21.12, kami menunjukkan klip layar dari berkas yang sama, yang juga menunjukkan angka-angka yang akan berlaku jika sampel tersebut mewakili seluruh populasi, serta perhitungan rekonsiliasi berdasarkan ukuran sampel. Gambar 5.13 menunjukkan contoh fungsi lain yang terkait dengan deviasi, termasuk fungsi DEVSQ (yang menjumlahkan kuadrat deviasi dari rata-rata, sehingga mirip dengan VAR.P setelah dibagi dengan ukuran sampel total, tanpa faktor koreksi), serta fungsi AVEDEV, yang mengambil rata-rata deviasi absolut dari rata-rata.

	A	B	C	D	E	F
1						
2						
3		Data		Result		Formula
4		-1.4%		0.001		=VAR.S(B4:B63)
5		-2.2%		3.58%		=STDEV.S(B4:B63)
6		1.4%				
7		1.3%		0.0013		=VAR.P(B4:B63)
8		1.6%		3.55%		=STDEV.P(B4:B63)
9		3.1%				
10		2.4%		0.0755		=DEVSQ(B4:B63)
11		2.1%		60		=COUNT(B4:B63)
12		0.5%		0.0013		=D10/D11
13		0.0%		3.55%		=SQRT(D10/D11)
14		-3.1%				
15		1.2%		2.6%		=AVEDEV(B4:B63)

Gambar 5.13 Contoh Fungsi DEVSQ dan AVEDEV

**Contoh: Skewness dan Kurtosis**

Koefisien skewness (atau hanya skewness atau skew) adalah ukuran ketidaksimetrisan (atau asimetri), yang didefinisikan sebagai:

$$\text{Koefisien dari skew} = \frac{\sum P_i(x_i - \mu)^3}{\sigma^3}$$

Pembilang mewakili rata-rata "pangkat tiga jarak dari rata-rata", dan penyebut adalah pangkat tiga deviasi standar, sehingga kemiringan adalah kuantitas non-dimensi (yaitu nilai numerik, dan tidak memiliki satuan, seperti dolar, waktu atau ruang). Untuk data populasi, fungsi SKEW.P dapat digunakan. Untuk data sampel, fungsi ini akan meremehkan kemiringan populasi; faktor koreksi perkalian yang tepat untuk memberikan estimasi kemiringan populasi yang tidak bias adalah  $\frac{n^2}{(n-1)(n-2)}$ .

Faktor ini dibangun ke dalam fungsi SKEW (saat ini tidak ada fungsi SKEW.S), yang karenanya memberikan estimasi yang tidak bias berdasarkan sampel. Meskipun ada beberapa aturan umum untuk menginterpretasikan kemiringan, interpretasi yang tepat dan umum sulit dilakukan karena mungkin ada pengecualian dalam beberapa kasus. Prinsip umum meliputi:

- Distribusi simetris akan memiliki kemiringan nol. Ini selalu benar, karena setiap nilai yang lebih besar dari rata-rata akan memiliki nilai yang mengimbangi tepat di bawah rata-rata; penyimpangannya di sekitar rata-rata akan saling meniadakan jika dipangkatkan ke pangkat ganjil, seperti saat dipangkatkan tiga.
- Kemiringan positif menunjukkan bahwa ekor berada di sisi kanan. Secara umum, ketika kemiringan di atas sekitar 0,3, ketidaksimetrisan distribusi terlihat jelas saat data ditampilkan dalam format grafis (histogram atau yang serupa). Kurtosis dihitung sebagai pangkat empat rata-rata jarak dari mean (dibagi dengan pangkat empat deviasi standar, menghasilkan kuantitas non-dimensi):

$$\text{Kurtosis} = \frac{\sum P_i(x_i - \mu)^4}{\sigma^4}$$

Kurtosis sulit ditafsirkan, tetapi dalam arti tertentu, ia memberikan uji sejauh mana distribusi memuncak di area pusat, sementara secara bersamaan memiliki ekor yang relatif gemuk. Distribusi normal memiliki kurtosis tiga; distribusi yang kurtosisnya sama dengan, lebih besar atau kurang dari tiga masing-masing dikenal sebagai mesokurtik, leptokurtik atau platikurtik.

Fungsi KURT Excel mengurangi tiga dari perhitungan standar untuk hanya menunjukkan kurtosis "berlebih":

$$\text{Pengecualian Kurtosis} = \frac{\sum P_i(x_i - \mu)^4}{\sigma^4} - 3$$

	A	B	C	D	E
1					
2					
3		Data	Result	Formula	
4		-1.4%	-0.754	=SKEW.P(B4:B63)	
5		-2.2%	-0.773	=SKEW(B4:B63)	
6		1.4%			
7		1.3%	1.913	=KURT(B4:B63)	
8		1.6%			

**Gambar 5.14 Contoh Fungsi SKEW.S dan KURT**

Untuk deviasi standar dan kemiringan, koreksi tersedia untuk memperkirakan kurtosis populasi dari kurtosis sampel dengan cara yang tidak bias. Namun, hal ini tidak disertakan dalam Excel pada saat penulisan.

**Contoh: Volatilitas Satu Sisi (Semi-deviasi)**

Seperti yang dibahas sebelumnya, fungsi tipe STDEV dapat digunakan untuk menghitung atau memperkirakan deviasi standar, dan karenanya volatilitas. Ukuran seperti itu sering kali sesuai sebagai indikasi risiko atau variasi umum, dan memang menjadi dasar dari banyak teori dan konsep standar inti dalam pengukuran risiko dan optimalisasi portofolio untuk aplikasi pasar keuangan. Di sisi lain, simpangan baku menggambarkan efek dari penyimpangan di sekitar rata-rata, baik yang menguntungkan maupun yang tidak menguntungkan.

Dalam beberapa perhitungan risiko, seseorang mungkin ingin hanya fokus pada hasil yang tidak menguntungkan (risiko negatif atau penurunan), dan tidak memasukkan hasil yang menguntungkan. Kasus ekstrem dari pendekatan semacam itu adalah ketika ukuran risiko yang digunakan hanya didasarkan pada kemungkinan hasil terburuk. Kemungkinan yang tidak terlalu ekstrem adalah menggunakan semi-deviasi; ini analog dengan simpangan baku, kecuali bahwa satu-satunya hasil yang dimasukkan dalam perhitungan adalah yang tidak menguntungkan relatif terhadap rata-rata.

Pada saat penulisan, tidak ada fungsi Excel untuk menghitung semi-deviasi dari kumpulan data secara langsung. Oleh karena itu, ada tiga pendekatan alternatif utama:

- Lakukan perhitungan langkah demi langkah yang eksplisit dalam rentang tabel di Excel (yaitu hitung rata-rata kumpulan data, lalu jumlahkan deviasi kuadrat dari setiap nilai yang lebih tinggi atau lebih rendah dari rata-rata, hitung berapa banyak item dalam setiap kategori, dan terakhir lakukan perhitungan akar kuadrat).
- Gunakan fungsi array untuk menghitung jumlah dan jumlah deviasi di atas atau di bawah rata-rata (yaitu tanpa harus menghitung nilai yang sesuai untuk setiap titik data secara eksplisit).
- Buat fungsi yang ditentukan pengguna yang menghitung nilai secara langsung dari data (dengan semua perhitungan terjadi dalam kode VBA, dan parameter untuk

menunjukkan apakah deviasi positif atau negatif akan disertakan dalam perhitungan). Seperti banyak fungsi yang ditentukan pengguna lainnya, pendekatan ini dapat membantu menghindari keharusan membuat tabel rumus perhitungan secara eksplisit di lembar Excel, memungkinkan referensi langsung dari kumpulan data ke statistik terkait dan memungkinkan adaptasi rumus yang lebih cepat jika ukuran kumpulan data berubah.

Gambar 5.15 menunjukkan contoh perhitungan langkah demi langkah yang eksplisit dan penggunaan rumus array. Klip layar pada Gambar 5.15 menunjukkan pendekatan menggunakan rumus array.

Perlu dicatat bahwa definisi klasik dari semi-deviasi didasarkan pada deviasi titik-titik dari nilai rata-ratanya (average). Akan tetapi, mudah untuk menggeneralisasi hal ini, sehingga deviasi diukur sehubungan dengan angka referensi lainnya (seperti pengembalian minimum yang dapat diterima), dengan menggunakan angka ini dalam perhitungan, sebagai pengganti nilai rata-rata.

	A	B	C	D	E	F	G	H
1								
2		Data	Mean		0.89%			
3		-1.4%						
4		-2.2%						
5		1.4%		Sum of squared deviations (array function)	0.0	0.0		=SUM(IF(B3:B12>=\$E\$2,(B3-B12-\$E\$2)^2,0))
6		1.3%		Count (array function)	4	6		=SUM(IF(B3:B12>=\$E\$2,1,0))
7		1.6%		Ratio	0.0	0.0		=F5/F6
8		3.1%		Square root (=Semi-dev)	1.98%	1.27%		=SQRT(F7)
9		2.4%						
10		2.1%						
11		0.5%						
12		0.0%						

**Gambar 5.15 Menggunakan Rumus Array untuk Menghitung Semi-Deviasi dari Kumpulan Data Pengembalian**

#### 5.4 APLIKASI PRAKTIS: HUBUNGAN SAMA DAN KETERGANTUNGAN

Baik dalam analisis data maupun pemodelan, penting untuk mengeksplorasi hubungan apa yang mungkin ada antara variabel; diagram X–Y (atau sebaran) adalah titik awal yang baik untuk analisis tersebut, karena inspeksi visual membantu dalam pengembangan hipotesis tentang kemungkinan hubungan, seperti:

- Tidak ada hubungan yang tampak dalam bentuk apa pun, dengan titik-titik tersebar secara acak.
- Hubungan linier umum tetapi dengan beberapa variasi (yang belum) dijelaskan (atau acak) di sekitarnya. Hubungan semacam itu bisa bersifat positif (peningkatan nilai satu variabel umumnya dikaitkan dengan peningkatan nilai variabel lain) atau negatif. Jika hubungan tampak bersifat cukup linier, seseorang juga dapat memilih untuk meminta Excel menunjukkan garis regresi, persamaannya, dan statistik terkait.
- Jenis hubungan yang lebih kompleks, seperti kurva U, atau hubungan yang menunjukkan adanya hubungan erat antara variabel di bagian rentang, tetapi

hubungan yang lebih longgar di bagian rentang lainnya.

Perhatikan bahwa, saat membuat diagram sebar X–Y untuk tujuan eksplorasi, belum perlu ada gagasan tentang variabel dependen atau independen (meskipun biasanya variabel yang dianggap lebih mungkin independen akan dipilih untuk ditempatkan pada sumbu x).

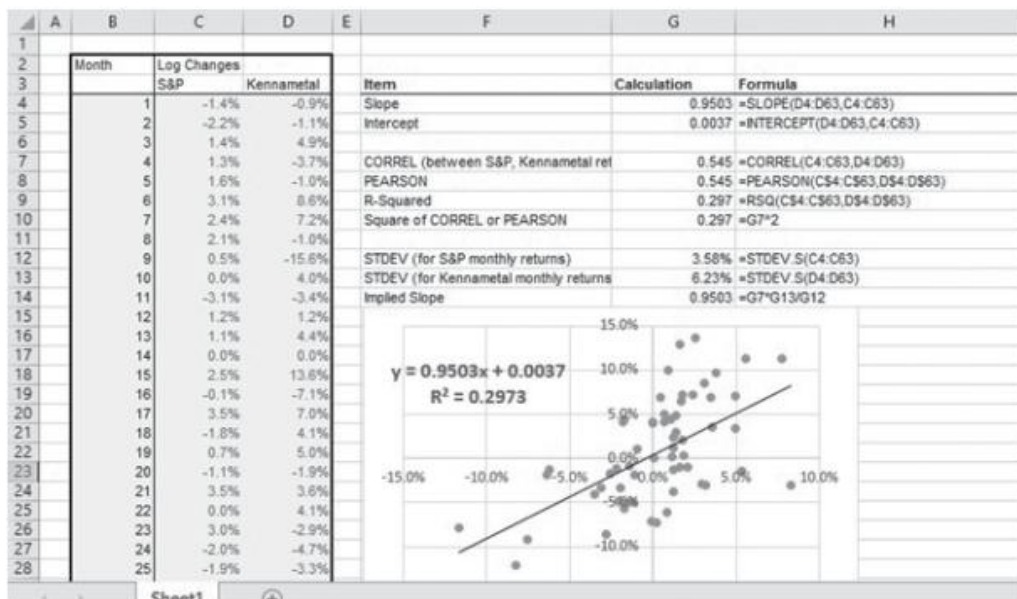
### Contoh: Diagram Sebar (Diagram X–Y) dan Mengukur Korelasi

Gambar 5.16 menunjukkan contoh diagram sebar (Gambar 5.16); beberapa statistik terkait ditempatkan pada diagram dengan mengklik kanan pada titik data mana pun pada diagram (untuk memanggil menu peka konteks di Excel):

- Fungsi SLOPE dan INTERCEPT digunakan untuk menghitung kemiringan garis regresi (yaitu yang pada prinsipnya harus sama dengan nilai yang ditunjukkan oleh persamaan diagram).
- Metode produk atau Pearson digunakan untuk menghitung korelasi "linier", menggunakan salah satu (atau keduanya) fungsi CORREL atau PEARSON. Fungsi RSQ menghitung kuadratnya.
- Fungsi STDEV.S menghitung deviasi standar setiap set data.
- Kemiringan yang tersirat dengan menggunakan hubungan matematis antara koefisien korelasi dan deviasi standar. Yaitu:

$$\text{Slope} = \frac{\rho_{xy}\sigma_y}{\sigma_x}$$

Tentu saja, kemiringan garis juga menggambarkan seberapa besar nilai y akan bergerak jika nilai x berubah satu satuan. Oleh karena itu, persamaan di atas menunjukkan bahwa jika nilai x diubah sebesar  $\sigma_x$  maka nilai y akan berubah sebesar  $\rho_{xy}\sigma_y$ .



Gambar 5.16 Menggunakan Scatter Plot dan Hubungan Antara Kemiringan dan Korelasi

**Contoh: Lebih Lanjut tentang Koefisien Korelasi dan Korelasi Peringkat**

Contoh di atas menunjukkan penggunaan fungsi CORREL dan PEARSON untuk menghitung korelasi (linier, produk atau PEARSON) antara variabel. Dalam istilah matematika, ( $\rho$ ) ini didefinisikan sebagai:

$$\rho = \frac{\Sigma(x - \mu_x)(y - \mu_y)}{\sqrt{\Sigma(x - \mu_x)^2 \Sigma(y - \mu_y)^2}}$$

di mana  $x$  dan  $y$  mewakili nilai-nilai individual dari masing-masing set data, dan  $\mu_x$  mewakili rata-rata (nilai tengah atau nilai yang diharapkan) dari set data  $X$  (dan juga  $\mu_y$  untuk set data  $Y$ ). Dari rumus tersebut, kita dapat melihat bahwa:

- Korelasi antara dua set data dapat dihitung jika setiap set memiliki jumlah titik yang sama.
- Koefisien adalah kuantitas non-dimensi yang terletak antara  $-1$  dan  $1$  (dan biasanya dinyatakan sebagai persentase antara  $-100\%$  dan  $100\%$ ).
- Korelasi menangkap pergerakan simultan setiap variabel relatif terhadap nilai rata-ratanya sendiri, yaitu mengukur korelasi antara dua proses dalam arti apakah, ketika diamati secara simultan, masing-masing cenderung mengambil nilai yang berada di atas atau di bawah nilai rata-ratanya sendiri, atau apakah tidak ada hubungan antara kemunculan nilai relatif tersebut: dengan mempertimbangkan pembilang dalam rumus di atas, jelas bahwa satu nilai,  $x$ , dan padanannya,  $y$ , akan berkontribusi positif pada kalkulasi korelasi ketika kedua nilai berada di atas atau di bawah nilai rata-ratanya masing-masing (sehingga pembilangnya positif, yaitu hasil perkalian dua angka positif atau dua angka negatif).
- Koefisien korelasi tidak akan berubah jika konstanta ditambahkan ke setiap item dalam salah satu set data, karena efek konstanta tersebut akan dihilangkan selama kalkulasi (karena nilai rata-rata setiap set data dikurangi dari setiap nilai).
- Koefisien korelasi tidak akan berubah jika setiap item dalam kumpulan data dikalikan dengan suatu konstanta (karena efek dari hal tersebut akan sama pada pembilang dan penyebut).

Dari perspektif pemodelan, keberadaan koefisien korelasi (yang signifikan secara statistik) tidak menyiratkan adanya ketergantungan langsung antara item-item tersebut (dalam arti arah atau kausalitas dari satu item ke item lainnya). Sebaliknya, variasi pada setiap item mungkin didorong oleh suatu faktor yang tidak eksplisit atau diketahui, tetapi yang menyebabkan setiap item bervariasi, sehingga keduanya tampak bervariasi secara bersamaan.

Misalnya, korelasi akan ditunjukkan antara perubahan nilai pasar dari dua produk akhir yang berasal dari minyak; biaya produksi masing-masing sebagian akan ditentukan oleh harga minyak, tetapi akan ada juga beberapa komponen biaya independen lainnya, serta faktor-faktor lain (di luar item biaya) yang memengaruhi harga pasar masing-masing.

Lebih jauh, koefisien yang diukur umumnya tunduk pada kesalahan statistik yang tinggi,

dan mungkin tidak signifikan secara statistik atau tidak stabil; kumpulan data yang cukup besar diperlukan untuk mengurangi kesalahan tersebut. Meskipun pengukuran korelasi yang digunakan di atas (yaitu metode linear, produk atau Pearson, menggunakan CORREL atau PEARSON) adalah yang paling umum untuk tujuan umum, ada beberapa cara lain untuk menentukan kemungkinan ukuran korelasi:

- Metode peringkat atau Spearman (kadang-kadang disebut korelasi "non-linear"), yang melibatkan penggantian setiap nilai dengan bilangan bulat yang mewakili posisinya (atau peringkat) dalam set datanya sendiri (yaitu ketika tercantum dalam urutan menaik atau menurun), dan menghitung korelasi linear dari set bilangan bulat ini. Perhatikan bahwa ukuran korelasi ini menunjukkan hubungan yang kurang ketat antara variabel. Misalnya, dua variabel yang plot sebarannya menunjukkan tren peningkatan umum yang bukan garis lurus yang sempurna dapat memiliki korelasi peringkat yang 100%. (Definisi yang lebih longgar ini memungkinkan fleksibilitas tambahan dalam aplikasi seperti pembuatan dan pengambilan sampel distribusi multivariat berkorelasi saat menggunakan teknik simulasi.) Pada saat penulisan tidak ada fungsi Excel langsung untuk menghitung korelasi peringkat (Spearman); ini dapat dilakukan dengan menggunakan langkah-langkah individual yang eksplisit, atau rumus array atau fungsi yang ditentukan pengguna VBA.
- Koefisien tau Kendall. Ini menggunakan peringkat titik data, dan dihitung dengan memperoleh jumlah peringkat pasangan yang sesuai. Yaitu, dua titik (masing-masing dengan koordinat x-y) dianggap sesuai jika perbedaan peringkat nilai-x mereka memiliki tanda yang sama dengan perbedaan peringkat nilai-y mereka. Perhatikan bahwa perhitungan mengharuskan seseorang untuk membandingkan peringkat setiap titik dengan peringkat setiap titik lainnya. Oleh karena itu, jumlah operasi sebanding dengan kuadrat jumlah titik. Sebaliknya, untuk metode korelasi Pearson dan peringkat, jumlah operasi berskala linier dengan jumlah titik data, karena hanya deviasi setiap titik dari rata-rata set datanya sendiri yang memerlukan perhitungan.

Gambar 5.17 menunjukkan contoh perhitungan korelasi peringkat antara dua set data. Gambar 5.17 memperlihatkan perhitungan eksplisit di Excel, yang pertama-tama melibatkan perhitungan peringkat nilai (Kolom F dan G memperlihatkan peringkat, dihitung dari data mentah di Kolom B dan C), sebelum fungsi CORREL diterapkan (sel G14). Lebih jauh, rumus array di Sel C16 memperlihatkan bagaimana perhitungan yang sama dapat dibuat langsung dari data mentah tanpa menghitung nilai peringkat secara eksplisit:

$$C16 = \{CORREL(RANK.AVG(B3: B12, B\$3: B\$12,1), RANK.AVG(C3 : C12, C\$3: C\$12,1))\}$$

#### **Contoh: Mengukur Kovariansi**

Kovariansi antara dua set data terkait erat dengan korelasi di antara keduanya:

$$\text{Covar}(X, Y) = \Sigma(x - \mu_x)(y - \mu_y)$$

Atau

$$\text{Covar}(X, Y) = \rho\sigma_x\sigma_y$$

	A	B	C	D	E	F	G	H
1						Explicit Calculation Steps		
2	X	Y				R(X)	R(Y)	
3		0.059	0.084			1	3	=RANK.AVG(C3,C\$3:C\$12,1)
4		0.339	0.061			7	1	=RANK.AVG(C4,C\$3:C\$12,1)
5		0.165	0.072			3	2	=RANK.AVG(C5,C\$3:C\$12,1)
6		0.226	0.206			4	6	=RANK.AVG(C6,C\$3:C\$12,1)
7		0.266	0.095			5	4	=RANK.AVG(C7,C\$3:C\$12,1)
8		0.095	0.153			2	5	=RANK.AVG(C8,C\$3:C\$12,1)
9		0.530	0.384			10	10	=RANK.AVG(C9,C\$3:C\$12,1)
10		0.445	0.281			8	7	=RANK.AVG(C10,C\$3:C\$12,1)
11		0.500	0.336			9	8	=RANK.AVG(C11,C\$3:C\$12,1)
12		0.318	0.336			6	9	=RANK.AVG(C12,C\$3:C\$12,1)
13								
14	Correl of Data		74%	=CORREL(B3:B12,C3:C12)			60%	
15								
16	Correl of Ranked Data		60%	(=CORREL(RANK.AVG(B3:B12,B\$3:B\$12,1),RANK.AVG(C3:C12,C\$3:C\$12,1)))				
17								

**Gambar 5.17** Perhitungan Korelasi Berperingkat Menggunakan Langkah Eksplisit dan Rumus Array

Jelas dari rumus-rumus ini (dan rumus-rumus yang berkaitan dengan simpangan baku) bahwa:

- Penskalaan (atau pembobotan) semua titik dalam satu set data akan menghasilkan set data baru yang kovariansinya berskala dengan cara yang sama:

$$\text{Covar}(wX, Y) = \rho w\sigma_x\sigma_y$$

- Kovariansi suatu proses dengan dirinya sendiri sama saja dengan variansinya (karena korelasinya adalah 100%):

$$\text{Covar}(X, X) = \sigma_x\sigma_x = V(x)$$

Fungsi COVARIANCE.S (COVAR dalam versi sebelumnya) menghitung kovariansi sampel (yaitu estimasi kovariansi populasi, dengan asumsi bahwa data yang diberikan hanya berupa sampel populasi), dan COVARIANCE.P menghitung kovariansi populasi dari dua set data yang mewakili keseluruhan populasi.

#### **Contoh: Matriks Kovariansi, Volatilitas Portofolio, dan Skala Waktu Volatilitas**

Jika terdapat data pada lebih dari dua variabel, matriks korelasi penuh dapat dihitung. Setiap elemen ini hanyalah korelasi antara variabel yang sesuai dengan baris dan kolom matriks (yaitu Baris 1 matriks berhubungan dengan variabel pertama, seperti halnya Kolom 1, dst.) Jelas dan langsung terlihat dari rumus yang mendefinisikan koefisien korelasi bahwa:

- Elemen diagonal matriks korelasi sama dengan satu (atau 100%), karena setiap item berkorelasi sempurna dengan dirinya sendiri.



- Matriksnya simetris: set data X dan Y memiliki peran yang sama dan dapat dipertukarkan satu sama lain tanpa mengubah koefisien yang dihitung.

Serupa dengan itu, seseorang dapat membuat matriks kovarians, di mana setiap elemen setara dengan korelasi yang sesuai dikalikan dengan deviasi standar variabel yang terkait.

	A	B	C	D	E	F	G	H
1								
2		Month	Log Changes					
3			S&P	Kennametal	Item	Calculation	Formula	
4		1	-1.4%	-0.9%	Covariance (formula)	0.00122	=COVARIANCE.S(C54:C563,D54:D563)	
5		2	-2.2%	-1.1%	Covariance (formula)	0.00120	=COVARIANCE.P(C54:C563,D54:D563)	
6		3	1.4%	4.9%				
7		4	1.3%	-3.7%	CORREL (between S&P, Kennametal ref	0.545	=CORREL(C4:C63,D4:D63)	
8		5	1.6%	-1.0%	STDEV (for S&P monthly returns)	3.58%	=STDEV.S(C4:C63)	
9		6	3.1%	8.6%	STDEV (for Kennametal monthly returns)	6.23%	=STDEV.S(D4:D63)	
10		7	2.4%	7.2%	Covariance (implied)	0.00122	=G7*G8*G9	
11		8	2.1%	-1.0%				
12		9	0.5%	-15.6%				
13		10	0.0%	-4.0%				

**Gambar 5.18 Perhitungan Kovariansi Antara Dua Kumpulan Data Menggunakan Berbagai Fungsi**

Aplikasi korelasi dan matriks kovariansi yang sangat penting adalah dalam analisis portofolio. Secara khusus, portofolio yang terdiri dari dua aset dengan volatilitas yang sama akan kurang volatil daripada portofolio dengan nilai total yang sama yang hanya terdiri dari satu aset kecuali aset tersebut berkorelasi sempurna. Hal ini semata-mata karena akan muncul hasil di mana satu aset memiliki nilai yang lebih tinggi sementara yang lain memiliki nilai yang lebih rendah (atau sebaliknya), sehingga menciptakan efek pemusatan (diversifikasi) ketika nilai-nilai tersebut ditambahkan bersama-sama. Semakin rendah korelasi antara aset, semakin kuat efek ini. Perhatikan bahwa jika suatu variabel, X, adalah jumlah dari proses (atau aset) lainnya (fundamental atau yang mendasarinya):

$$X = Y_1 + \dots + Y_n$$

Kemudian

$$V(X) = \sum_{i=1}^n \sum_{j=1}^n Cov(Y_i, Y_j)$$

di mana Cov mewakili (definisi) kovariansi antara Y. Artinya, varians portofolio adalah jumlah semua kovariansi antara proses. Oleh karena itu, jika portofolio disusun dari serangkaian proses (atau aset) yang mendasarinya yang tertimbang:

$$X = w_1 Y_1 + \dots + w_n Y_n$$

Kemudian

$$V(X) = \sum_{i=1}^n \sum_{j=1}^n Cov(w_i Y_i, w_j Y_j)$$

Jadi

$$V(X) = \sum_{i=1}^n \sum_{j=1}^n w_i w_j Cov(Y_i, Y_j)$$

Secara praktis, ketika menghitung varians (atau deviasi standar) pengembalian portofolio, seseorang mungkin diberikan data tentang pengembalian komponen yang mendasarinya, bukan yang tertimbang. Misalnya, ketika memegang portofolio beberapa saham individual dalam proporsi yang berbeda (misalnya campuran Vodafone, Apple, Amazon, Exxon, . . .), data pengembalian yang diberikan langsung oleh umpan data akan berhubungan dengan saham individual. Oleh karena itu, tergantung pada keadaan, mungkin lebih mudah untuk menggunakan salah satu atau beberapa rumus di atas.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1														
2				Variance of Returns					Weights				FROM UNDERLYING ASSETS WITH WEIGHTING	
3				Asset 1	Asset 2	Asset 3			Asset 1	Asset 2	Asset 3		Variance	
4				0.00176	0.00140	0.00151	=VAR.S(F8:F207)		20%	60%	20%		0.00088	=MMULT(I4:K4,MMULT(I8:K10,TRANSPOSE(I4:K4)))
5														
6				Returns Per Period					COVARIANCE.S					
7	Period			Asset 1	Asset 2	Asset 3		Asset 1	Asset 2	Asset 3				
8	1			3.24%	4.75%	6.09%		0.00176	0.00053	0.00017		=COVARIANCE.S(F8:F207,D8:D207)		
9	2			-0.45%	3.89%	8.16%		0.00053	0.00140	0.00044		=COVARIANCE.S(F8:F207,E8:E207)		
10	3			2.18%	5.53%	7.17%		0.00017	0.00044	0.00151		=COVARIANCE.S(F8:F207,F8:F207)		
11	4			1.55%	-2.34%	1.25%								
12	5			3.40%	3.34%	-2.70%								
13	6			4.01%	0.87%	4.88%								

**Gambar 5.19 Perhitungan Volatilitas Portofolio Berdasarkan Data Aset Dasar**

Gambar 5.19 berisi contoh setiap pendekatan. Kumpulan data pengembalian untuk tiga aset dasar terdapat di Kolom D:F. Gambar 5.19 menunjukkan perhitungan di mana data pengembalian diambil dari aset dasar, dan menggunakan bobot portofolio secara eksplisit dalam rumus (array) untuk varians (sel M4):

$$M4 = \text{MMULT}(14: K4, \text{MMULT}(18: K10, \text{TRANSPOSE}(14: K4)))$$

Di sisi lain, Gambar 5.20 menunjukkan perhitungan di mana pengembalian untuk aset tertimbang (yaitu komposisi portofolio aktual) dihitung, sehingga varians (sel U4) hanyalah jumlah elemen matriks kovariansi yang sesuai:

$$U4 = \text{SUM}(V8: X10)$$

	A	B	C	D	E	F	G	O	P	Q	R	S	T	U	V	W	X	Y	Z	AA	AB
1																					
2																					
3																					
4																					
5																					
6																					
7																					
8																					
9																					
10																					
11																					
12																					
13																					

**Gambar 5.20** Perhitungan Volatilitas Portofolio Berdasarkan Data Aset Tertimbang

Perhatikan bahwa, jika kovariansi antara proses adalah nol (seperti halnya proses independen dalam teori untuk set data besar), maka varians jumlah mereka sama dengan jumlah varians (yaitu jumlah kovariansi setiap proses dengan dirinya sendiri). Jadi, dalam kasus khusus deret waktu di mana pengembalian dalam setiap periode independen dari pengembalian dalam periode lain, varians dari total pengembalian akan menjadi jumlah varians pengembalian dalam periode individual. Jika varians ini konstan dari waktu ke waktu (yaitu sama dalam setiap periode), total varians pengembalian akan meningkat secara linear seiring waktu, sehingga deviasi standar akan berskala dengan akar kuadrat waktu:

$$\sigma_T = \sigma\sqrt{T}$$

di mana  $\sigma$  adalah deviasi standar dari pengembalian periodik individual dan  $\sigma_T$  adalah untuk periode  $T$ . Kita dapat menggunakan ini untuk mengonversi antara volatilitas tahunan, bulanan, atau harian.

## 5.5 APLIKASI PRAKTIS: DISTRIBUSI KEMUNGKINAN

Penggunaan distribusi frekuensi (kemungkinan) pada dasarnya adalah cara meringkas seluruh rangkaian nilai yang mungkin untuk proses yang tidak pasti (acak atau berisiko) dengan menunjukkan kemungkinan relatif (atau pembobotan) dari setiap nilai. Penting untuk membedakan dengan benar nilai-nilai suatu proses (sumbu-x) dan kemungkinan (atau kemungkinan kumulatif) yang terkait dengan setiap nilai (sumbu-y). Dengan demikian, ada dua jenis fungsi yang berbeda:

- Proses "X-ke-P", di mana, dari nilai masukan yang diberikan, kita menentukan kemungkinan bahwa nilai yang kurang (atau lebih) dari ini akan terjadi (atau kemungkinan relatif bahwa nilai yang tepat akan terjadi). Fungsi seperti NORM.DIST (NORMDIST sebelum Excel 2010) berbentuk seperti ini.
- Proses "P-ke-X", di mana probabilitas merupakan input, dan seseorang ingin menemukan nilai (persentil) yang sesuai yang terkait dengan persentase tersebut. Ini sama dengan menginversikan distribusi frekuensi. Di Excel, fungsi seperti NORM.INV (atau NORMINV) berbentuk seperti ini. Area aplikasi utamanya adalah pengambilan sampel acak, pengujian hipotesis, dan interval kepercayaan kalkulasi. Fungsi tipe

PERCENTILE bersifat analog, tetapi diterapkan pada kumpulan data titik individu atau sampel, bukan pada fungsi distribusi.

Di bagian ini, kami menyediakan beberapa contoh penggunaan beberapa distribusi ini (presentasi terperinci berada di luar cakupan teks ini; Risiko Bisnis dan Pemodelan Simulasi dalam Praktik karya penulis menyediakan analisis terperinci dari sekitar 20 distribusi utama yang tersedia secara langsung, atau dapat dibuat dengan mudah, di Excel). Fungsi distribusi X-to-P (atau definisi standar) Excel meliputi:

- BETA.DIST menghitung distribusi beta dalam bentuk kepadatan atau kumulatif (BETA.DIST dalam versi sebelumnya hanya menyediakan bentuk kumulatif).
- BINOM.DIST (BINOMDIST dalam versi sebelumnya) menghitung kepadatan atau bentuk kumulatif dari distribusi binomial; BINOM.DIST.RANGE (Excel 2013 dan seterusnya) menghitung probabilitas dari rentang hasil yang ditentukan.
- CHISQ.DIST.RT menghitung probabilitas satu sisi kanan dari distribusi Chi-kuadrat, baik dalam bentuk kepadatan maupun kumulatif (CHIDIST dalam versi sebelumnya hanya menghitung bentuk kumulatif). Demikian pula, CHISQ.DIST menghitung probabilitas satu sisi kirinya.
- EXPON.DIST (EXPONDIST dalam versi sebelumnya) menghitung kepadatan atau probabilitas kumulatif untuk distribusi eksponensial.
- F.DIST.RT menghitung bentuk ekor kanan dari distribusi F dalam bentuk kepadatan atau kumulatif (FDIST dalam versi sebelumnya hanya menghitung bentuk kumulatif). Demikian pula, F.DIST menghitung bentuk ekor kiri.
- GAMMA.DIST (GAMMADIST pada versi sebelumnya) menghitung distribusi gamma dalam bentuk kumulatif atau kerapatan.
- HYPGEOM.DIST menghitung probabilitas untuk distribusi hipergeometrik baik dalam bentuk kerapatan maupun kumulatif (HYPGEOMDIST pada versi sebelumnya hanya menghitung probabilitas kumulatif yang terkait dengan suatu nilai).
- LOGNORM.DIST menghitung kerapatan atau probabilitas kumulatif untuk distribusi Lognormal berdasarkan parameterisasi logaritmik (LOGNORMDIST pada versi sebelumnya hanya menghitung probabilitas kumulatif).
- NEGBINOM.DIST menghitung probabilitas untuk distribusi binomial negatif dalam bentuk kerapatan atau kumulatif (NEGBINOMDIST pada versi sebelumnya hanya menghitung probabilitas kumulatif).
- NORM.DIST dan NORM.S.DIST (NORMDIST dan NORMSDIST pada versi sebelumnya) masing-masing menghitung distribusi kumulatif normal umum dan standar. PHI (Excel 2013 dan seterusnya) menghitung nilai kerapatan untuk distribusi normal standar GAUSS (Excel 2013 dan seterusnya) menghitung 0,5 kurang dari distribusi kumulatif normal standar.
- POISSON.DIST (POISSON dalam versi sebelumnya) menghitung kerapatan atau bentuk kumulatif dari distribusi Poisson.
- PROB menggunakan serangkaian nilai diskrit dan probabilitas terkait untuk

menghitung probabilitas bahwa nilai tertentu terletak di antara batas bawah dan atas yang ditentukan (yaitu dalam rentang yang ditentukan oleh batas-batas ini).

- T.DIST.2T (TDIST dalam versi sebelumnya) menghitung probabilitas dua sisi untuk distribusi Student (T). Demikian pula, T.DIST.RT menghitung probabilitas sisi kanan, dan T.DIST menghitung probabilitas sisi kiri.
- WEIBULL.DIST (WEIBULL dalam versi sebelumnya) menghitung kepadatan dan bentuk kumulatif dari probabilitas nilai yang terkait dengan distribusi Weibull.

Fungsi distribusi P-ke-X (invers atau persentil) Excel yang tersedia secara langsung adalah:

- BINOM.INV (sebelum Excel 2007, tidak ada fungsi yang setara, yaitu "BINOMINV" tidak ada tetapi fungsi serupa CRITBINOM tersedia).
- NORM.INV (atau NORMINV) untuk distribusi Normal umum dan NORM.S.INV atau NORMSINV untuk distribusi Normal standar (dengan mean 0 dan deviasi standar 1).
- LOGNORM.INV (atau LOGINV) menghitung invers dari distribusi kumulatif Lognormal (berdasarkan parameter logaritmik, bukan parameter alami).
- BETA.INV (atau BETAINV) untuk distribusi Beta.
- GAMMA.INV (atau GAMMAINV) untuk distribusi Gamma.
- T.INV.2T (TINV pada versi sebelumnya) menghitung kebalikan dari distribusi T (Siswa) dua arah. T.INV mengembalikan arah kiri (perhatikan modifikasi sintaksis antar versi).
- CHISQ.INV.RT (CHIINV pada versi sebelumnya) menghitung kebalikan dari arah kanan untuk distribusi Chi-kuadrat. CHISQ.INV menghitung kebalikan dari arah kiri (sekali lagi, perhatikan modifikasi sintaksis).
- F.INV.RT (FINV pada versi sebelumnya) menghitung kebalikan dari arah kanan dari distribusi F. F.INV mengembalikan kebalikan dari distribusi arah kiri (perhatikan modifikasi sintaksis).

#### **Contoh: Kemungkinan Sejumlah Keberhasilan Proses Eksplorasi Minyak**

Proses Bernoulli (atau distribusi) adalah proses yang hanya memiliki dua kemungkinan hasil dari percobaan yang dilakukan satu kali (sering kali dicirikan sebagai keberhasilan/kegagalan, atau kepala/ekor, atau 0/1). Proses binomial adalah generalisasi, yang mungkin memiliki beberapa percobaan, yang masing-masing independen terhadap yang lain, dan yang masing-masing memiliki probabilitas kejadian yang sama.

Contoh proses binomial meliputi beberapa kali pelemparan koin, pelaksanaan serangkaian kegiatan pengeboran minyak di area geologi yang terpisah (di mana setiap pengeboran dapat berhasil atau gagal), dan seterusnya. Dalam konteks seperti itu, seseorang dapat mengajukan pertanyaan seperti:

- Berapa kemungkinan akan ada tepat tiga keberhasilan?
- Berapa kemungkinan akan ada nol, satu, dua atau tiga keberhasilan (yaitu tiga atau kurang)?
- Berapa kemungkinan bahwa jumlah keberhasilan akan berada di antara 2 dan 5?

Gambar 5.21 dan gambar 5.22 berisi contoh fungsi BINOM. DIST dan BINOM.DIST.RANGE untuk menjawab pertanyaan tersebut (lihat lembar kerja terpisah untuk setiap fungsi dalam

file), dengan cuplikan layar masing-masing ditampilkan pada Gambar 5.21 dan Gambar 5.22. Perhatikan bahwa fungsi BINOM.DIST.RANGE lebih umum, karena dapat menghasilkan bentuk ekuivalen kepadatan dengan menyetel kedua argumen angka agar sama atau bentuk kumulatif dengan menyetel argumen angka bawah ke nol atau bentuk rentang dengan memilih argumen angka bawah dan atas secara tepat. Perhatikan juga bahwa urutan parameter yang diperlukan tidak sama untuk kedua fungsi tersebut.

	A	B	C	D
1				
2			<b>Binom.Dist Density</b>	<b>Binom.Dist Cumulative</b>
3		Number_s (number of successes)	3	3
4		Trials (number of trials)	10	10
5		Probability_s (probability of success)	30%	30%
6		Cumulative	0	1
7		Result	26.7%	65.0%
8		Formulae in row 7	=BINOM.DIST(C3,C4,C5,C6)	=BINOM.DIST(D3,D4,D5,D6)

**Gambar 21.21 Penggunaan Fungsi BINOM.DIST Dalam Bentuk Kepadatan Dan Kumulatif**

	A	B	C	D	E
1					
2			<b>Binom.Dist.Range Single Point</b>	<b>Binom.Dist.Range Range from Lower</b>	<b>Binom.Dist.Range with Range</b>
3		Trials (number of trials)	10	10	10
4		Probability_s (probability of success)	30%	30%	30%
5		Number_s (number of successes)	3	0	2
6		Number_s2 (upper number of successes)	3	3	5
7		Result	26.7%	65.0%	80.3%
8		Formulae in row 7	=BINOM.DIST.RANGE(C3,C4,C5,C6)	=BINOM.DIST.RANGE(D3,D4,D5,D6)	=BINOM.DIST.RANGE(E3,E4,E5,E6)

**Gambar 21.22 Penggunaan Fungsi BINOM.DIST.RANGE Dalam Bentuk Kepadatan, Kumulatif, Dan Rentang**

#### Contoh: Frekuensi Hasil dalam Satu atau Dua Deviasi Standar

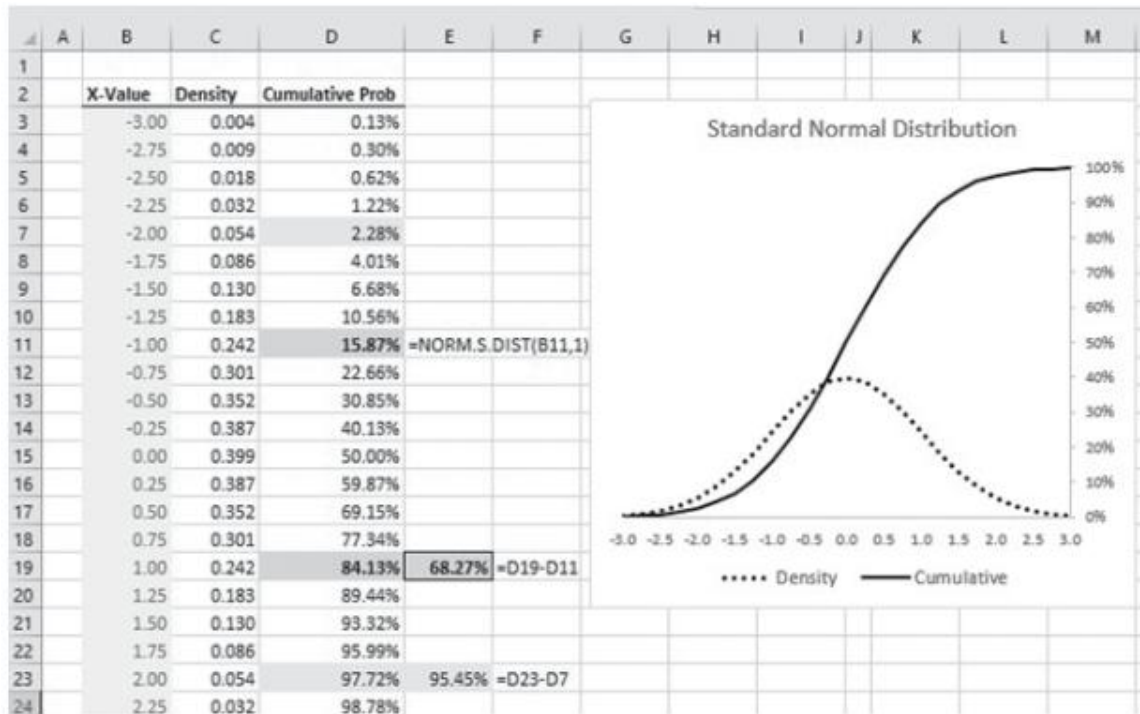
Fungsi NORM.S.DIST dapat digunakan untuk menghitung bentuk kepadatan dan kumulatif dari distribusi normal standar (dengan rata-rata sama dengan nol dan deviasi standar sama dengan satu), dengan menggunakan parameter opsional (argumen) yang sesuai. Gambar 5.23 menunjukkan penggunaan fungsi untuk menghitung kurva kepadatan dan probabilitas kumulatif untuk rentang nilai.

Dengan mengurangi probabilitas kumulatif titik yang satu deviasi standar lebih besar dari rata-rata dari probabilitas kumulatif titik yang satu deviasi standar di bawah rata-rata, kita dapat melihat (Sel E19) bahwa sekitar 68,27% hasil untuk distribusi normal berada dalam rentang yang satu deviasi standar di kedua sisi rata-rata. Demikian pula, sekitar 95,45% hasil berada dalam rentang yang berada di kedua sisi rata-rata dengan dua deviasi standar (Sel E23) (lihat Gambar 5.23).

#### Contoh: Membuat Sampel Acak dari Distribusi Probabilitas

Fungsi P-ke-X atau invers (persentil) dapat digunakan untuk membuat sampel acak dari distribusi mana pun yang menyediakan fungsi invers. Pada dasarnya, seseorang menggunakan

sampel acak yang diambil dari proses acak seragam standar (yaitu rentang kontinu antara nol dan satu) untuk menentukan persentase, yang kemudian digunakan untuk menghitung nilai persentil yang sesuai. Karena persentase dipilih secara seragam, nilai persentil yang dihasilkan akan muncul dengan frekuensi yang benar.



Gambar 5.23 Distribusi Normal dan Frekuensi Hasil dalam Rentang di Sekitar Rata-rata

	A	B	C	D	E
1					
2		<b>Item</b>	<b>Result</b>	<b>Formulae</b>	
3		Random Percentage	85.6%	=RAND()	
4		Random Sample	1.06	=NORM.S.INV(C3)	

Gambar 5.24 Pembuatan Sampel Acak dari Distribusi Normal Standar

Gambar 5.24 menunjukkan contoh di mana fungsi RAND() Excel digunakan untuk menghasilkan probabilitas persentase acak, dan fungsi NORM.S.INV digunakan untuk menemukan nilai distribusi normal standar yang sesuai dengan probabilitas kumulatif tersebut. Distribusi normal standar memiliki rata-rata nol dan deviasi standar satu; Pembuatan sampel dari distribusi normal lain dapat dilakukan dengan mengalikan sampel dari distribusi normal baku dengan deviasi baku yang diinginkan dan menambahkan nilai rata-rata, atau dengan menerapkan proses inversi yang sama pada fungsi NORM.INV yang menggunakan nilai rata-rata dan deviasi baku sebagai parameternya.

#### Contoh: Fungsi Invers yang Ditentukan Pengguna untuk Pengambilan Sampel Acak

Fungsi invers yang disediakan di Excel (seperti NORM.S.INV) umumnya adalah fungsi

yang proses inversnya tidak dapat ditulis sebagai rumus analitik sederhana. Untuk banyak distribusi, proses inversi sebenarnya dapat dilakukan secara analitis (yang mungkin menjadi salah satu alasan mengapa Excel tidak menyediakannya sebagai fungsi terpisah): untuk melakukannya, diperlukan penyetaraan nilai P dengan ekspresi matematika yang mendefinisikan probabilitas kumulatif dari setiap titik x, dan menyelesaikan persamaan ini untuk x dalam bentuk P. Misalnya, distribusi Weibull paling sering digunakan untuk menggambarkan probabilitas untuk waktu terjadinya (pertama) suatu proses dalam waktu kontinu, di mana intensitas kejadiannya mungkin tidak konstan. Distribusi ini memiliki dua parameter  $\alpha$  dan  $\beta$ , dengan  $\beta$  bertindak sebagai parameter skala. Fungsi kerapatannya adalah (untuk  $x > 0$ ):

$$f(x) = \frac{\alpha x^{\alpha-1}}{\beta^\alpha} e^{-\left(\frac{x}{\beta}\right)^\alpha}$$

dan fungsi kumulatifnya adalah:

$$F(x) = 1 - e^{-\frac{x^\alpha}{\beta^\alpha}}$$

Dengan mengganti sisi kiri persamaan terakhir dengan nilai probabilitas tertentu, P, dan menyelesaikannya untuk x, kita memperoleh:

$$x = \beta \cdot \left( \text{LN} \left( \frac{1}{1-P} \right) \right)^{1/\alpha}$$

sehingga sisi kanan adalah fungsi invers (P-ke-X, atau persentil).

	A	B	C	D
1				
2		<b>Item</b>	<b>Result</b>	<b>Formulae</b>
3		Random Percentage	19.9%	=RAND()
4				
5		Alpha		2
6		Beta		2
7		Random Sample	0.94	=C6*(LN(1/(1-C3)))^(1/C5))

**Gambar 5.25 Pembuatan Sampel Acak dari Distribusi Weibull**



	A	B	C	D
1				
2				
3			CRITBINOM (legacy)	BINOM.INV
4		Trials (number of trials)	10	10
5		Probability_s (probability of success)	30%	30%
6		Alpha	65.0%	65.0%
7		Result	3	3
8		Formula in row 7	=CRITBINOM(C4,C5,C6)	=BINOM.INV(D4,D5,D6)

**Gambar 5.26 Fungsi Invers Untuk Proses Binomial**

Gambar 5.25 menunjukkan contoh pembuatan sampel acak dari distribusi Weibull. Di dalam file, seseorang juga dapat melihat representasi grafis dari distribusi Weibull kumulatif, yang dibentuk dengan mengevaluasi rumus untuk masing-masing dari beberapa persentase tetap.

**Contoh: Nilai yang Terkait dengan Probabilitas untuk Proses Binomial**

Proses inversi yang sama berlaku untuk distribusi diskrit (seperti binomial) seperti halnya untuk distribusi kontinu, meskipun dalam beberapa kasus proses pencarian iteratif harus digunakan untuk menemukan nilai yang sesuai yang terkait dengan probabilitas kumulatif. Gambar 5.26 menunjukkan contoh fungsi BINOM. INV, serta fungsi CRITBINOM lama untuk menentukan jumlah hasil yang terkait dengan probabilitas kumulatif. Gambar 5.26 menunjukkan cuplikan layar, yang darinya dapat dilihat bahwa, dengan probabilitas (kira-kira) 65%, jumlah keberhasilan akan menjadi tiga atau kurang.

**Contoh: Interval Kepercayaan untuk Rata-rata Menggunakan Distribusi Student (T) dan Normal**

Pada bagian sebelumnya, kami mencatat bahwa statistik yang dihitung dari kumpulan data (seperti rata-rata dan simpangan baku) memberikan estimasi angka sebenarnya (tetapi tidak diketahui) dari distribusi yang menghasilkan data (dengan kata lain, distribusi seluruh populasi). Namun, distribusi dengan rata-rata yang berbeda juga dapat menghasilkan kumpulan data yang sama, meskipun hal ini menjadi kurang mungkin jika rata-rata distribusi kedua ini berbeda secara signifikan dari rata-rata kumpulan data.

Kami juga mencatat bahwa rata-rata (atau mean) dari sampel memberikan estimasi yang tidak bias dari mean sebenarnya (yaitu estimasi yang tidak bias, meskipun tetap tunduk pada ketidakpastian), sedangkan simpangan baku dari sampel perlu memiliki faktor koreksi yang diterapkan untuk memberikan estimasi yang tidak bias (tetapi juga tidak pasti) dari simpangan baku populasi sebenarnya. Faktanya, rentang nilai yang mungkin (atau interval kepercayaan) dari mean populasi diberikan oleh:

$$\mu_s \pm t \frac{\sigma_s}{\sqrt{n}}$$

di mana  $t$  adalah nilai distribusi T yang sesuai dengan persentase keyakinan yang diinginkan (dengan kata lain, setelah memilih persentase, seseorang perlu menemukan nilai distribusi

invers yang sesuai) dan  $\sigma$  adalah simpangan baku yang tidak bias (dikoreksi) sebagaimana diukur dari sampel. Dengan kata lain, meskipun seseorang dapat (100%) yakin bahwa rata-rata sebenarnya adalah antara dan , informasi tersebut tidak terlalu berguna untuk tujuan praktis. Sebaliknya, untuk dapat menentukan rentang yang layak secara bermakna, seseorang perlu menentukan tingkat keyakinan: untuk keyakinan 95%, seseorang akan menentukan setengah lebar dari distribusi T yang sesuai sehingga 95% dari area tersebut berada di kedua sisi rata-rata sampel (atau 2,5% di setiap ekor).

Distribusi T standar adalah distribusi simetris yang berpusat pada nol, yang memiliki satu parameter,  $v$ , yang dikenal sebagai jumlah derajat kebebasan. Simpangan bakunya adalah  $\sqrt{\frac{v}{v-2}}$ , dan karenanya lebih besar dari 1, tetapi sangat dekat dengan 1 ketika  $v$  besar (dalam hal ini distribusi-T sangat mirip dengan distribusi normal standar; untuk ukuran sampel yang besar, distribusi Normal sering digunakan sebagai pengganti distribusi-T). Perlu dicatat bahwa, ketika menggunakan fungsi Excel untuk distribusi-T, seseorang harus berhati-hati mengenai versi mana yang digunakan: sedangkan T.INV mengembalikan nilai-t yang terkait dengan probabilitas kumulatif (yaitu hanya pada ekor kiri distribusi), fungsi lama TINV mengembalikan nilai yang terkait dengan probabilitas yang terakumulasi pada ekor kiri dan kanan, serta memiliki pembalikan tanda.

Jadi, TINV(10%) akan memberikan nilai-t untuk kasus di mana 5% hasil dikecualikan pada setiap sisi ekor, sedangkan ini akan sesuai dengan T.INV(5%), selain perubahan tanda. Faktanya, fungsi TINV lama sesuai dengan fungsi T.INV.2T yang lebih baru. Dengan demikian, tingkat keyakinan (misalnya) 95% berarti bahwa 5% dikecualikan secara total, atau 2,5% dari setiap ekor. Oleh karena itu, seseorang dapat menggunakan T.INV setelah membagi probabilitas pengecualian dengan 2, atau menggunakan T.INV.2T yang diterapkan secara langsung pada probabilitas pengecualian, serta memastikan bahwa tanda-tanda ditangani dengan tepat. Gambar 5.28 berisi contoh fungsi-fungsi ini dalam menentukan interval keyakinan untuk rata-rata sampel data. Perhatikan bahwa sampel data aktual tidak diperlukan: satu-satunya informasi yang diperlukan adalah statistik ringkasan mengenai rata-rata dan deviasi standar sampel, jumlah total titik data (ukuran sampel) dan tingkat keyakinan yang diinginkan; dari sini, persentil yang dibutuhkan dihitung menggunakan fungsi invers, dan angka-angka (non-dimensi) ini kemudian diskalakan dengan tepat (lihat Gambar 5.28).

#### **Contoh: Fungsi CONFIDENCE.T dan CONFIDENCE.NORM**

Mengenai interval kepercayaan untuk mean, Excel memiliki dua fungsi yang memungkinkan perhitungan lebih langsung daripada yang ditunjukkan pada contoh sebelumnya; yaitu CONFIDENCE.T dan CONFIDENCE.NORM, tergantung pada apakah seseorang ingin mendasarkan analisis pada distribusi T atau pada distribusi normal (yang sangat mirip kecuali untuk ukuran sampel yang sangat kecil, seperti yang disebutkan di atas).

	A	B	C	D	E
1					
2		Degrees of freedom			
3		999			
4					
5		Cumulated Probability	T.INV(p)	TINV(p)	T.INT.2T
6		5%	-1.646	1.962	1.962
7		10%	-1.282	1.646	1.646
8		15%	-1.037	1.441	1.441
9		20%	-0.842	1.282	1.282
10		25%	-0.675	1.151	1.151
11		30%	-0.525	1.037	1.037
12		35%	-0.385	0.935	0.935
13		40%	-0.253	0.842	0.842
14		45%	-0.126	0.756	0.756
15		50%	0.000	0.675	0.675
16		55%	0.126	0.598	0.598
17		60%	0.253	0.525	0.525
18		65%	0.385	0.454	0.454
19		70%	0.525	0.385	0.385
20		75%	0.675	0.319	0.319
21		80%	0.842	0.253	0.253
22		85%	1.037	0.189	0.189
23		90%	1.282	0.126	0.126
24		95%	1.646	0.063	0.063

Gambar 5.27 Berbagai Bentuk Fungsi Distribusi Siswa Invers di Excel

	A	B	C	D
1				
2		Sample Data		
3		Mean of Sample	30.0	
4		StdDev of Sample	5.0	
5		Number of Data Points	1000	
6		Standard error of mean	0.16	=C4/SQRT(C5)
7		Degrees of Freedom	999	=C5-1
8				
9		Mean		
10				
11		Confidence Level	95.0%	
12		Exclusion probability	5.0%	=1-C11
13		t-value using "T.INV.2T"	1.96	=T.INV.2T(C12,C7)
14		Probability outside either tail	2.5%	=(1-C11)/2
15		t-value using "-T.INV"	1.96	=-T.INV(C\$14,C\$7)
16				
17		Lower Band Around Sample Statistic	-0.31	=-C\$13*C\$6
18		Upper Band Around Sample Statistic	0.31	=+C\$13*C\$6
19		Lower estimate	29.69	=C\$3+C17
20		Upper estimate	30.31	=C\$3+C18

Gambar 5.28 Interval Keyakinan untuk Rata-rata Berdasarkan Data Sampel

	A	B	C	D
1				
2		<b>Sample Data</b>		
3		Mean of Sample	30.0	
4		StdDev of Sample	5.0	
5		Number of Data Points	1000	
6		Standard error of mean	0.16	=C\$4/SQRT(C\$5)
7		Degrees of Freedom	999	=C5-1
8				
9		<b>Mean</b>		
10				
11		Confidence Level	95.0%	
12				
13		Width of interval using CONFIDENCE.T	0.3103	=CONFIDENCE.T(1-C11,C4,C5)
14		Lower estimate	29.69	=C\$3-C13
15		Upper estimate	30.31	=C\$3+C13
16				
17				
18		Width of interval using CONFIDENCE.NORM	0.3099	=CONFIDENCE.NORM(1-C11,C4,C5)

**Gambar 5.29 Interval Keyakinan untuk Rata-rata Menggunakan Fungsi KEYAKINAN**

#### **Contoh: Interval Keyakinan untuk Deviasi Standar Menggunakan Chi-kuadrat**

Interval keyakinan untuk deviasi standar dapat dihitung dengan menginversikan distribusi Chi-kuadrat (bukan distribusi T, seperti halnya untuk rata-rata). Dalam kasus ini, distribusi Chi-kuadrat memiliki parameter dan derajat kebebasan ( $\nu$ ) yang sama dengan distribusi T pada contoh sebelumnya. Namun, ini adalah distribusi yang condong positif yang menjadi lebih simetris (yaitu kurang condong) saat  $\nu$  meningkat; dengan demikian, interval keyakinan untuk deviasi standar juga condong positif, tidak seperti untuk rata-rata (yang simetris). Lihat Gambar 5.30 merupakan contoh yang menggunakan distribusi terpisah untuk probabilitas ekor kiri dan kanan, yaitu CHISQ.INV dan CHISQ.INV.RT, meskipun fungsi CHIINV ekor kiri yang lama juga dapat digunakan.

#### **Contoh: Interval Keyakinan untuk Kemiringan Garis Regresi (atau Beta)**

Sebelumnya dalam bab ini, kita mencatat bahwa kemiringan garis regresi linier terkait dengan deviasi standar dan korelasi antara kedua variabel:

$$Slope = \frac{\rho_{xy}\sigma_y}{\sigma_x}$$

Dalam konteks pasar keuangan, jika nilai-x adalah pengembalian periodik pada indeks pasar yang terdiversifikasi dengan baik, dan nilai-y adalah pengembalian periodik untuk beberapa aset (seperti perubahan harga saham), maka kemiringan garis regresi dikenal sebagai beta ( $\beta$ ) (empiris) dari aset tersebut:

$$\beta_s = \frac{\rho_{sm}\sigma_s}{\sigma_m}$$

(di mana  $\rho_{sm}$  merupakan koefisien korelasi antara pengembalian pasar dan aset, sedangkan  $\sigma_s$  dan  $\sigma_m$  masing-masing merupakan deviasi standar pengembalian aset dan pasar).

	A	B	C	D
1				
2		<b>Sample Data</b>		
3		Mean of Sample	30.0	
4		StdDev of Sample	5.0	
5		Number of Data Points	1000	
6		Standard error of mean	0.16	=C\$4/SQRT(C\$5)
7		Degrees of Freedom	999	=C5-1
8				
9		<b>StdDev</b>		
10				
11		Confidence Interval	95.0%	
12		Exclusion probability	5.0%	=1-C11
13		Single-sided probability	2.5%	=(1-C11)/2
14				
15		Lower Band Scaling	0.96	=SQRT(C\$7/CHISQ.INV.RT(C\$13,C\$7))
16		Upper Band Scaling	1.05	=SQRT(C\$7/CHISQ.INV(C\$13,C\$7))
17		Lower estimate	4.79	=C\$4*C15
18		Upper estimate	5.23	=C\$4*C16

**Gambar 21.30 Interval Kepercayaan untuk Deviasi Standar Berdasarkan Data Sampel**

Dengan demikian, beta (sebagai kemiringan garis) mewakili sensitivitas (rata-rata) pengembalian aset terhadap pergerakan indeks pasar. Pentingnya (untuk aset apa pun) kemampuan memperkirakan beta-nya adalah karena penggunaannya dalam Model Penetapan Harga Aset Modal (CAPM). Model teoritis tentang penetapan harga aset ini didasarkan pada gagasan bahwa harga aset harus disesuaikan ke tingkat di mana pengembalian yang diharapkan proporsional dengan bagian risiko yang tidak dapat didiversifikasi, yaitu yang berkorelasi dengan pasar keseluruhan semua aset.

Dengan kata lain, pengembalian yang diharapkan atas suatu aset (dan karenanya ekspektasi untuk harga saat ini) terkait secara linier dengan beta-nya. Tentu saja, perhitungan kemiringan garis regresi dari data yang diamati hanya akan memberikan perkiraan beta (yang sebenarnya). Interval kepercayaan untuk kemiringan garis regresi (yaitu beta) dapat dihitung menggunakan rumus dari statistik klasik (seperti contoh sebelumnya menghitung interval kepercayaan untuk rata-rata populasi dan deviasi standar).

Gambar 5.31 menunjukkan contoh rumus yang diperlukan (lihat Gambar 5.31). Perhitungannya melibatkan (di antara langkah-langkah lainnya) menghitung residual (yaitu kesalahan) antara data aktual dan garis regresi pada nilai-x tersebut, yang pertama-tama mengharuskan kemiringan dan intersep garis regresi diketahui. Perlu dicatat bahwa (untuk data bulanan selama 5 tahun, yaitu 60 titik data), interval kepercayaan untuk kemiringan agak

besar. Sementara ukuran sampel yang lebih besar untuk satu aset secara umum akan meningkatkan keakuratan perhitungan, semakin lama periode waktu pengumpulan data, semakin besar kemungkinan aspek fundamental bisnis atau lingkungan ekonomi makro telah berubah (sehingga sensitivitas bisnis yang sebenarnya terhadap pasar mungkin telah berubah).

	A	B	C	D	E	F	G	H	I	J	K
1											
2		SLOPE	0.950	=SLOPE(D13:D72,C13:C72)							
3		INTERCEPT	0.31%	=INTERCEPT(D13:D72,C13:C72)							
4											
5		Degrees of Freedom	58	=COUNT(D13:D72)-2	StdDev of Residuals	5.23%	=STDEV.S(D13:D72)				
6		STDEV (S&P)	3.6%	=STDEV.S(C13:C72)	Std Dev of Beta	1.461	=G5/G5				
7		STDEV (Kernametal)	6.2%	=STDEV(D13:D72)	Standard Error of Beta	0.192	=G6/SQRT(C5)				
8		CORREL	54.5%	=CORREL(C13:C72,D13:D72)							
9		RSQ	0.297	=RSQ(C13:C72,D13:D72)							
10											
11		Month	Log Changes		Kernametal from S&P	Errors/		Confidence Intervals for Beta	T-Distribution		
12			S&P	Kernametal	Forecasted value	Residuals		Confidence level	95%		
13		1	-1.4%	-0.9%	-0.96%	0.09%		Standard Deviations Required	2.002	=T.INV.2T(1-J12,C5)	
14		2	-2.2%	-1.1%	-1.73%	0.63%		Band Width	0.384	=J13*G7	
15		3	1.4%	4.9%	1.70%	3.19%		Base Figure	0.950	=C52	
16		4	1.2%	-3.7%	1.57%	-5.20%		Lower Figure	0.566	=C52-J14	
17		5	1.6%	-1.0%	1.93%	-2.95%		Upper Figure	1.334	=C52+J14	
18		6	3.1%	0.6%	3.32%	5.23%					

**Gambar 5.31 Interval Kepercayaan Untuk Kemiringan (Beta) Garis Regresi Dengan Perhitungan Eksplisit**

Dengan demikian, kumpulan data mungkin tidak konsisten secara internal, dan karenanya kurang valid. Hal ini menghadirkan satu tantangan untuk penilaian nilai beta aset yang benar. Oleh karena itu, dalam praktiknya, untuk memperkirakan beta suatu aset (misalnya saham perusahaan tertentu), regresi jarang diterapkan pada data aset tersebut saja; ukuran sampel yang lebih besar dibuat dengan menggunakan data beberapa perusahaan dalam suatu industri, untuk memperkirakan beta industri.

## 5.6 APLIKASI PRAKTIS: INFORMASI TENTANG ANALISIS REGRESI DAN PERAMALAN

Selain fungsi dasar yang berkaitan dengan regresi dan peramalan yang dibahas sebelumnya (seperti diagram sebaran X–Y, regresi linier, peramalan, dan interval keyakinan), Excel memiliki beberapa fungsi lain yang menyediakan data tentang nilai parameter regresi linier, interval keyakinan untuk nilai-nilai ini, kinerja regresi berganda sederhana, dan peramalan nilai masa depan berdasarkan tren historis, termasuk:

- LINEST, yang menghitung parameter tren linier dan estimasi kesalahannya.
- STEYX, yang menghitung kesalahan standar nilai y yang diprediksi untuk setiap x dalam regresi linier.
- LOGEST, yang menghitung parameter tren eksponensial.
- TREND dan GROWTH, yang menghitung nilai sepanjang tren linier dan eksponensial. Demikian pula, FORECAST.LINEAR (FORECAST dalam versi sebelum Excel 2016) menghitung nilai masa depan berdasarkan nilai yang ada.

Fungsi tipe FORECAST.ETS. Ini termasuk:

- FORECAST.ETS, yang menghitung nilai masa depan berdasarkan nilai (historis) yang ada

- dengan menggunakan algoritma exponential triple smoothing (ETS).
- FORECAST.ETS.CONFINT, yang menghitung interval kepercayaan untuk nilai perkiraan pada tanggal target yang ditentukan.
- FORECAST.ETS.SEASONALITY, di mana Excel mendeteksi dan menghitung panjang pola berulang dalam deret waktu.
- FORECAST.ETS.STAT, yang menghitung nilai statistik yang terkait dengan metode perkiraan deret waktu bawaan.

### Contoh: Menggunakan LINEST untuk Menghitung Interval Keyakinan untuk Kemiringan (atau Beta)

LINEST adalah fungsi array yang menghitung tidak hanya kemiringan dan intersep garis regresi, tetapi juga beberapa item yang berkaitan dengan kesalahan statistik dan interval keyakinan. Faktanya, informasi yang diberikan oleh LINEST dapat digunakan untuk menentukan interval keyakinan untuk kemiringan (beta) garis regresi tanpa harus melakukan peramalan residu secara eksplisit (seperti yang dilakukan di bagian sebelumnya).

Gambar 5.32 menunjukkan sebuah contoh (lihat Gambar 5.32). Fungsi tersebut telah dimasukkan dalam rentang H3:I7, dengan rentang F3:G7 menunjukkan label (yang dimasukkan secara manual) yang menjelaskan arti masing-masing. Data yang diperlukan untuk menjalankan perhitungan interval kepercayaan ditunjukkan dalam sel yang disorot (H3, H4, I6) dan perhitungan interval kepercayaan yang sebenarnya ditunjukkan dalam rentang G13:G15 (seperti sebelumnya, ini memerlukan asumsi tambahan tentang tingkat kepercayaan yang diinginkan, dan inversi distribusi T).

	A	B	C	D	E	F	G	H	I
1									
2		Month	Log Changes			LINEST Array function			
3			S&P	Kennametal		SLOPE (BETA)	INTERCEPT	0.950	0.37%
4		1	-1.4%	-0.9%		Std Error of SLOPE	Std Error of INTERCEPT	0.192	0.68%
5		2	-2.2%	-1.1%		R <sup>2</sup>	Std Error of y estimate	0.297	5.27%
6		3	1.4%	4.9%		F-stat	Degrees of Freedom	24.539	58
7		4	1.3%	-3.7%		Regression sum of squares	Residual sum of squares	0.068	0.161
8		5	1.6%	-1.0%					
9		6	3.1%	0.6%		Confidence Intervals for Slope or Beta		T-distribution	
10		7	2.4%	7.2%		Confidence level		95%	
11		8	2.1%	-1.0%		Standard Deviations Required		2.002	=T.INV.2T(1-G10,6)
12		9	0.5%	-15.6%		Band Width		0.384	=G11*\$H4
13		10	0.0%	4.0%		Base Figure		0.950	=SH3
14		11	-3.1%	-3.4%		Lower Figure		0.566	=SH3-G12
15		12	1.2%	1.2%		Upper Figure		1.334	=SH3+G12
16		13	1.1%	2.2%					

Gambar 5.32 Interval Keyakinan untuk Kemiringan (Beta) Garis Regresi Menggunakan LINEST

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1														
2		DATA						LINEST						
3		Floor space	Offices	Entrances	Age	Value (DEPENDENT)		-702	9909	29328	48	142035	=LINEST(F4:F14,B4:E14,TRUE,TRUE)	
4		2610	2	2	20	333,700		85.0	2516.4	1946.9	23.1	59052.5		
5		2636	2	2	12	338,400		0.986	4720.8	#N/A	#N/A	#N/A		
6		2662	3	1.5	31	354,850		106.2	6.0	#N/A	#N/A	#N/A		
7		2688	3	2	27	352,500		9464641005	133715017	#N/A	#N/A	#N/A		
8		2714	2	3	53	326,650								
9		2740	4	2	23	397,150		PREDICTION For New Items						
10		2766	2	1.5	75	296,100		Floor space	Offices	Entrances	Age	Value		
11		2792	2	2	34	335,815		2479	2	2	45	308734	=H11*K3+I11*J3+J11*I3+K11*M3+L3	
12		2818	3	3	23	383,050								
13		2844	4	4	55	397,150								
14		2870	2	3	22	350,150								
15														

**Gambar 5.33 Menggunakan LINEST untuk Melakukan Regresi Ganda dan Prakiraan Terkait**

Fungsi STEYX di Excel mengembalikan kesalahan standar estimasi Y, yang setara dengan nilai yang ditunjukkan oleh fungsi LINEST di sel I5. Perlu dicatat juga bahwa, jika seseorang hanya tertarik pada parameter kemiringan dan intersep (dan tidak pada ukuran statistik lainnya), parameter terakhir (opsional) dari fungsi tersebut dapat dibiarkan kosong; dalam kasus ini, fungsi tersebut dimasukkan sebagai fungsi array dalam satu baris (bukan dalam lima baris).

#### **Contoh: Menggunakan LINEST untuk Melakukan Regresi Ganda**

Fungsi LINEST juga dapat digunakan untuk melakukan regresi ganda (yaitu ketika ada beberapa variabel independen dan satu variabel dependen). Setiap variabel independen akan memiliki kemiringan terkait, yang menggambarkan sensitivitas variabel dependen terhadap perubahan pada input independen tersebut. Gambar 5.33 menunjukkan contoh penggunaan LINEST untuk memperoleh koefisien kemiringan, intersep, dan informasi statistik relevan lainnya yang berkaitan dengan nilai pasar berbagai kantor berdasarkan karakteristiknya (lihat Gambar 5.33). Beberapa hal yang perlu diperhatikan:

- Rentang kolom tempat fungsi dimasukkan diperluas untuk mencerminkan jumlah variabel (yaitu, ada lima baris dan jumlah kolom sama dengan jumlah total variabel): satu kolom diperlukan untuk setiap variabel independen dan satu untuk intersep.
- Nilai dalam setiap sel mengikuti kunci yang sama seperti yang ditunjukkan pada contoh sebelumnya, hanya saja baris pertama menunjukkan kemiringan untuk setiap variabel independen (dengan kolom terakhir menunjukkan intersep), dan baris kedua menunjukkan galat standarnya.
- Kemiringan dan galat standar ditampilkan dalam urutan terbalik dengan urutan data pada variabel independen yang dimasukkan; Jadi, saat menggunakan data ini untuk membuat prediksi, seseorang perlu berhati-hati dalam menghubungkan kalkulasi prediktif dengan koefisien kemiringan (seperti yang ditunjukkan dalam rumus untuk sel L11).

Sekali lagi, jika seseorang hanya tertarik pada koefisien kemiringan regresi dan intersep, maka parameter terakhir (opsional) dari fungsi LINEST dapat dibiarkan kosong, dan fungsi dimasukkan dalam satu baris (bukan dalam lima baris).

#### **Contoh: Menggunakan LOGEST untuk Menemukan Kesesuaian Eksponensial**

Dengan cara yang sama seperti LINEST, fungsi LOGEST dapat digunakan jika seseorang



ingin menggambarkan hubungan antara variabel dengan kurva seperti:

$$y = bm^x$$

Hal ini juga dapat ditulis sebagai:

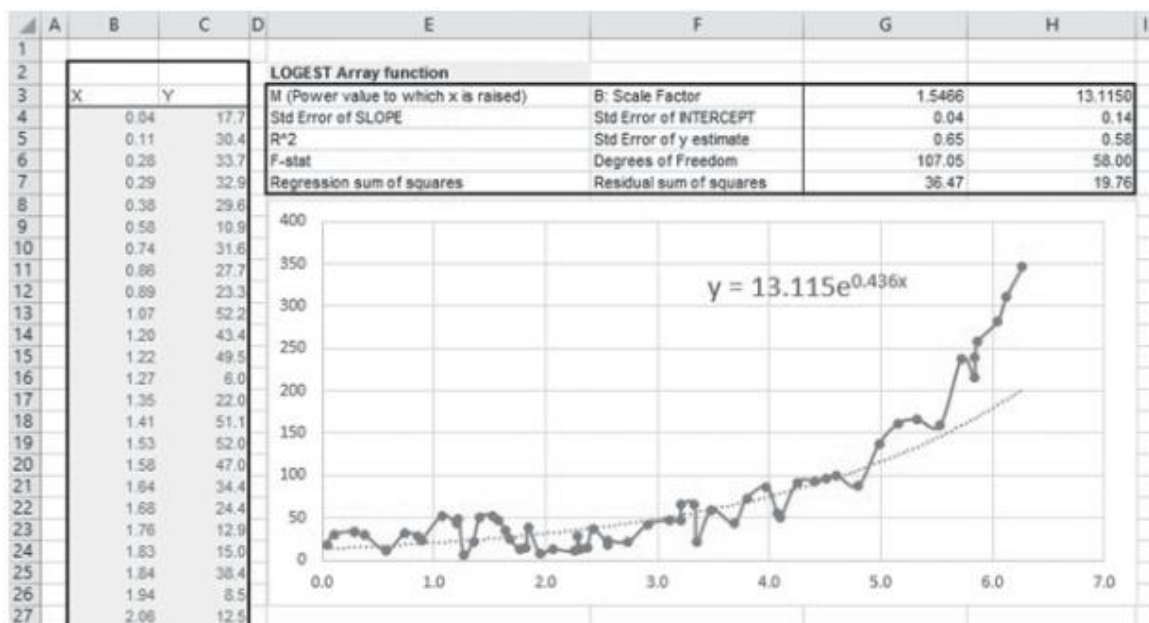
$$y = be^{x \log(m)}$$

menunjukkan bahwa hal ini setara dengan kurva eksponensial. Selain itu, hubungan tersebut dapat dituliskan sebagai berikut:

$$\log(y) = \log(b) + x \log(m)$$

Dalam formulasi terakhir ini, di mana data Y asli diubah dengan mengambil logaritmanya, fungsi LINEST dapat diterapkan untuk menghitung intersep dan kemiringan, yang masing-masing akan setara dengan  $\log(b)$  dan  $\log(m)$  (dari mana b dan m dapat dihitung). Fungsi LOGEST akan memberikan nilai-nilai (b dan m) ini secara langsung dari data (non-logaritma atau tidak diubah).

Gambar 5.34 menunjukkan contoh di mana (dalam lembar kerja bernama Data) fungsi LOGEST diterapkan pada data yang tampaknya memiliki hubungan eksponensial. Perhatikan bahwa garis tren eksponensial juga telah ditambahkan ke grafik (dengan mengklik kanan pada titik data untuk menghasilkan menu peka konteks). Ini juga menunjukkan nilai parameter yang disesuaikan (lihat Gambar 5.34); untuk persamaan garis tren perhatikan bahwa  $e^{0,436}$  1,5466 (mis. seperti yang ditunjukkan di sel G3).



Gambar 5.34 Penggunaan Fungsi LOGEST

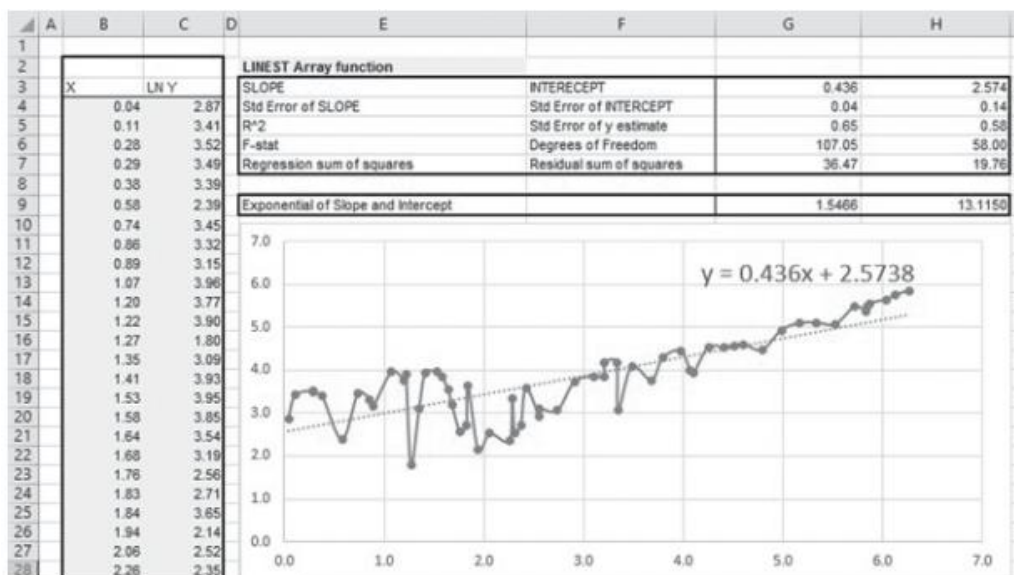
Dalam berkas yang sama, lembar kerja bernama LNofData menunjukkan hasil penerapan fungsi LINEST pada logaritma natural data. Lembar kerja tersebut juga menunjukkan bahwa rekonsiliasi antara koefisien yang dihitung untuk kemiringan dan intersep dari kecocokan linear ini sama dengan logaritma nilai yang dihitung dengan menggunakan LOGEST (lihat Gambar 5.35). Fungsi LOGEST juga dapat digunakan untuk beberapa variabel independen, dengan cara yang sama seperti yang ditunjukkan untuk LINEST guna melakukan regresi berganda, yaitu untuk konteks di mana seseorang ingin menggambarkan hubungan dengan:

$$y = bm_1^{x_1} m_2^{x_2} \dots m_n^{x_n}$$

### Contoh: Menggunakan TREND dan GROWTH untuk Meramalkan Tren Linear dan Eksponensial

Fungsi TREND (pertumbuhan linear) dan GROWTH (pertumbuhan eksponensial) dapat digunakan untuk menghitung (untuk nilai x tertentu) perkiraan nilai y dari garis tren. Tentu saja, orang akan berharap bahwa penggunaan fungsi-fungsi ini akan menghasilkan hasil yang sama seperti jika perkiraan eksplisit dibuat menggunakan data tentang kemiringan dan intersep garis tren:

- Untuk fungsi TREND, perhitungan eksplisit yang setara akan memerlukan penggunaan SLOPE dan INTERCEPT, atau nilai setara yang diambil dari hasil LINEST.
- Untuk fungsi GROWTH, prakiraan eksplisit dapat dilakukan dengan terlebih dahulu menghitung logaritma data Y, kemudian menerapkan SLOPE dan INTERCEPT pada data tersebut (untuk meramalkan nilai y logaritma) dan mengambil eksponen dari hasilnya. Atau, parameter skala yang diperlukan untuk prakiraan eksplisit (b dan m) dapat dihitung dari fungsi LOGEST.



Gambar 5.35 Perbandingan LOGEST dan LINEST yang Diterapkan pada Logaritma Data

Gambar 5.36 menunjukkan contoh setiap pendekatan (lihat Gambar 5.36). Sebagai penjelasan, himpunan data X bersifat umum dalam semua kasus, himpunan data Y1 diasumsikan memiliki hubungan linier pada dasarnya, dan himpunan data Y2 pada dasarnya bersifat eksponensial. Kedua pendekatan perhitungan untuk prakiraan ditunjukkan, yaitu menggunakan TREND yang diterapkan pada logaritma data atau GROWTH yang diterapkan pada data asli. Perhatikan bahwa dalam pendekatan di mana nilai Y2 diramalkan menggunakan keluaran dari LOGEST, fungsi tersebut telah digunakan dalam bentuk baris tunggal, di mana hanya koefisien regresi (dan bukan statistik lengkap) yang diberikan.

	A	B	C	D	E	F	G	H	I	J	K
1											
2		X	Y1: Linear	Y2: Exponential	LN Y2				Y1: TREND		
3		0.04	2.98	110.0	4.70			X-Value	1.00		
4		0.11	3.51	174.5	5.16			TREND Function	3.878	=TREND(D3:D62,B3:B62,J3)	
5		0.28	3.68	216.4	5.38			Calc using SLOPE and INTERCEPT	3.878	=INTERCEPT(D3:D62,B3:B62)+SLOPE(D3:D62,B3:B62)*J3	
6		0.29	3.54	202.9	5.31						
7		0.30	3.55	198.6	5.24			LN Y2: TREND			
8		0.50	2.40	67.2	4.21			X-Value	1.00		
9		0.74	3.62	176.8	5.18			TREND Function	4.806	=TREND(I3:I62,B3:B62,J6)	
10		0.86	3.40	177.4	5.18			Calc using SLOPE and INTERCEPT	4.806	=INTERCEPT(I3:I62,B3:B62)+SLOPE(I3:I62,B3:B62)*J6	
11		0.88	3.24	139.6	4.94			Y2 from TREND	122.28	=EXP(J6)	
12		1.07	4.04	309.6	5.74			Y2 from SLOPE and INTERCEPT	122.28	=EXP(J10)	
13		1.20	3.80	283.4	5.67			Y2: GROWTH			
14		1.22	3.91	314.0	5.75			X-Value	1.00		
15		1.27	1.80	34.8	3.55			GROWTH Function	122.28	=GROWTH(F3:F62,B3:B62,J15)	
16		1.35	3.14	141.7	4.95			LOGEST	122.28	=LOGEST(F3:F62,B3:B62,J15)	
17		1.41	3.97	324.3	5.78						
18		1.53	3.97	320.7	5.77			LOGEST without Statistics			
19		1.50	3.92	285.7	5.67			M (Power value to which x is raised)	B Scale Factor		
20		1.64	3.88	212.5	5.36						
21		1.60	3.30	145.8	4.98				1.54	79.28	=LOGEST(F3:F62,B3:B62,TRUE)

**Gambar 5.36 Penggunaan Fungsi TREND Dan GROWTH Dan Perbandingan Dengan Perhitungan Eksplisit**

#### Contoh: Peramalan Linier Menggunakan FORECAST.LINEAR

Fungsi FORECAST.LINEAR (dan fungsi FORECAST lama) melakukan peramalan linier, dan mirip dengan TREND, kecuali bahwa urutan parameternya berbeda. Berkas gambar 5.37 menunjukkan contoh yang seharusnya dapat dijelaskan sendiri oleh pembaca yang telah mengikuti pembahasan pada contoh sebelumnya.

	A	B	C	D	E	F	G
1							
2		X	Y		Y using FORECAST.LINEAR		
3		-1.4%	-0.9%		X-Value	1.00	
4		-2.2%	-1.1%		Calc using SLOPE and INTERCEPT	0.954	=INTERCEPT(C3:C62,B3:B62)+SLOPE(C3:C62,B3:B62)*F3
5		1.4%	4.9%		FORECAST.LINEAR Function	0.954	=FORECAST.LINEAR(F3,\$C\$3:\$C\$62,\$B\$3:\$B\$62)
6		1.3%	-3.7%				
7		1.6%	-1.0%				

**Gambar 5.37 Penggunaan Fungsi FORECAST.LINEAR Dan Perbandingan Dengan Perhitungan Eksplisit**

#### Contoh: Peramalan Menggunakan Rangkaian Fungsi FORECAST.ETS

Rangkaian fungsi tipe FORECAST.ETS menyediakan kemampuan untuk melakukan peramalan deret waktu yang lebih canggih daripada metode linier yang dibahas sebelumnya. Fungsi-fungsi tersebut didasarkan pada metode "perataan rangkap tiga eksponensial", yang

memungkinkan seseorang untuk menangkap tiga karakteristik utama dari rangkaian data:

- Nilai dasar.
- Tren jangka panjang.
- Potensi musim.

Gambar 5.38 menunjukkan sebuah contoh (lihat Gambar 5.38). Fungsi intinya adalah FORECAST.ETS, yang menyediakan peramalan pusat pada tanggal tertentu di masa mendatang. Fungsi FORECAST.ETS.CONFINT menyediakan lebar pita keyakinan di sekitar peramalan tersebut (untuk tingkat keyakinan tertentu, secara default 95%).



**Gambar 5.38 Contoh Penggunaan Rangkaian FORECAST.ETS**

Fungsi FORECAST.ETS.SEASONALITY dapat digunakan untuk melaporkan panjang periode musiman yang telah dideteksi fungsi tersebut dalam data. Terakhir, fungsi FORECAST.ETS.STAT dapat digunakan untuk memberikan nilai dari delapan parameter statistik utama yang tersirat dalam metode pemulusan rangkap tiga eksponensial. Perlu dicatat bahwa tanggal harus diatur dengan langkah konstan antara titik waktu, misalnya bulanan atau tahunan.

Parameter opsional meliputi musiman (yaitu panjang periode musiman yang akan digunakan, atau 1 agar musiman dapat dideteksi secara otomatis), pelengkapan data (yaitu 0 untuk memperlakukan data yang hilang sebagai nol, 1 untuk melengkapi titik yang hilang berdasarkan rata-rata titik yang berdekatan) dan agregasi (yaitu bagaimana titik pada tanggal yang sama harus diperlakukan, defaultnya adalah rata-rata).

Dalam konteks ini, ada baiknya disebutkan fitur Excel lain yang berpotensi berguna, yaitu grafik prakiraan; setelah memilih kumpulan data garis waktu dan nilai, dan memilih “Lembar Prakiraan” pada tab Data Excel, Excel akan membuat lembar kerja baru dengan grafik prakiraan, termasuk interval keyakinan. Gambar 5.39 menunjukkan hasil dari tindakan ini.



Gambar 5.39 Hasil Penggunaan Fitur Lembar Prakiraan

## BAB 6

### FUNGSI INFORMASI

#### 6.1 PENDAHULUAN

Bab ini membahas fungsi Informasi Excel. Fungsi ini umumnya menyediakan informasi tentang konten, posisi, atau format sel, atau tentang lokasi dan lingkungan sel, rentang, lembar kerja, atau buku kerja:

- ISTEK mengembalikan TRUE jika argumennya berupa teks; ISNONTEXT mengembalikan TRUE jika argumennya bukan teks.
- ISNUMBER mengembalikan TRUE jika argumennya berupa angka.
- ISBLANK mengembalikan TRUE jika argumennya kosong.
- ISFORMULA mengembalikan TRUE jika merujuk ke sel yang berisi rumus.
- ISLOGICAL mengembalikan TRUE jika argumennya berupa nilai logika.
- ISREF mengembalikan TRUE jika argumennya berupa referensi.
- ISEVEN atau ISODD mengembalikan TRUE jika angka yang dirujuk masing-masing genap atau ganjil.
- ISERROR mengembalikan TRUE jika nilainya adalah nilai kesalahan apa pun. ISERR mengembalikan TRUE jika nilainya adalah nilai kesalahan apa pun kecuali #N/A.
- ERROR.TYPE mengembalikan angka yang sesuai dengan jenis kesalahan.
- NA mengembalikan nilai kesalahan #N/A.
- ISNA mengembalikan TRUE jika nilainya adalah nilai kesalahan #N/A.
- N mengembalikan nilai yang dikonversi menjadi angka.
- TYPE menunjukkan jenis data suatu nilai.
- CELL menyediakan informasi tentang pemformatan, lokasi, atau konten sel.
- INFO menyediakan informasi tentang lingkungan operasi saat ini.
- SHEET mengembalikan nomor lembar dari lembar yang dirujuk, dan SHEETS mengembalikan jumlah lembar dalam referensi.

Perlu diketahui bahwa ada fungsi Excel lain yang menyediakan bentuk informasi yang mirip dengan beberapa fungsi ini. Misalnya, fungsi ADDRESS, ROW, ROWS, COLUMN dan COLUMNS (dalam kategori Lookup & Reference Excel) memiliki beberapa karakteristik yang sama dengan fungsi CELL; hal ini dibahas dalam Bab 9.

#### 6.2 APLIKASI PRAKTIS

Bagian ini menyediakan beberapa contoh penggunaan fungsi Informasi terpilih dalam analisis dan manipulasi data, termasuk menilai integritas data, dan menyediakan informasi lainnya.

##### 6.2.1 Contoh: Komentar Rumus dengan ISTEK, ISNUMBER, atau N

Fungsi ISTEK memungkinkan seseorang untuk mencatat atau mendokumentasikan

model dengan menggunakan komentar dalam rumus. Ini merupakan alternatif untuk menggunakan komentar yang dilampirkan ke sel, dengan keuntungan bahwa komentar dapat dilihat saat Excel dialihkan ke tampilan rumusnya (Rumus/Tampilkan Rumus). Jadi, saat mengaudit model dengan memeriksa rumusnya, komentar tersedia secara langsung. File Ch22.1.InCellComments.xlsx menunjukkan contoh dari dua pendekatan (saat model ditampilkan dalam tampilan rumus) (lihat Gambar 6.1). Di Sel C6, fungsi ISTEEXT digunakan untuk menyediakan komentar; karena fungsi ini dievaluasi sebagai TRUE (diperlakukan sebagai 1 oleh Excel), hasil numerik dari perhitungan sama dengan nilai asli (12000).

	A	B	C	D
1				
2			Salaries 2017 \$	
3			In-Formula Comment	Attached Comment
4	Fred		27000	27000
5	Bill		35000	35000
6	Harry		=12000*ISTEEXT("Works half-time")	12000
7	Jo		43000	43000
8	Mary		57000	57000
9	Christine		45000	45000
10				

**Gambar 6.1 Penggunaan ISTEEXT untuk Membuat Komentar dalam Rumus**

Di Sel D6, pendekatan komentar terlampir ditampilkan, yaitu di mana komentar harus diperiksa secara terpisah untuk melihat isinya. Perhatikan bahwa fungsi ISNUMBER atau N juga dapat digunakan dalam konteks ini: misalnya, ISNUMBER("Komentar") atau N("Komentar. . .") akan ditambahkan ke rumus sel inti (daripada dikalikan), karena keberadaan komentar teks akan mengakibatkan setiap fungsi mengevaluasi ke nol.

#### **Contoh: Membangun Model Prakiraan yang Dapat Diperbarui dengan Angka Aktual yang Dilaporkan**

Fungsi ISBLANK dapat digunakan untuk menentukan apakah sel kosong (terkadang, ISBLANK yang disematkan dalam fungsi NOT diperlukan untuk menentukan bahwa sel tidak kosong). Salah satu aplikasinya adalah ketika model dibangun untuk diperbarui dengan aktual saat tersedia, tetapi untuk menggunakan angka prakiraan hingga aktual tersedia. Gambar 6.2 menunjukkan sebuah contoh (lihat Gambar 6.2):

- Bagian pertama dari file (sel B2:I4) menunjukkan model perkiraan tradisional di mana penjualan masa depan dihitung berdasarkan asumsi tentang nilai pada tahun awal (2016) dan tingkat pertumbuhan pada setiap tahun mendatang. Pendekatan pertama cocok untuk penggunaan satu kali (seperti untuk transaksi keuangan perusahaan). Namun, pengerjaan ulang yang signifikan akan diperlukan jika model tersebut harus diperbarui saat data untuk tahun mendatang tersedia (seperti data 2017 pada akhir tahun tersebut). Tentu saja, untuk model dua baris, pengerjaan ulang ini dapat dilakukan dengan sangat cepat, tetapi untuk model yang lebih besar yang berisi banyak item baris, prosesnya akan rumit dan rawan kesalahan.
- Bagian kedua dari file menunjukkan bagaimana model dapat diadaptasi, sehingga

asumsi yang digunakan untuk membuat perkiraan bergantung pada apakah data aktual yang dilaporkan ada, yang dideteksi menggunakan fungsi ISBLANK. Jika tidak ada data yang dilaporkan, asumsi prakiraan awal digunakan, jika tidak, asumsi diatur ulang sehingga prakiraan yang dihasilkan mereplikasi angka yang dilaporkan.

Perhatikan bahwa kombinasi  $=NOT(ISBLANK(G3))$  akan mengembalikan angka 1 dalam semua kasus di mana sel G3 memiliki konten apa pun, sedangkan (seperti yang disebutkan dalam Bab 1), penggunaan  $=IF(G3,1,0)$  akan mengembalikan angka 1 jika sel G3 berisi angka bukan nol, angka nol jika G3 adalah nol, dan #VALUE dalam kasus entri teks. Perlu dicatat juga bahwa dalam konteks seperti itu, perbedaan antara sel kosong dan sel yang berisi angka nol di bidang aktual sangat mendasar; sel kosong berarti belum ada angka aktual yang dilaporkan, sedangkan sel yang berisi angka nol berarti angka aktual telah dilaporkan dan nilainya adalah nol (yang tentu saja merupakan nilai yang valid dalam banyak konteks, seperti untuk jumlah kecelakaan serius yang mungkin terjadi dalam sebulan).

	A	B	C	D	E	F	G	H	I	J
1										
2		TRADITIONAL		2016	2017	2018	2019	2020	2021	
3		SALES	Historic/Forecast	100.0	105.0	110.3	115.8	121.6	127.6	$=H3*(1+I4)$
4			Assumptions		5.0%	5.0%	5.0%	5.0%	5.0%	
5										
6		UPDATING		2016	2017	2018	2019	2020	2021	
7		SALES	Historic/Actuals	100.0	104.0					
8			Assumptions if No Actuals		5.0%	5.0%	5.0%	5.0%	5.0%	
9			Assumptions To Use		4.0%	5.0%	5.0%	5.0%	5.0%	$=IF(ISBLANK(I7),I8,I7/H10-1)$
10		SALES To Use	Growth Rate	100.0	104.0	109.2	114.7	120.4	126.4	$=H10*(1+I9)$

**Gambar 6.2 Penggunaan ISBLANK untuk Membuat Prakiraan yang Diperbarui saat Data yang Dilaporkan Dimasukkan**

### Contoh: Mendeteksi Konsistensi Data dalam Basis Data

Fungsi ISBLANK dapat digunakan dalam beberapa kasus untuk mendeteksi ketidakkonsistenan dalam kumpulan data. lihat Gambar 6.3 menunjukkan contoh sekumpulan kontrak, yang masing-masing harus memiliki tanggal mulai dan berakhir (untuk kontrak yang ada) atau memiliki kedua kolom tanggal kosong (untuk kontrak yang sedang dinegosiasikan, misalnya).

Fungsi ISBLANK digunakan dalam pernyataan IF untuk mendeteksi nilai yang tidak konsisten, yang bernilai nol (Pemformatan Bersyarat juga digunakan untuk menyorot nilai-nilai ini). Tentu saja, perlu dicatat bahwa fungsi lain (seperti ISNUMBER atau ISTEXT) mungkin diperlukan dalam situasi yang serupa.

#### 6.2.2 Contoh: Penggunaan "N/A" yang Konsisten dalam Model

Eksresi teks "NA" sering digunakan dalam pernyataan IF, misalnya untuk menunjukkan bahwa ketidakkonsistenan telah muncul dalam data. Secara umum, penggunaan bidang teks tersebut tidak disukai (dalam serangkaian kalkulasi sementara), karena kalkulasi selanjutnya cenderung lebih rentan terhadap kesalahan.

Misalnya, kebutuhan untuk memulai banyak rumus dependen dengan item seperti  $IF(G56="NA")$  akan menambah kerumitan, mengurangi transparansi, dan cenderung



menyebabkan kesalahan (misalnya karena kesalahan ejaan atau interpretasi yang salah dari bidang teks "NA", seperti jika ditulis "#N/A"). Alternatifnya meliputi:

- Menggunakan indikator numerik (seperti nol) untuk menunjukkan ketidakkonsistenan. Ini umumnya merupakan pendekatan yang disukai penulis, karena mempertahankan integritas dasar (sebagian besar) model (sebagai serangkaian kalkulasi numerik), dan cenderung membuat rumus selanjutnya lebih sederhana.
- Menggunakan fungsi "=NA()" untuk secara konsisten menangani item tersebut di Excel.

	A	B	C	D	E
1					
2		<b>Start</b>	<b>End</b>	<b>Consistency Check</b>	<b>Formula</b>
3		1-Nov-16	31-Oct-19	1	=IF(ISBLANK(B3)=ISBLANK(C3),1,0)
4		1-Sep-15	31-Aug-19	1	=IF(ISBLANK(B4)=ISBLANK(C4),1,0)
5		1-Sep-15	31-Aug-19	1	=IF(ISBLANK(B5)=ISBLANK(C5),1,0)
6		1-Nov-16	31-Oct-19	1	=IF(ISBLANK(B6)=ISBLANK(C6),1,0)
7		1-Jan-17	31-Dec-20	1	=IF(ISBLANK(B7)=ISBLANK(C7),1,0)
8		1-Jul-15	30-Jun-19	1	=IF(ISBLANK(B8)=ISBLANK(C8),1,0)
9		1-Oct-15	30-Sep-19	1	=IF(ISBLANK(B9)=ISBLANK(C9),1,0)
10			31-Oct-19	0	=IF(ISBLANK(B10)=ISBLANK(C10),1,0)
11		1-Jun-18	31-May-19	1	=IF(ISBLANK(B11)=ISBLANK(C11),1,0)
12		1-Nov-18	31-Oct-19	1	=IF(ISBLANK(B12)=ISBLANK(C12),1,0)
13		1-Nov-18	31-Oct-19	1	=IF(ISBLANK(B13)=ISBLANK(C13),1,0)
14		1-Jul-15	30-Jun-19	1	=IF(ISBLANK(B14)=ISBLANK(C14),1,0)
15		1-Oct-19		0	=IF(ISBLANK(B15)=ISBLANK(C15),1,0)
16		1-May-18	31-May-19	1	=IF(ISBLANK(B16)=ISBLANK(C16),1,0)
17		1-Jun-16	31-May-20	1	=IF(ISBLANK(B17)=ISBLANK(C17),1,0)

Gambar 6.3 Penggunaan ISBLANK Untuk Mendeteksi Entri Data Yang Tidak Konsisten

	A	B	C	D	E	F	G	H
1								
2								
3		Count		14	14			
4		Sum		13926	#N/A			
5		Multiplication of element		#VALUE!	#N/A			
6								
7		<b>Start</b>	<b>End</b>	<b>Using "NA" Text Field</b>	<b>Using =NA() Function</b>			
8		1-Nov-18	31-Oct-19	1094	1094			
9		1-Sep-15	31-Aug-19	1460	1460			
10		1-Sep-15	31-Aug-19	1460	1460			
11		1-Nov-16	31-Oct-19	1094	1094			
12		1-Jan-17	31-Dec-20	1460	1460			
13		1-Jul-15	30-Jun-19	1460	1460			
14		1-Oct-15	30-Sep-19	1460	1460			
15			31-Oct-19	NA	#N/A			
16		1-Jun-18	31-May-19	364	364			

Gambar 6.4 Penggunaan Fungsi NA()

Gambar 6.4 menunjukkan contoh pendekatan terakhir ini, yang membandingkan penggunaan kolom teks "NA" dengan fungsi "=NA()". Fungsi terakhir umumnya lebih konsisten daripada yang pertama dalam hal memberikan nilai kesalahan #N/A yang konsisten ketika rumus yang merujuk ke data tidak dapat diterapkan. Gambar 6.4 menunjukkan cuplikan layar dari contoh tersebut.

### 6.2.3 Contoh: Aplikasi Fungsi INFO dan CELL: Tinjauan Umum

Fungsi INFO dan CELL dapat berguna untuk mendokumentasikan model dan mencari tahu informasi tentang lingkungan operasi saat ini. Fungsi ini dapat sangat penting dalam aplikasi yang lebih canggih, termasuk penulisan kode VBA yang kuat. Fungsi INFO umumnya menyediakan fungsionalitas yang berkaitan dengan buku kerja atau lingkungan operasi:

- Jalur direktori atau folder saat ini. Fungsi ini sering kali penting untuk diketahui saat menulis kode VBA yang perlu bekerja dengan semua file dalam direktori yang sama dengan model utama.
- Jumlah lembar kerja aktif dalam buku kerja terbuka.
- Referensi sel dari sel paling atas dan paling kiri yang terlihat (sebagai teks).
- Versi sistem operasi saat ini (sebagai teks).
- Mode perhitungan ulang saat ini: Otomatis atau Manual.
- Versi Microsoft Excel yang digunakan (sebagai teks). Ini dapat berguna saat menulis kode VBA untuk memastikan bahwa versi terbaru dari fungsionalitas Excel sedang digunakan, seperti dalam pembuatan fungsi "wrapper"
- Nama lingkungan operasi: "mac" atau "pcdos".

Fungsi CELL menyediakan informasi tentang posisi, konten, atau format sel atau rentang yang direferensikan, termasuk:

- Jalur lengkap file yang berisi rentang referensi (sebagai teks).
- Alamat (sebagai teks) sel pertama dalam rentang yang direferensikan, atau alamat sel yang terakhir diubah dalam buku kerja jika rentang yang direferensikan dihilangkan.
- Nomor kolom sel yang direferensikan.
- Nomor baris sel yang direferensikan.
- Apakah sel diformat dalam warna untuk nilai negatif atau tidak.
- Konten sel kiri atas dalam rentang yang direferensikan.
- Format angka sel.
- Apakah sel diformat dengan tanda kurung untuk nilai positif atau untuk semua nilai.
- Awalan label sel, yaitu mengembalikan tanda kutip tunggal (') jika sel berisi teks rata kiri, tanda kutip ganda (") jika sel berisi teks rata kanan, dst.
- Apakah sel terkunci (dilindungi) atau tidak.
- Jenis data dalam sel (misalnya mengembalikan "b" untuk kosong jika sel kosong, "l" untuk label jika sel berisi konstanta teks, dan "v" untuk nilai jika sel berisi hal lain).
- Lebar kolom sel (dibulatkan ke bilangan bulat), di mana setiap unit lebar kolom sama dengan lebar satu karakter dalam ukuran font default.

#### Contoh: Membuat Label Pembaruan yang Merujuk ke Data atau Rumus

Terkadang berguna untuk membuat label yang mereferensikan sel atau rentang. Misalnya, jika kumpulan data dimulai di Sel C5, seseorang dapat ingin membuat label seperti "Data dimulai di C5", dengan label yang merujuk ke sel (sehingga akan diperbarui jika baris atau kolom baru diperkenalkan). Label tersebut dapat dibuat dengan rumus seperti:

"Data dimulai di "& CELL("alamat",C5)

	A	B	C	D
1				
2		<b>INFO Function</b>		
3		<b>Type</b>	<b>Example</b>	<b>Formulae</b>
4		Directory	C:\Users\Michael\Documents\	=INFO(B4)
5		Numfile		=INFO(B5)
6		Origin	\$A:\$A\$1	=INFO(B6)
7		OSVersion	Windows (32-bit) NT 10.00	=INFO(B7)
8		Recalc	Automatic	=INFO(B8)
9		Release	16.0	=INFO(B9)
10		System	pcdos	=INFO(B10)
11				
12		<b>CELL Function</b>		
13		<b>Type</b>	<b>Example</b>	<b>Formula</b>
14		Filename	C:\Users\Michael\Desktop\taaaFMPIIDraft\FM	=CELL(B14)
15		Address	\$A\$1	=CELL(B15)
16		Address	\$B\$5	=CELL(B16,\$B\$5:\$C\$6)
17		Col		=CELL(B17,\$B\$5:\$C\$6)
18		Row		=CELL(B18,\$B\$5:\$C\$6)
19		Color		=CELL(B19,\$B\$5:\$C\$6)
20		Contents	Numfile	=CELL(B20,\$B\$5:\$C\$6)
21		Format	G	=CELL(B21,\$B\$5:\$C\$6)
22		Parentheses		=CELL(B22,\$B\$5:\$C\$6)
23		Prefix	,	=CELL(B23,\$B\$5:\$C\$6)
24		Protect		=CELL(B24,\$B\$5:\$C\$6)
25		Type	I	=CELL(B25,\$B\$5:\$C\$6)
26		Width		=CELL(B26,\$B\$5:\$C\$6)

**Gambar 6.5** Gambaran Umum Fungsi INFO dan CELL

**Contoh: Menunjukkan kepada Pengguna Mode Perhitungan Ulang yang Digunakan pada File**

Dalam beberapa kasus, penting untuk memberi tahu pengguna secara jelas dan eksplisit mode perhitungan ulang yang digunakan Excel. Misalnya, beberapa model mungkin perlu diatur ke Perhitungan manual karena ukurannya besar (atau lambat dihitung ulang karena alasan lain), sehingga perhitungan ulang dapat dilakukan hanya jika diperlukan, bukan secara otomatis setelah setiap perubahan dilakukan. Tentu saja, pengaturan buku kerja ke Perhitungan ulang manual berpotensi rawan kesalahan jika seseorang mengubah data input dan menggunakan hasilnya sambil mengabaikan kebutuhan untuk menghitung ulang buku kerja.

Oleh karena itu, seseorang mungkin ingin menunjukkan mode perhitungan ulang secara eksplisit serta menyorot saat diatur ke Manual dengan menggunakan Pemformatan Bersyarat Excel. Gambar 6.6 berisi contoh penggunaan bentuk "Recalc" dari fungsi INFO (dalam praktiknya, ini akan ditempatkan di suatu tempat yang jelas bagi pengguna atau dalam deskripsi model di dalam file Excel itu sendiri). Pemformatan Bersyarat digunakan untuk memberi bayangan pada sel (kuning dalam file elektronik) saat sel menunjukkan nilai "Manual".

	A	B	C	D	E	F
1						
2		Data For Vodafone starts in \$C\$5		=Data For Vodafone starts in "&CELL("address",C5)		
3						
4			Daily returns			
5		Day	VOD	ICI	Barclays	
6		1	1.7%	1.0%	-2.5%	
7		2	-2.1%	1.1%	1.1%	
8		3	0.1%	2.9%	-1.3%	
9		4	0.0%	-1.7%	0.2%	
10		5	-1.0%	-0.7%	-1.8%	
11		6	1.1%	0.3%	0.9%	
12		7	-1.3%	-3.6%	-1.9%	
13		8	0.0%	-1.1%	-1.3%	
14		9	1.4%	-1.7%	-0.4%	
15		10	-2.1%	-0.8%	0.3%	

**Gambar 6.6 Penggunaan Fungsi CELL untuk Membuat Label Pembaruan yang Mengacu pada Lokasi Data**

**Contoh: Menemukan Versi Excel yang Digunakan dan Membuat Rumus yang Kompatibel dengan Versi Sebelumnya**

Penggunaan INFO("RELEASE") akan memberikan informasi tentang versi Excel yang digunakan; Excel 2003 adalah rilis 11, Excel 2007 rilis 12, Excel 2010 rilis 14, Excel 2013 rilis 15, dan Excel 2016 adalah rilis 16. Salah satu contoh penggunaan pengetahuan ini adalah untuk membuat rumus yang menggunakan bentuk fungsi terbaru, sebagaimana tersedia dalam versi Excel milik pengguna. Misalnya, pada prinsipnya seseorang mungkin ingin model tersebut menggunakan fungsi NORM.S.INV (lihat Bab 5), tetapi fungsi NORMSINV akan digunakan untuk pengguna dengan versi 11 atau sebelumnya. Gambar 6.8 berisi sebuah contoh (lihat Gambar 6.8).

Perhatikan bahwa informasi rilis yang dihasilkan dari penggunaan INFO("RELEASE") adalah kolom teks, tetapi untuk sebagian besar tujuan dapat digunakan tanpa konsekuensi seolah-olah berupa angka. Namun, umumnya lebih kuat untuk menerapkan fungsi VALUE (lihat Bab 8) untuk mengubah kolom teks yang tampaknya numerik ini menjadi angka yang sesuai. Rumus di Sel C6 kemudian menggunakan satu atau beberapa fungsi inversi, tergantung pada versi rilis, untuk menghitung kebalikan (persentil) dari distribusi normal standar untuk nilai persentase yang dihasilkan oleh fungsi RAND(). Fungsionalitas tersebut dapat dibangun ke dalam fungsi "wrapper" yang ditentukan pengguna VBA, yang akan lebih mudah dipelihara (seiring dengan tersedianya versi Excel baru), terutama jika digunakan di beberapa tempat.

	A	B	C	D
1				
2		The recalculation of this workbook is:	Manual	=INFO("Recalc")
3				

**Gambar 22.7 Penggunaan CELL untuk Menampilkan Mode Perhitungan Ulang**

	A	B	C	D
1				
2		<b>Item</b>	<b>Calculation</b>	<b>Formula</b>
3		Release as Text	16.0	=INFO("RELEASE")
4		Release as Value	16	=VALUE(C3)
5		Rand	0.194	=RAND()
6		Percentile of N(0,1)	-0.865	=IF(C4>=12,NORM.S.INV(C5),NORMSINV(C5))
7				

**Gambar 6.8 Penggunaan Informasi Rilis untuk Membuat Fungsi yang Kompatibel dengan Versi Sebelumnya**

#### **Contoh: Lokasi File dan Informasi Struktural Menggunakan CELL, INFO, SHEET, dan SHEETS**

Seperti disebutkan di atas, fungsi CELL dan INFO dapat digunakan untuk memberikan nama file dan direktori file saat ini. Informasi berguna lainnya dapat berupa jumlah lembar kerja dalam buku kerja, serta (terutama saat menulis VBA) nomor lembar kerja. Gambar 6.9 menunjukkan contoh penggunaan fungsi CELL dan INFO, serta fungsi SHEETS dan SHEET (lihat Gambar 6.9).

	A	B	C	D
1				
2		Note:	Sheet 2 is placed after Sheet 1 but is hidden; Sheet 3 is placed after Sheet 4.	
3				
4		<b>Function Examples</b>	<b>Formulae</b>	<b>Description</b>
5		C:\Users\Michael\Desktop\aaa\FMPIDraft\FMPIL.Mc	=CELL("filename")	Filename
6		C:\Users\Michael\Documents\	=INFO("DIRECTORY")	Directory
7			=SHEETS()	The number of worksheets in this workbook (including hidden sheets)
8			=SHEETS(Sheet1:Sheet3!B7)	The number of worksheets in this workbook (including hidden sheets)
9			=SHEET(Sheet4!B5)	The sheet number of the referenced sheet
10				
11				

**Gambar 6.9 Contoh Terpadu Penggunaan Fungsi INFO, CELL, SHEETS, dan SHEET**

Perhatikan bahwa urutan lembar kerja adalah Sheet1, Sheet2, Sheet4, Sheet3, dengan Sheet2 tersembunyi. Fungsi SHEETS menunjukkan bahwa ada empat lembar (termasuk lembar tersembunyi) dan fungsi SHEET menunjukkan bahwa Sheet4 adalah lembar ketiga buku kerja (termasuk lembar tersembunyi). Fungsi SHEETS juga menghitung lembar bagan (bukan hanya lembar kerja); Dengan demikian, jika seseorang memasukkan sejumlah data ke dalam berkas contoh dan menekan pintasan F11 untuk menyisipkan lembar bagan, maka akan terlihat jumlah lembar bertambah.

## BAB 7

### FUNGSI TANGGAL DAN WAKTU

#### 7.1 PENDAHULUAN

Bab ini membahas fungsi-fungsi dalam kategori Tanggal dan Waktu Excel, termasuk:

- DATE mengembalikan nomor seri tanggal, yang sesuai dengan tahun, bulan, dan hari yang ditentukan.
- DATEDIF menghitung usia dalam satuan, yaitu jumlah hari, bulan, atau tahun antara dua tanggal.
- DATEVALUE mengonversi tanggal dalam bentuk teks menjadi nomor seri.
- DAY mengonversi angka menjadi hari dalam sebulan.
- DAYS mengembalikan jumlah hari antara dua tanggal.
- DAYS360 menghitung jumlah hari antara dua tanggal berdasarkan tahun 360 hari.
- EDATE mengembalikan jumlah tanggal yang merupakan jumlah bulan yang ditunjukkan sebelum atau setelah tanggal mulai.
- EOMONTH menghitung jumlah hari terakhir bulan sebelum atau setelah jumlah bulan yang ditentukan.
- YEAR mengonversi angka menjadi tahun.
- MONTH mengonversi angka menjadi bulan.
- WEEKDAY mengonversi angka menjadi hari dalam seminggu.
- HOUR mengonversi angka menjadi jam.
- MINUTE mengonversi angka menjadi menit.
- SECOND mengonversi angka menjadi detik.
- TODAY mengembalikan angka tanggal hari ini.
- NOW mengembalikan angka tanggal dan waktu saat ini.
- TIME mengembalikan angka waktu.
- ISOWEEKNUM mengembalikan angka minggu ISO dalam setahun untuk tanggal tertentu.
- WEEKNUM menunjukkan nomor minggu dari tanggal dalam tahun yang bersangkutan.
- NETWORKDAYS mengembalikan jumlah hari kerja penuh antara dua tanggal.
- NETWORKDAYS.INTL mengembalikan jumlah hari kerja penuh antara dua tanggal menggunakan parameter untuk menentukan akhir pekan.
- TIMEVALUE mengonversi waktu dalam bentuk teks menjadi nomor seri.
- WORKDAY mengembalikan jumlah tanggal sebelum atau sesudah sejumlah hari kerja tertentu.
- WORKDAY.INTL mengembalikan jumlah tanggal sebelum atau sesudah sejumlah hari kerja tertentu menggunakan parameter untuk menentukan akhir pekan.
- YEARFRAC mengembalikan pecahan tahun yang mewakili jumlah hari penuh antara

awal dan tanggal.

## 7.2 APLIKASI PRAKTIS

Berikut ini, perlu diingat bahwa tanggal di Excel adalah angka yang diformat sebagai tanggal, dan mewakili jumlah hari sejak 1 Januari 1900. Kolom teks mungkin tampak seperti tanggal, tetapi tidak sama dan umumnya tidak memungkinkan manipulasi numerik langsung atau perhitungan untuk dilakukan.

### 7.2.1 Contoh: Durasi Tugas, Estimasi Sumber Daya dan Biaya

Gambar 7.1 menunjukkan bahwa dua tanggal dapat dikurangi untuk menemukan jumlah hari di antara keduanya. Untuk tugas yang dimulai pada suatu hari (di awal hari itu) dan berlangsung hingga akhir hari lain, total durasi tugas adalah selisih antara tanggal ditambah 1 (misalnya, tugas yang dimulai pada awal hari ke-2 dan berlangsung hingga akhir hari ke-4 memiliki durasi 3 hari).

Kita juga dapat menggunakan fungsi DAYS untuk menghitung durasi tugas, jika kita ingat bahwa fungsi DAYS mencatat durasi dari awal hingga awal (dan bukan dari awal hingga akhir), sehingga hasilnya mungkin perlu disesuaikan, tergantung pada konteksnya. Durasi tugas dapat dikalikan dengan persyaratan intensitas sumber daya (seperti orang yang setara dengan waktu penuh (FTE) atau biaya per hari, dan seterusnya) untuk menghitung total persyaratan sumber daya (lihat Gambar 7.1).

	A	B	C	D	E	F	G	H	I	J	K
1											
2			Start	Finish	Duration (in Days)		DAYS			FTEs/Day	FTEs
3			Beginning of ...	End of ...							
4		Task 1	13/12/2014	17/01/2015	36	=D4-C4+1	35	=DAYS(D4,C4)		10	360
5		Task 2	03/02/2015	07/04/2015	64	=D5-C5+1	63	=DAYS(D5,C5)		50	3200
6		Task 3	08/04/2015	28/04/2015	21	=D6-C6+1	20	=DAYS(D6,C6)		70	1470
7		Task 4	01/05/2015	16/05/2015	16	=D7-C7+1	15	=DAYS(D7,C7)		30	480
8		Task 5	17/05/2015	26/06/2015	41	=D8-C8+1	40	=DAYS(D8,C8)		10	410
9		Total			178	=SUM(E4:E8)					5920

**Gambar 7.1 Penggunaan Perhitungan Tanggal dan Fungsi DAYS dalam Durasi Tugas dan Estimasi Sumber Daya**

### Contoh: Melacak Pemesanan, Reservasi, atau Aktivitas Lainnya

Fungsi DATE mengubah input yang sesuai dengan tahun, bulan, dan hari menjadi tanggal. Gambar 7.2 menunjukkan contoh, di mana tahun, bulan, dan tanggal dari setiap set reservasi hotel dimasukkan, dari situ fungsi DATE membuat tanggal, dan fungsi DAYS menghitung lama menginap (dibandingkan dengan contoh sebelumnya, tidak perlu menambahkan satu pun ke hasil untuk memperoleh angka yang benar dalam konteks ini) (lihat Gambar 7.2).

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1															
2			ARRIVAL					DEPARTURE					Length of Stay (Nights)		
3			Year	Month	Day	DATE		Year	Month	Day	DATE		Using DAYS		
4	Person 1		2017	3	2	=DATE(D4,E4,F4)	=DATE(D4,E4,F4)	2017	3	9	09/03/2017	=DATE(I4,J4,K4)		7	=DAYS(L4,G4)
5	Person 2		2017	8	23	=DATE(D5,E5,F5)	=DATE(D5,E5,F5)	2017	9	6	06/09/2017	=DATE(I5,J5,K5)		14	=DAYS(L5,G5)
6	Person 3		2017	4	18	=DATE(D6,E6,F6)	=DATE(D6,E6,F6)	2017	4	28	28/04/2017	=DATE(I6,J6,K6)		10	=DAYS(L6,G6)

Gambar 7.2 Menggunakan DAYS untuk Menghitung Lama Menginap Tamu Hotel

### 7.2.2 Contoh: Membuat Sumbu Waktu yang Tepat

Fungsi EOMONTH dan EDATE dapat digunakan untuk membuat sumbu waktu yang tepat. Misalnya, EOMONTH menghitung tanggal bulan untuk sejumlah bulan tertentu setelah tanggal awal. Jika tanggal awal ditetapkan sebagai awal bulan, maka fungsi tersebut dapat digunakan untuk menghitung periode bulan penuh guna membuat sumbu waktu secara otomatis. EDATE menghitung tanggal yang merupakan sejumlah bulan tertentu setelah tanggal awal (misalnya, 15 Maret adalah dua bulan setelah 15 Januari). Ini berguna jika periode model tidak selaras dengan awal bulan atau tahun. Selain itu, pengetahuan tentang jumlah hari yang tepat dalam suatu periode mungkin diperlukan untuk perkiraan terperinci, seperti perhitungan angka total produksi setiap kuartal, saat proses produksi berjalan setiap hari atau 24 jam per hari.

Gambar 7.3 menunjukkan contoh di mana sumbu triwulanan dibuat menggunakan rumus berdasarkan awal bulan (1 Januari 2017) dan periode waktu yang diasumsikan 3 bulan per kolom (nilai di Sel C3 dapat diubah untuk membuat kolom dengan durasi waktu sebulan penuh: bulanan, triwulanan, tahunan, atau lainnya) (lihat Gambar 7.3). Perhatikan bahwa durasi periode juga dapat dihitung dari tanggal mulai dan berakhir (baik menggunakan selisih antara tanggal atau fungsi DAYS, seperti yang dijelaskan sebelumnya).

	A	B	C	D	E	F	G
1							
2		Period Start	01-Jan-17	01-Apr-17	01-Jul-17	01-Oct-17	=E4+1
3		Months in Period	3	3	3	3	=5C3
4		Period End	31-Mar-17	30-Jun-17	30-Sep-17	31-Dec-17	=EOMONTH(F2,F3-1)
5		Days in Period	90	91	92	92	=F4-F2+1
6							
7		Production Volume Tons/Day	800	805	900	950	
8		Production Volume	72000	73255	82800	87400	=F7*F5
9							
10		Sales					
11		Costs....					
12							

Gambar 7.3 Menggunakan EOMONTH untuk Membuat Sumbu Waktu



	A	B	C	D	E	F	G
1							
2		Period Start	15-Jan-17	15-Apr-17	15-Jul-17	15-Oct-17	=E4+1
3		Months in Period	3	3	3	3	=C3
4		Period End	14-Apr-17	14-Jul-17	14-Oct-17	14-Jan-18	=EDATE(F2,F3)-1
5		Days in Period	90	91	92	92	=F4-F2+1
6							
7		Production Volume Tons/Day	800	805	900	950	
8		Production Volume	72000	73255	82800	87400	=F7*F5
9							
10		Sales					
11		Costs....					
12							

**Gambar 23.4 Menggunakan EDATE untuk Membuat Sumbu Waktu**

#### Contoh: Menghitung Tahun dan Bulan pada Tanggal

Gambar 7.5 menunjukkan penggunaan fungsi YEAR dan MONTH untuk menentukan tahun dan bulan (angka) mana yang sesuai dengan tanggal (lihat Gambar 7.5).

	A	B	C	D	E	F	G	H
1								
2		Date	YEAR			MONTH		
3		01/04/2015	2015	=YEAR(B3)		4	=MONTH(B3)	
4								

**Gambar 7.5 Contoh Fungsi YEAR dan MONTH**

#### Contoh: Menghitung Kuartal di Mana Suatu Tanggal Terjadi

Meskipun Excel memiliki fungsi YEAR dan MONTH, (pada saat penulisan) tidak ada fungsi QUARTER. Jika memang ada, fungsi tersebut mungkin harus memiliki fleksibilitas untuk menangani berbagai definisi kuartal yang digunakan di seluruh dunia: periode Januari hingga Maret secara luas dianggap sebagai kuartal pertama, namun tahun keuangan Inggris berakhir pada awal April, misalnya.

Gambar 7.6 menunjukkan beberapa cara untuk menghitung kuartal suatu tanggal, setelah terlebih dahulu menghitung bulannya (menggunakan MONTH), dengan asumsi bahwa periode Januari hingga Maret merupakan kuartal pertama. Metode pertama menggunakan fungsi INT, metode kedua menggunakan ROUNDUP, dan metode ketiga menggunakan fungsi pencarian berdasarkan tabel yang telah ditentukan sebelumnya (fungsi-fungsi ini telah dijelaskan sebelumnya dalam teks, kecuali fungsi pencarian, yang dibahas dalam Bab 9).

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
1																
2		Date	MONTH			QUARTER: Method 1			QUARTER: Method 2			QUARTER: Method 3			Month	Quarter
3		01/04/2015	4 =MONTH(B3)			2 =INT((C3-1)/3)+1			2 =ROUNDUP(C3/3,0)			2 =INDEX(\$P\$3:\$P\$14,MA			1	1
4															2	1
5															3	1
6															4	2

**Gambar 23.6** Berbagai Cara Menghitung Kuartal pada Tanggal Tertentu

### Contoh: Membuat Laporan dan Model Berbasis Waktu dari Kumpulan Data

Dengan tahun, bulan, dan kuartal dari tanggal apa pun, laporan sering kali dapat dibuat dari basis data. Memang, terkadang cara yang paling tepat untuk menyusun keseluruhan model adalah dengan membuatnya berdasarkan kumpulan data, dengan model pada dasarnya berupa serangkaian laporan (dan mungkin beberapa kalkulasi lebih lanjut berdasarkan laporan tersebut); pendekatan pemodelan ini dibahas dalam Bab 9.

Gambar 7.7 menunjukkan contoh di mana kumpulan data item yang berkaitan dengan penyewaan properti (pendapatan sewa, biaya agen, biaya perbaikan, dll.) digunakan untuk membuat laporan triwulanan tentang pendapatan, total biaya, dan laba. Tanggal digunakan untuk menghitung tahun, bulan, dan kuartal, yang darinya kunci teks (seperti "2014 Q1") dibuat. Laporan (dalam rentang K4:M12) dibuat menggunakan fungsi SUMIFS untuk menjumlahkan item dalam kumpulan data berdasarkan kunci tersebut. Akhirnya, laba untuk setiap periode triwulanan dihitung dari laporan ini (Kolom N) (lihat Gambar 7.7).

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1		Database of revenue and costs feeds summary calculation of revenue and costs, which is then used to calculate quarterly profit												
2		Items can be added at any point in the database, not only in time order												
3														
4		Date	Item	Category	Amount	Month	Year	Quarter	Year-Quarter Key		Quarter	Revenue	Cost	Profit
5		11-Mar-13	Rental Income	Revenue	1064.52	3	2013	1	2013 Q1		2013 Q1	1065	2211	-1147
6		11-Mar-13	Agents Fees (Mgmt)	Cost	5130	3	2013	1	2013 Q1		2013 Q2	9000	4240	4760
7		11-Mar-13	Agents Fees (Rental upfront)	Cost	2860.00	3	2013	1	2013 Q1		2013 Q3	9000	3390	5610
8		1-Apr-13	Rental Income	Revenue	3000.00	4	2013	2	2013 Q2		2013 Q4	9000	3390	5610
9		1-Apr-13	Agents Fees (Mgmt)	Cost	150.00	4	2013	2	2013 Q2		2014 Q1	9000	7423	1577
10		1-May-13	Rental Income	Revenue	3000.00	5	2013	2	2013 Q2		2014 Q2	9000	6200	2800
11		1-May-13	Agents Fees (Mgmt)	Cost	150.00	5	2013	2	2013 Q2		2014 Q3	9000	6200	2800
12		1-Jun-13	Rental Income	Revenue	3000.00	6	2013	2	2013 Q2		2014 Q4	9000	6200	2800
13		1-Jun-13	Agents Fees (Mgmt)	Cost	150.00	6	2013	2	2013 Q2					

**Gambar 23.7** Membuat Laporan dan Model dari Kumpulan Data Berbasis Waktu

### Contoh: Mencari Tahu Hari Apa dalam Seminggu Anda Lahir

Fungsi WEEKDAY dapat digunakan untuk mencari hari apa dalam seminggu pada tanggal tertentu. Perhatikan bahwa konvensi default adalah minggu dimulai pada hari Minggu; jika seseorang ingin minggu dimulai pada hari Senin, maka argumen opsional (2) dapat digunakan.

### Contoh: Menghitung Tanggal Jumat Terakhir Setiap Bulan

Fungsi Tanggal dapat digunakan dalam kombinasi. Misalnya, untuk perhitungan perencanaan arus kas terperinci, seseorang mungkin ingin mengetahui tanggal pasti Jumat terakhir setiap bulan (misalnya, sebagai tanggal pembayaran gaji perusahaan).

	A	B	C	D
1				
2		Date	WEEKDAY	
3		28 January 1974	2	=WEEKDAY(B3)
4		28 January 1974	1	=WEEKDAY(B4,2)

**Gambar 23.8 Contoh Penggunaan Fungsi WEEKDAY**

Gambar 7.9 menunjukkan serangkaian perhitungan yang melibatkan fungsi EOMONTH dan WEEKDAY, yang menghitung tanggal Jumat terakhir bulan tersebut berdasarkan tanggal input apa pun dalam bulan tersebut (lihat Gambar 7.9).

	A	B	C	D	E	F	G
1							
2		Date	EOMONTH	WEEKDAY	Next Friday	EOMONTH	LastFriday
3		01 February 2016	29/02/2016	1	04/03/2016	31/03/2016	26/02/2016

**Gambar 23.9 Menghitung Tanggal Jumat Terakhir Bulan untuk Tanggal Tertentu**

	A	B	C	D	E	F	G	H
1								
2		First Date	Second Date	Interval	DATEDIF			Between dates:
3		01-Jan-15	01-Mar-15	d	59	=DATEDIF(B3,C3,D3)		Number of days
4		01-Jan-16	01-Mar-16	d	60	=DATEDIF(B4,C4,D4)		Number of days
5		04-Jan-16	03-Mar-16	m	1	=DATEDIF(B5,C5,D5)		Complete calendar months
6		01-Jan-16	05-Mar-18	y	2	=DATEDIF(B6,C6,D6)		Complete calendar years
7		01-Jan-16	05-Mar-18	ym	2	=DATEDIF(B7,C7,D7)		Complete months, ignoring year
8		01-Jan-16	01-Mar-23	yd	60	=DATEDIF(B8,C8,D8)		Complete days, ignoring year
9		01-Jan-16	05-Mar-16	md	4	=DATEDIF(B9,C9,D9)		Complete days, ignoring month and year

**Gambar 23.10 Contoh Fungsi DATEDIF**

#### Contoh: Fungsi DATEDIF dan Periode Waktu yang Selesai

Pada contoh sebelumnya, kami menggunakan fungsi EOMONTH dan WEEKDAY untuk menghitung tanggal Jumat terakhir dalam bulan apa pun. Contoh serupa lainnya melibatkan perhitungan jumlah bulan yang telah dimulai sejak tanggal tertentu (yaitu bulan yang telah selesai sepenuhnya ditambah bulan yang dipertimbangkan; banyak kontrak yang mengharuskan pembayaran penuh bulan tersebut segera setelah bulan dimulai). Beberapa pendekatan dapat dilakukan:

- Hitung jumlah bulan yang dimulai dengan menggunakan EDATE (bukan EOMONTH), yang dikombinasikan dengan YEAR dan MONTH. Latihan ini diserahkan kepada pembaca.
- Gunakan fungsi DATEDIF. Fungsi ini menghitung selisih antara dua tanggal dalam berbagai interval yang berbeda, seperti jumlah tahun, bulan, atau hari di antara tanggal-tanggal tersebut. Misalnya, jika diberikan dua tanggal, fungsi ini dapat

memberikan jumlah bulan yang telah selesai (yaitu jumlah bulan yang dimulai). Meskipun fungsi ini berpotensi hebat, tidak jelas (pada saat penulisan) sejauh mana fungsi ini merupakan fungsi yang didukung sepenuhnya (misalnya, meskipun tercantum di situs web Dukungan Microsoft Office saat mencari fungsi Tanggal Excel, fungsi ini tidak muncul di menu fungsi Tanggal Excel).

## BAB 8

### FUNGSI DAN FUNGSIONALITAS TEKS

#### 8.1 PENDAHULUAN

Bab ini membahas fungsi Teks Excel. Fungsi ini dapat berguna dalam banyak konteks, termasuk memfasilitasi manipulasi bidang numerik (seperti dengan mengubahnya terlebih dahulu menjadi teks, mengoperasikan bidang teks yang dihasilkan, dan mengubah teks baru kembali menjadi bidang numerik). Fungsi yang menjadi fokus kami adalah:

- CONCAT (dan CONCATENATE yang lama) menggabungkan teks dari beberapa rentang dan/atau string. CONCAT juga memungkinkan rentang beberapa sel untuk digabungkan menjadi satu sel. TEXTJOIN memiliki fungsi tambahan yang memungkinkan pembatas tertentu digunakan (setara dengan penggabungan jika tidak ada pembatas yang ditentukan), dan untuk menentukan apakah sel kosong akan diabaikan atau tidak.
- VALUE mengonversi bidang teks yang "berpenampilan numerik" menjadi angka yang mirip. NUMBERVALUE mengonversi teks menjadi angka dengan cara yang tidak bergantung pada lokal, dengan mengizinkan spesifikasi bidang yang menentukan desimal dan pemisah grup. T mengembalikan argumen teks sebagai teks yang sama, jika tidak, ia mengembalikan sel teks kosong.
- LEFT dan RIGHT mengembalikan karakter paling kiri dan paling kanan dari bidang teks, dan MID mengembalikan sejumlah karakter tertentu dari tengah bidang teks.
- LEN mengembalikan jumlah karakter dalam bidang teks.
- FIND menemukan satu nilai teks dalam yang lain (peka huruf besar/kecil). SEARCH serupa tetapi tidak peka huruf besar/kecil.
- REPLACE mengganti semua karakter (antara dan titik awal dan akhir yang ditentukan) dengan teks baru yang ditentukan. SUBSTITUTE mengganti karakter tertentu dengan karakter alternatif (secara default) dalam keseluruhan bidang teks yang diberikan, dan mengizinkan pengguna untuk menentukan hanya sebagian dari contoh yang harus diganti.
- REPT mengulang teks sejumlah kali tertentu.
- TRIM menghapus karakter spasi ASCII 7-bit (nilai kode 32) kecuali spasi tunggal di antara kata-kata. CLEAN menghapus karakter ASCII 7-bit (yang tidak dapat dicetak) (nilai kode 0 hingga 31).
- CHAR dan UNICHAR menghasilkan karakter ASCII dan Unicode untuk nomor kode tertentu, dan CODE dan UNICODE menyediakan nomor kode untuk karakter (atau karakter pertama dalam string teks).
- LOWER dan UPPER mengonversi teks menjadi huruf kecil dan huruf besar. PROPER mengkapitalisasi huruf pertama di setiap kata dalam bidang teks, dan menyajikan huruf berikutnya di setiap kata dalam huruf kecil.

- EXACT memeriksa untuk melihat apakah dua nilai teks identik.
- TEXT mengonversi angka menjadi teks, menggunakan format yang ditentukan.

Fungsi Teks lainnya yang tidak secara eksplisit dicakup dalam teks ini meliputi ASC (yang mengubah huruf Inggris bita ganda dalam string karakter menjadi karakter bita tunggal), DBCS (yang mengubah huruf Inggris bita tunggal dalam string karakter menjadi karakter bita ganda), BAHTTEXT (yang mengonversi angka menjadi teks, menggunakan format mata uang baht), PHONETIC (yang mengekstrak karakter fonetik dari string teks), FIXED (yang memformat angka sebagai teks dengan jumlah desimal yang tetap) dan DOLLAR (yang mengonversi angka menjadi teks, menggunakan format mata uang \$ (dolar)). Fungsi seperti LEFTB, LENB, RIGHTB, MIDB, FINDB, SEARCHB, dan REPLACEB diperuntukkan bagi bahasa yang menggunakan set karakter double-byte (termasuk Jepang, Cina, dan Korea).

## 8.2 APLIKASI PRAKTIS

### 8.2.1 Contoh: Menggabungkan Teks Menggunakan CONCAT dan TEXTJOIN

Ada berbagai cara untuk menggabungkan kolom teks. Cara yang paling sederhana adalah operator "&", yang memilih setiap kolom teks satu per satu. Ini mirip dengan bentuk dasar fungsi CONCAT, serta CONCATENATE yang lama. Fungsi CONCAT juga dapat digunakan untuk menggabungkan elemen individual dari rentang multisel, sedangkan fungsi CONCATENATE tidak dapat melakukannya.

	A	B	C	D	E	F	G	
1								
2		Country	Country Code	Area Code	Number	Result	Formula	
3		UK		44		456789	UK44456789	=B3&C3&D3&E3
4		UK		44		456789	UK44456789	=CONCAT(B4,C4,D4,E4)
5		UK		44		456789	UK44456789	=CONCATENATE(B5,C5,D5,E5)
6		UK		44		456789	UK44456789	=CONCAT(B6:E6)
7		UK		44		456789	#VALUE!	=CONCATENATE(B7:E7)
8		UK		44		456789	UK44456789	=TEXTJOIN(,0,B8,C8,D8,E8)
9		UK		44		456789	UK44456789	=TEXTJOIN(,0,B9:E9)
10		UK		44		456789	UK-44-456789	=TEXTJOIN("-",1,B10:E10)
11		UK		44		456789	UK-44-456789	=TEXTJOIN("-",0,B11:E11)

**Gambar 8.1 Contoh Fungsi CONCAT, TEXTJOIN dan Fungsi Terkait**

Pendekatan yang paling umum dan fleksibel adalah dengan menggunakan fungsi TEXTJOIN. Ini memungkinkan sel atau rentang individual untuk digabungkan, untuk menggunakan pembatas di antara elemen yang digabungkan, dan untuk menentukan apakah sel kosong akan diabaikan atau tidak (tanpa kemungkinan penggunaan pembatas, fungsi terakhir ini tidak diperlukan, dan defaultnya adalah mengabaikan sel kosong). Saat menggabungkan kolom teks, sering kali lebih baik melakukannya dengan pemisah. Melakukan hal itu tidak hanya memudahkan pembacaan tetapi juga akan memudahkan proses untuk membagi kolom hasil secara penuh (misalnya, jika hasil gabungan disalin ke berkas lain, dan data asli yang dipisah menjadi hilang).

Gambar 8.1 menunjukkan contoh (lihat Gambar 8.1). Perhatikan khususnya bahwa di

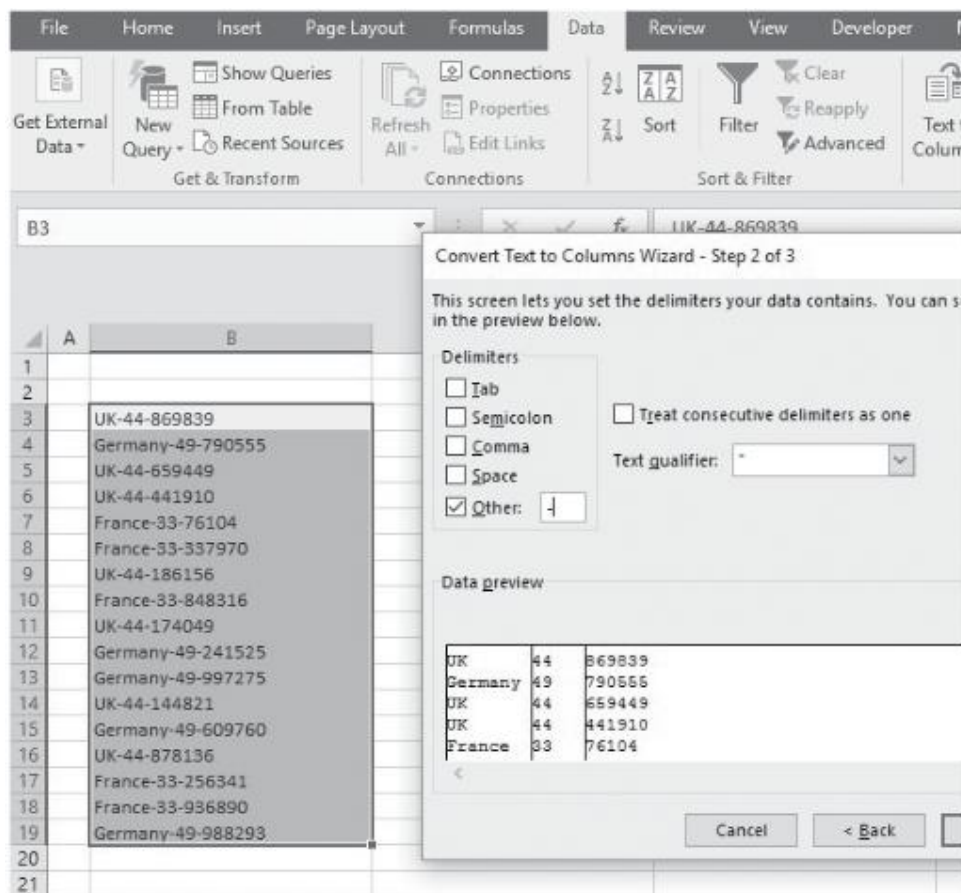
Sel F10 (yang rumusnya ditampilkan di sel G10), fungsi TEXTJOIN diatur sehingga spasi diabaikan, dan karenanya pemisah muncul secara berurutan hanya sekali, sedangkan di Sel F11 (yang rumusnya ditampilkan di sel G11), spasi tidak diabaikan, dan karenanya pemisah muncul secara berurutan karena Kolom D kosong.

### 8.2.2 Contoh: Memisahkan Data Menggunakan Panduan Text-to-columns

Menu Text-to-Columns langkah demi langkah (Panduan) pada tab Data Excel memungkinkan satu set data dipecah menjadi beberapa komponen secara satu kali (yaitu tanpa hubungan fungsional atau dinamis antara data asli dan hasilnya). Proses ini terdiri dari tiga langkah:

- Pemilihan data, dan apakah akan dipecah menjadi komponen berukuran tetap atau berdasarkan keberadaan pembatas.
- Penetapan pemisahan tetap atau jenis pembatas.
- Pemilihan tempat untuk meletakkan data dan formatnya (default akan menimpa set data asli).

Gambar 8.2 memperlihatkan satu set data dan hasil pemisahan di mana data baru ditempatkan ke dalam rentang yang dimulai di Sel C3. Gambar 8.2 memperlihatkan langkah kedua dari proses ini.



Gambar 8.2 Langkah Kedua dari Wizard Konversi Teks ke Kolom

### Contoh: Mengubah Teks Numerik Menjadi Angka

Fungsi VALUE mengubah teks yang tampak numerik menjadi nilai yang sesuai. Faktanya, Excel sering memperlakukan teks (yang tampak numerik) tersebut seolah-olah merupakan angka. Namun, dalam beberapa kasus kesalahan atau perilaku yang tidak diharapkan dapat terjadi (seperti saat menggunakan fungsi Basis Data Excel, di mana basis data berisi teks yang tampak numerik tetapi kriteria yang diuji ditulis menggunakan bidang numerik; lihat Bab 10). Oleh karena itu, secara umum, lebih kuat untuk memastikan bahwa transformasi penuh menjadi bidang numerik telah dilakukan secara eksplisit. Gambar 8.3 menunjukkan contoh (lihat Gambar 8.3) di mana bidang teks "23" di Sel B3 digunakan dalam rumus sederhana (di Sel C3), dengan Excel secara implisit menafsirkan angka ini seolah-olah merupakan nilai untuk tujuan rumus.

	A	B	C	D
1				
2		<b>Text</b>	<b>Result</b>	<b>Formula</b>
3		23	23	=B3*1
4		23	23	=VALUE(B3)
5		1987/05/13	31910	=B5*1
6		1987/05/13	31910	=VALUE(B6)
7				
8		23	23	=NUMBERVALUE(B8)
9		1987/05/13	31910	=NUMBERVALUE(B9)
10		2p500	2.5	=NUMBERVALUE(B10,"p")

**Gambar 8.3 Menggunakan Fungsi VALUE dan NUMBERVALUE**

Sel C4 menunjukkan bahwa fungsi VALUE secara eksplisit mengubah bidang teks tersebut menjadi nilainya (petunjuk bahwa B4 adalah teks, tetapi C4 adalah nilai diberikan oleh perataan default Excel, dengan teks diratakan ke kiri dan angka diratakan ke kanan). Komentar serupa berlaku untuk bidang teks yang tampaknya terlihat seperti tanggal (misalnya bilangan bulat). Fungsi NUMBERVALUE serupa, tetapi memungkinkan parameter tambahan untuk digunakan dalam aplikasi khusus (atau penanganan yang tidak bergantung pada bahasa), di mana pemisah desimal dan grup dapat didefinisikan secara eksplisit.

### Contoh: Pemisahan Teks Secara Dinamis menjadi Komponen I

Fungsionalitas Text-to-Columns yang ditunjukkan sebelumnya paling ampuh untuk analisis satu kali. Fungsi ini juga hanya benar-benar praktis jika kesamaan yang menentukan bagaimana komponen dibatasi (misalnya dengan tetap, atau pembatas standar) memiliki bentuk yang relatif sederhana. Dalam banyak kasus, akan lebih tepat untuk membagi data menggunakan fungsi: ini akan membuat serangkaian kalkulasi langsung yang menghubungkan data (yang digabungkan) ke data yang dibagi (dan karenanya dapat digunakan kembali jika data diperbarui).

Ini juga berpotensi memberikan pendekatan yang lebih fleksibel dan canggih jika



kesamaan yang diperlukan untuk mengidentifikasi titik-titik pemisahan bersifat lebih kompleks. Fungsi LEFT dan RIGHT menyediakan metode yang sederhana dan pada dasarnya dapat dijelaskan sendiri untuk memisahkan karakter kiri dan kanan dari bidang data. Jumlah karakter dapat diketahui sebelumnya (seperti memisahkan karakter paling kiri atau paling kanan), atau mungkin perlu ditentukan dengan metode lain.

	A	B	C	D	E	F	G	H
1								
2		Combined Loan-to-Income						
3		Loan Balance	& Risk History	Loan-to-income	Formula	Risk History	Formula	
4		49,878		E1	E	=LEFT(\$C3,1)	1	=VALUE(RIGHT(\$C3,1))
5		4,433		A1	A	=LEFT(\$C4,1)	1	=VALUE(RIGHT(\$C4,1))
6		52,774		E3	E	=LEFT(\$C5,1)	3	=VALUE(RIGHT(\$C5,1))
7		41,708		C3	C	=LEFT(\$C6,1)	3	=VALUE(RIGHT(\$C6,1))
8		89,999		G1	G	=LEFT(\$C7,1)	1	=VALUE(RIGHT(\$C7,1))
9		38,299		F2	F	=LEFT(\$C8,1)	2	=VALUE(RIGHT(\$C8,1))
10		43,900		G1	G	=LEFT(\$C9,1)	1	=VALUE(RIGHT(\$C9,1))

**Gambar 8.4 Menggunakan Fungsi KIRI dan KANAN**

Gambar 8.4 menunjukkan contoh di mana bidang teks di Kolom C dibagi menjadi karakter tunggal paling kiri dan paling kanan, dengan bidang yang tampak numerik di sebelah kanan juga diubah menjadi nilai asli dengan menggunakan fungsi VALUE yang diterapkan pada hasil fungsi KANAN (lihat Gambar 8.4).

Juga ditunjukkan dalam file (tetapi tidak dalam klip layar) adalah penerapan kumpulan data yang dihasilkan untuk membuat penjumlahan bersyarat (menggunakan SUMIFS), seperti jumlah saldo pinjaman untuk kombinasi tertentu dari rasio pinjaman terhadap pendapatan dan riwayat risiko. Pembaca dapat memverifikasi bahwa fungsi SUMIFS akan tetap menghasilkan nilai yang benar meskipun fungsi VALUE tidak diterapkan.

#### **Contoh: Memisahkan Teks Secara Dinamis menjadi Komponen II**

Fungsi MID mengembalikan bagian dari string teks yang dimulai pada titik tertentu dan memiliki panjang tertentu. Fungsi ini praktis digunakan saat mengekstrak teks yang berada di tengah kolom teks. Gambar 8.5 menunjukkan aplikasi sederhana yang membagi rangkaian digit menjadi komponen-komponennya. Dalam kasus ini, setiap komponen memiliki panjang 1, dengan titik awal pemisahan string ditentukan oleh bilangan bulat di Baris 2 (lihat Gambar 5.5). Sekali lagi, fungsi VALUE digunakan untuk memastikan bahwa hasilnya terdiri dari nilai numerik asli.

	A	B	C	D	E	F	G	H	I
1									
2		Number	1	2	3	4	5	6	Formula
3		201739	2	0	1	7	3	9	=VALUE(MID(\$B3,H\$2,1))
4		103401	1	0	3	4	0	1	=VALUE(MID(\$B4,H\$2,1))
5		208736	2	0	8	7	3	6	=VALUE(MID(\$B5,H\$2,1))

**Gambar 8.5 Menggunakan Fungsi MID untuk Mengekstrak Bagian dari Bidang Teks**

	A	B	C	D	E	F	G	H	I	J	K
1											
2		Data		Chars Reqd	LEFT	Left via MID		Chars Reqd	RIGHT	Right via MID	
3		201739		2	20	20		2	39	39	=MID(B3,LEN(B3)-H3+1,H3)
4		103401		2	10	10		2	01	01	=MID(B4,LEN(B4)-H4+1,H4)
5		208730		2	20	20		2	36	36	=MID(B5,LEN(B5)-H5+1,H5)
6											

Gambar 8.6 Membandingkan LEFT, RIGHT, MID dan LEN

### 8.2.3 Contoh: Membandingkan LEFT, RIGHT, MID, dan LEN

Fungsi MID pada dasarnya adalah bentuk LEFT dan RIGHT yang lebih umum. Jika digunakan di mana titik awalnya adalah karakter pertama, maka pada dasarnya fungsi ini hampir identik dengan fungsi LEFT. Fungsi ini juga dapat dibuat serupa dengan fungsi RIGHT, dengan terlebih dahulu menggunakan LEN untuk menentukan panjang string penuh, yang darinya karakter awal untuk kolom yang akan diekstrak dapat ditentukan. Misalnya, jika seseorang ingin mengekstrak N karakter paling kanan, menggunakan MID dan bukan RIGHT, maka nomor awal karakter untuk proses ekstraksi diberikan oleh:

$$\text{StartNum} = \text{Lenght of FullString} - N + 1$$

Gambar 8.6 menunjukkan contoh rumus tersebut (lihat Gambar 8.6). Dalam contoh ini, untuk tetap fokus pada fungsionalitas komparatif, fungsi VALUE tidak digunakan.

#### Contoh: Memisahkan Teks Secara Dinamis ke dalam Komponen III

Fungsi SEARCH dapat berguna jika posisi yang menggambarkan karakter awal atau akhir dari bagian string teks tidak ditetapkan atau diketahui sebelumnya, dan perlu ditemukan. Gambar 8.7 menunjukkan contoh. Fungsi SEARCH digunakan untuk menemukan posisi tanda kurung di dalam setiap titik set data (di sel B7 ke bawah).

Tanda kurung adalah elemen umum dari setiap titik data yang posisinya memungkinkan nama pelanggan dan durasi kursus diidentifikasi misalnya CustA menjadi A menggunakan fungsi MID berdasarkan dua karakter sebelum tanda kurung, dan durasi kursus menggunakan fungsi MID berdasarkan karakter awal satu posisi setelah tanda kurung. Dari sini, laporan ringkasan (seperti menggunakan fungsi COUNTIFS untuk menghasilkan ringkasan total kursus menurut pelanggan berdasarkan durasi) dapat dibuat (lihat Gambar 8.7).

Perhatikan juga bahwa jika seseorang mencoba menghasilkan set data yang sama dengan membagi data menggunakan Text-to-Columns, proses tersebut perlu dilakukan beberapa kali untuk setiap set data, karena pembatas yang berbeda akan diperlukan dalam setiap kasus.

	A	B	C	D	E	F	G	H
1								
2		Customer	D					
3		Length	2					
4		Number of Courses	2	=COUNTIFS(F7:F25,C2,H7:H25,C3)				
5								
6		Cust & Length		SEARCH for Position of "("	Formula	Cust	Days as Text	Days as Value
7		CustA (1 Day)			=SEARCH("(",B7,1)	A	1	1
8		CustomerB (2 Day)			=SEARCH("(",B8,1)	B	2	2
9		CustC (1 Day)			=SEARCH("(",B9,1)	C	1	1
10		CustoD (2 Days)			=SEARCH("(",B10,1)	D	2	2
11		CustB (2 Days)			=SEARCH("(",B11,1)	B	2	2
12		CustomerB (2 Days)			=SEARCH("(",B12,1)	B	2	2
13		CustomerE (2 Days)			=SEARCH("(",B13,1)	E	2	2
14		CustF (2 Days)			=SEARCH("(",B14,1)	F	2	2
15		CustA (2 Days)			=SEARCH("(",B15,1)	A	2	2
16		CustomG (2 Days)			=SEARCH("(",B16,1)	G	2	2
17		CustA (1 Day)			=SEARCH("(",B17,1)	A	1	1
18		CustA (2 Days)			=SEARCH("(",B18,1)	A	2	2
19		CustmG (2 Days)			=SEARCH("(",B19,1)	G	2	2
20		CusA (1 Days)			=SEARCH("(",B20,1)	A	1	1
21		CustD (2 Days)			=SEARCH("(",B21,1)	D	2	2
22		CustB (3 Days)			=SEARCH("(",B22,1)	B	3	3

Gambar 8.7 Menggunakan Fungsi SEARCH dan MID yang Dikombinasikan

#### Contoh: Membandingkan FIND dan SEARCH

Fungsi FIND dan SEARCH memiliki konsep yang serupa, tetapi FIND peka huruf besar/kecil sedangkan SEARCH tidak; jelas, seseorang perlu berhati-hati untuk memastikan bahwa fungsi yang tepat digunakan, sesuai dengan keadaan.

#### 8.2.4 Contoh: Fungsi UPPER dan LOWER

Fungsi UPPER dan LOWER cukup mengubah argumen teksnya menjadi huruf besar dan kecil, dan pada dasarnya sudah jelas. Gambar 8.7 menunjukkan contoh terpadu, yang menggunakan fungsi SEARCH, MID, dan LEN. Proses keseluruhan mengekstrak huruf pertama dan bagian yang tersisa dari setiap nama dari set data, menggabungkannya kembali sehingga hanya nama yang ditampilkan, dan menyajikan nama dengan huruf kapital awal dan bukan huruf kapital setelahnya.

#### Contoh: Fungsi PROPER

Gambar 8.8 menunjukkan bagaimana hasil pada contoh sebelumnya dapat dicapai secara langsung dengan fungsi PROPER. Perhatikan juga bahwa (tidak ditampilkan dalam contoh) fungsi tersebut akan mengkapitalisasi huruf pertama di setiap kata kalimat (bukan hanya huruf pertama, misalnya).

	A	B	C	D	E	F	G	H
1								
2		Co./subject		Position of "S" using SEARCH	Formula		Position of "S" using FIND	Formula
3		CustA (1 Day)			=SEARCH("S",B3,1)		#VALUE!	=FIND("S",B3,1)
4		CustomerS (2 Day)			=SEARCH("S",B4,1)			=FIND("S",B4,1)
5		CustC (1 Day)			=SEARCH("S",B5,1)		#VALUE!	=FIND("S",B5,1)
6		CustoD (2 Days)			=SEARCH("S",B6,1)		#VALUE!	=FIND("S",B6,1)
7		CustB (2 Days)			=SEARCH("S",B7,1)		#VALUE!	=FIND("S",B7,1)

Gambar 8.8 Membandingkan Fungsi FIND dan SEARCH

	A	B	C	D	E	F	G	H	I	J
1										
2		Name	Position of Space	First Letter using MID	UPPER	Formula	Length after Space	Remainder in LOWER	Formula	First Name Pure
3		1 amEliA	2 a	A	=UPPER(D3)		6 melia		=LOWER(MID(B3,C3+2,G3-1))	Amelia
4		2 oLiVia	2 o	O	=UPPER(D4)		6 livia		=LOWER(MID(B4,C4+2,G4-1))	Olivia
5		3 emily	2 e	E	=UPPER(D5)		5 mily		=LOWER(MID(B5,C5+2,G5-1))	Emily
6		4 ava	2 a	A	=UPPER(D6)		3 va		=LOWER(MID(B6,C6+2,G6-1))	Ava
7		5 Isla	2 i	I	=UPPER(D7)		4 sla		=LOWER(MID(B7,C7+2,G7-1))	Isla

Gambar 8.9 Menggunakan Fungsi UPPER dan LOWER

	A	B	C	D	E	F
1						
2		Name	Position of Space	Length after Space	First Name Pure	Formula
3		1 amEliA		2	6 Amelia	=PROPER(MID(B3,C3+1,D3))
4		2 oLiVia		2	6 Olivia	=PROPER(MID(B4,C4+1,D4))
5		3 emily		2	5 Emily	=PROPER(MID(B5,C5+1,D5))
6		4 ava		2	3 Ava	=PROPER(MID(B6,C6+1,D6))

Gambar 8.10 Contoh Fungsi PROPER

### 8.2.5 Contoh: Fungsi EXACT

Fungsi EXACT memeriksa apakah dua bidang teks identik, termasuk memeriksa spasi kosong yang mungkin tidak langsung terlihat. Tentu saja, fungsi ini peka huruf besar/kecil. Berkas Ch24.11.EXACT.xlsx menunjukkan beberapa contoh (lihat Gambar 8.11). Fungsi LEN digunakan untuk mengilustrasikan bahwa ada spasi kosong di beberapa bidang. Perhatikan bahwa C5 berisi spasi kosong di belakangnya, sehingga panjang kata adalah 7. Hal ini tidak akan terlihat tanpa penyelidikan terperinci, namun fungsi EXACT mendeteksi adanya perbedaan antara B5 dan C5 (seperti halnya LEN).

	A	B	C	D	E	F	G	H
1								
2		Text1	Text2	EXACT			LEN(Text1)	LEN(Text2)
3		amelia	Amelia	FALSE	=EXACT(B3,C3)		6	6
4		Amelia	Amelia	FALSE	=EXACT(B4,C4)		7	6
5		Amelia	Amelia	FALSE	=EXACT(B5,C5)		6	7

Gambar 8.11 Penggunaan Fungsi EXACT

### Contoh: Membandingkan REPLACE dengan SUBSTITUTE

Fungsi REPLACE dan SUBSTITUTE secara konseptual serupa, tetapi sebenarnya memberikan aplikasi potensial yang cukup berbeda. Fungsi REPLACE akan, dimulai dari karakter yang ditentukan, mengganti semua karakter berikutnya (hingga angka yang ditentukan) dengan teks baru yang ditentukan.

Fungsi SUBSTITUTE akan (secara default dalam keseluruhan bidang teks yang diberikan), mengganti hanya karakter tertentu dengan karakter alternatif. Fungsi ini juga berisi argumen opsional untuk memungkinkan pengguna menentukan hanya sebagian dari contoh

yang harus diganti. Contoh kedua menggunakan SUBSTITUTE menunjukkan kasus di mana parameter contoh opsional telah digunakan untuk mengganti hanya kemunculan kedua dari karakter “.

	A	B	C	D
1				
2		<b>Text</b>	<b>Result</b>	<b>Formula</b>
3		1987.05.13	1987/05.13	=REPLACE(B3,5,1,"/")
4		1987.05.13	1987/13	=REPLACE(B4,5,4,"/")
5				
6		1987.05.13	1987/05/13	=SUBSTITUTE(B6,".","/")
7		1987.05.13	1987.05/13	=SUBSTITUTE(B7,".","/",2)

**Gambar 8.12 Membandingkan REPLACE dengan SUBSTITUTE**

#### Contoh: Fungsi REPT

Fungsi REPT dapat digunakan untuk mengulang bidang teks beberapa kali, dan dapat berguna dalam aplikasi seperti membuat papan skor visual, dasbor, atau tampilan grafis.

	A	B	C	D	E
1					
2		<b>Number</b>	<b>Symbol</b>	<b>Result</b>	<b>Formula</b>
3		10	✓	✓✓✓✓✓✓✓✓✓✓	=REPT(C3,B3)
4		4	*	****	=REPT(C4,B4)
5		3	☹	☹☹☹	=REPT(C5,B5)
6					

**Gambar 8.13 Contoh Fungsi REPT**

#### Contoh: Fungsi CLEAN dan TRIM

Fungsi CLEAN dan TRIM masing-masing menghapus karakter yang tidak dapat dicetak dan spasi dari kolom teks. Fungsi ini khususnya berguna pada tahap awal "pembersihan" data yang mungkin diterima dari sumber lain, seperti internet atau unduhan dari server mainframe. CLEAN menghapus sekumpulan besar karakter, yaitu karakter ASCII 7-bit yang tidak dapat dicetak, yang sesuai dengan nilai kode 0 hingga 31. TRIM menghapus karakter spasi (nilai kode 32), kecuali spasi tunggal di antara kata.

CLEAN juga dapat menghapus beberapa karakter tambahan yang tidak dapat dicetak dalam kumpulan Unicode yang lebih besar. Gambar 8.14 menunjukkan contoh fungsi CLEAN (di Kolom E) yang diterapkan pada bidang teks (di Kolom C), di mana bidang teks ini dihasilkan dari nomor kode menggunakan fungsi CHAR. Teks yang dihasilkan dengan cara ini (Kolom C) berisi karakter yang tidak dapat dicetak yang dikombinasikan dengan kata-kata "Teks Bersih", dan hanya kata-kata ini yang tersisa setelah fungsi CLEAN diterapkan (Kolom E).

Gambar 8.16 menunjukkan contoh fungsi TRIM untuk menghapus spasi awal atau akhir dari berbagai bidang teks. Adanya spasi tersebut dapat dilihat dari panjang setiap kolom, seperti yang terlihat pada Kolom C (menggunakan fungsi LEN), serta dari hasil penerapan fungsi LEFT dan RIGHT pada Kolom D dan Kolom E. Teks yang dipangkas pada Kolom F tidak

memiliki spasi di awal maupun di akhir, dan setiap kolom yang diolah memiliki panjang yang sama.

	A	B	C	D	E	F	G
1							
2		CodeNo	ASCII Item & Clean Text	Formula	CLEAN	Formula	
3		1	[Clean Text	=CHAR(B3)&"Clean Text"	Clean Text	=CLEAN(C3)	
4		2	Clean Text	=CHAR(B4)&"Clean Text"	Clean Text	=CLEAN(C4)	
5		3	[Clean Text	=CHAR(B5)&"Clean Text"	Clean Text	=CLEAN(C5)	
6		4	[Clean Text	=CHAR(B6)&"Clean Text"	Clean Text	=CLEAN(C6)	
7		5	[Clean Text	=CHAR(B7)&"Clean Text"	Clean Text	=CLEAN(C7)	
8		6	[Clean Text	=CHAR(B8)&"Clean Text"	Clean Text	=CLEAN(C8)	
9		7	[Clean Text	=CHAR(B9)&"Clean Text"	Clean Text	=CLEAN(C9)	
10		8	[Clean Text	=CHAR(B10)&"Clean Text"	Clean Text	=CLEAN(C10)	
11		9	Clean Text	=CHAR(B11)&"Clean Text"	Clean Text	=CLEAN(C11)	
12		10	Clean Text	=CHAR(B12)&"Clean Text"	Clean Text	=CLEAN(C12)	
13		11	[Clean Text	=CHAR(B13)&"Clean Text"	Clean Text	=CLEAN(C13)	

**Gambar 8.14 Contoh Penggunaan Fungsi CLEAN I**

	G	H	I	J	K	L
		Unicode	Unicode Item & Clean Text	Formula	CLEAN	Formula
		127	[Clean Text	=UNICHAR(H3)&"Clean Text"	Clean Text	=CLEAN(E3)
		129	Clean Text	=UNICHAR(H4)&"Clean Text"	Clean Text	=CLEAN(E4)
		141	Clean Text	=UNICHAR(H5)&"Clean Text"	Clean Text	=CLEAN(E5)
		143	Clean Text	=UNICHAR(H6)&"Clean Text"	Clean Text	=CLEAN(E6)
		144	Clean Text	=UNICHAR(H7)&"Clean Text"	Clean Text	=CLEAN(E7)
		157	Clean Text	=UNICHAR(H8)&"Clean Text"	Clean Text	=CLEAN(E8)
		160	Clean Text	=UNICHAR(H9)&"Clean Text"	Clean Text	=CLEAN(E9)

**Gambar 8.15 Contoh Penggunaan Fungsi CLEAN II**

### Contoh: Memperbarui Label Model dan Judul Grafik

Sering kali berguna untuk memiliki label yang diperbarui jika asumsi atau nilai model berubah. Misalnya, sel mungkin berisi label teks "Nilai sekarang bersih pada tingkat diskonto 10%", di mana angka 10% adalah asumsi yang terdapat dalam sel yang dapat diubah (misalnya menjadi 12%), dalam hal ini seseorang ingin labelnya berubah yaitu menjadi "Nilai sekarang bersih pada tingkat diskonto 12%".

Ini dapat sangat berguna ketika ada kasus atau sensitivitas yang berbeda yang diperlukan, dan terutama ketika beberapa bagan dari kasus ini akan dicetak atau disajikan dalam laporan. Dalam kasus seperti itu, seseorang mungkin menginginkan agar legenda dan judul bagan berubah secara otomatis saat nilai model berubah. Proses untuk bagan terdiri dari dua langkah:

- Buat, sebagai rumus Excel, label yang diinginkan (legenda dan judul) sehingga diperbarui saat data berubah. Bagian proses ini sama dengan kasus saat tidak ada bagan yang akan dibuat, tetapi seseorang hanya ingin label diubah dengan tepat. Ini mungkin sering memerlukan fungsi TEXT, untuk memformat dengan tepat angka apa

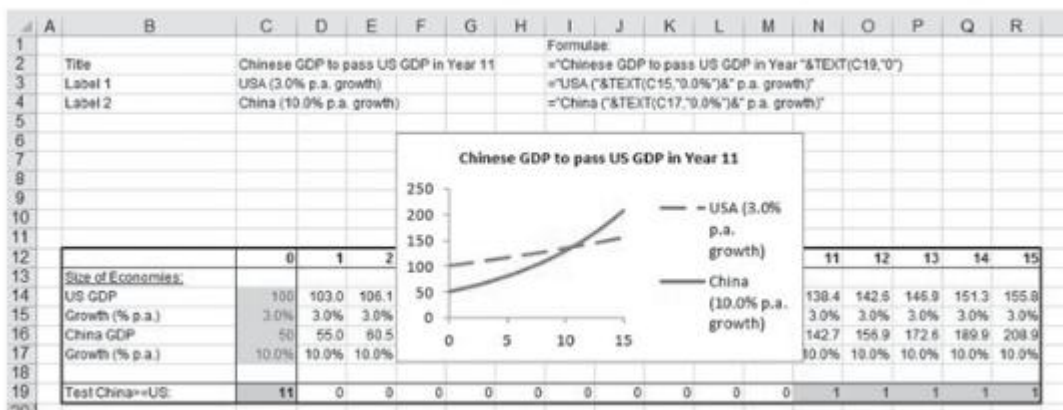
pun yang menjadi bagian dari label ini. Tentu saja, rumus tersebut juga dapat melibatkan fungsi Excel lainnya jika diperlukan (misalnya Bab 6 menunjukkan contoh penggunaan fungsi CELL untuk membuat label pembaruan).

- Tautkan rumus ini ke label dan judul bagan. Ini mudah untuk legenda (dengan mengedit sumber data bagan). Untuk judul bagan, seseorang harus mengedit bidang judul dengan mengetik "=" di dalam Bilah Rumus Excel, arahkan ke sel yang berisi judul yang diperlukan, dan tekan ENTER.

	A	B	C	D	E	F	G	H
1								
2			LEN	1st LEFT	Last RIGHT	TRIM	Formula	LEN
3		Hello World	11	H	d	Hello World	=TRIM(B3)	11
4		Hello World	13	H		Hello World	=TRIM(B4)	11
5		Hello World	12		d	Hello World	=TRIM(B5)	11
6		Hello World	25			Hello World	=TRIM(B6)	11
7		Hello World	17			Hello World	=TRIM(B7)	11
8		Hello World	13			Hello World	=TRIM(B8)	11

**Gambar 8.16 Menggunakan Fungsi TRIM**

Gambar 8.17 menunjukkan contoh (lihat Gambar 8.17). Perhatikan bahwa argumen kedua dari fungsi TEXT menentukan format yang akan digunakan (format lain yang sering digunakan adalah menggunakan "0.0" untuk menampilkan kolom numerik dengan satu titik desimal, atau "0" untuk tidak menampilkan tempat desimal, seperti yang digunakan untuk judul bagan). Beberapa spasi mungkin perlu ditambahkan di tempat yang sesuai dalam kolom teks, misalnya segera setelah tanda petik pembuka atau sebelum tanda petik penutup.



**Gambar 8.17 Memperbarui Label Model dan Judul Grafik Menggunakan TEXT untuk Memformat Bidang Numerik**

### Contoh: Membuat Pengidentifikasi atau Kunci Unik untuk Pencocokan Data

Dalam analisis dan manipulasi data, seseorang sering kali diminta untuk mencocokkan item dalam dua set data dengan cara (unik). Fungsi teks, yang sering dikombinasikan dengan

fungsi Excel lainnya, dapat menjadi cara yang mudah untuk melakukannya.

Quarter	Currency	Amount (local currency)
1	Euro	32,143
3	Pound	17,203
2	Euro	16,993
2	Dollar	78,888
3	Pound	43,957
4	Dollar	25,898
1	Pound	26,856
2	Dollar	63,652
3	Dollar	29,188
4	Euro	47,153
1	Euro	51,737
2	Pound	36,332
3	Euro	33,363
1	Euro	17,635

Orig Key	Xrate€
DollarQr_1	1.60
DollarQs_2	1.58
DollarQt_3	1.55
DollarQ_4	1.62
EuroQ_1	1.20
EuroQ_2	1.18
EuroQ_3	1.22
EuroQ_4	1.25
PoundQ_1	1.00
PoundQ_2	1.00
PoundQ_3	1.00
PoundQ_4	1.00

**Gambar 8.18 Dua Set Data yang Dapat Dihubungkan dengan Membuat Kunci Umum**

Gambar 8.18 menunjukkan contoh, yang bertujuan untuk mengekspresikan dalam mata uang umum (misalnya Poundsterling Inggris) dua set data yang awalnya dalam mata uang lokal mereka (lihat Gambar 8.18). Karena nilai tukar bervariasi menurut kuartal, data dalam dua kolom pertama dari set data pertama (Kuartal dan Mata Uang) dapat digabungkan untuk membuat kunci, sedangkan data dalam kolom pertama dari set data kedua perlu diekstraksi secara selektif dan digabungkan kembali untuk membuat kunci yang cocok. Rumus dalam berkas menunjukkan bagaimana kunci tersebut dapat dibuat (tidak terlihat pada klip layar) menggunakan fungsi seperti SEARCH, LEFT dan RIGHT serta penggabungan sederhana (prosedur untuk melakukan pencocokan dibahas dalam Bab 9).



## BAB 9

### FUNGSI PENCARIAN DAN REFERENSI

#### 9.1 PENDAHULUAN

Bab ini membahas fungsi Pencarian dan Referensi (yang, demi kesederhanaan, disebut dalam teks berikutnya hanya sebagai fungsi pencarian). Pengetahuan yang baik tentang fungsi ini merupakan salah satu kemampuan terpenting yang dibutuhkan untuk membangun model menengah dan lanjutan. Bab ini dimulai dengan contoh-contoh yang berhubungan dengan proses referensi dasar (yang sebagian besar pada dasarnya dapat dijelaskan sendiri, atau telah dibahas sebelumnya dalam teks):

- FORMULATEXT, yang menunjukkan (sebagai teks) rumus dalam sel.
- TRANSPOSE, yang mengubah urutan larik.
- COLUMN (ROW), yang mengembalikan nomor kolom (baris) sel atau rentang.
- COLUMNS (ROWS), yang menemukan jumlah kolom (baris) sel atau rentang.
- ADDRESS, yang menyediakan referensi sel sebagai teks.
- AREAS, yang menunjukkan jumlah area (rentang terpisah yang tidak bersebelahan) dalam suatu referensi.

Sebagian besar bab ini dikhususkan untuk contoh-contoh yang menggunakan fungsi inti lainnya dalam konteks tambahan, termasuk menggabungkan proses pencocokan dan referensi, serta pembuatan rentang dinamis dan struktur data yang fleksibel:

- INDEX mencari nilai dalam baris dan kolom tertentu dari rentang yang bersebelahan (sebagai matriks satu atau dua dimensi). Fungsi ini juga ada dalam bentuk referensi, yang mengembalikan referensi ke sel tertentu, bukan ke nilai sel.
- CHOOSE menggunakan salah satu dari sekumpulan nilai menurut nomor indeksasi. Fungsi ini sangat berguna (dibandingkan dengan fungsi pencarian lainnya) jika argumen berada (atau mungkin perlu berada) dalam rentang yang tidak bersebelahan.
- MATCH menemukan posisi relatif dari nilai tertentu.
- OFFSET memberikan nilai dalam sel yang merupakan sejumlah baris dan kolom tertentu dari sel atau rentang referensi. Nilai ini juga dapat digunakan untuk mengembalikan rentang sel (bukan sel individual) yang merupakan sejumlah baris dan kolom tertentu dari sel atau rentang referensi.
- INDIRECT mengembalikan rentang yang ditentukan oleh string teks.
- HLOOKUP (VLOOKUP) mencari baris atas (kolom kiri) tabel untuk nilai yang ditentukan dan menemukan kolom di baris tersebut (baris di kolom tersebut) yang berisi nilai tersebut. Kemudian, ia menyediakan nilai yang ada di baris (kolom) tertentu dalam tabel.
- LOOKUP mencari nilai dalam vektor atau larik. Dalam bentuk vektornya, ia mencari nilai

yang ditentukan dalam rentang satu dimensi (nilai yang harus dalam urutan menaik) dan mengembalikan nilai dari posisi yang sesuai dalam rentang satu dimensi lainnya. Kami juga menyebutkan secara singkat beberapa fungsi yang menyediakan kemampuan untuk menautkan ke set data, seperti hyperlink dan topik terkait. Penting untuk dicatat bahwa dalam banyak contoh yang diberikan, umumnya tidak ada cara unik untuk mencapai tujuan tertentu; formulasi alternatif dengan fungsi yang berbeda mungkin ada. Oleh karena itu, kami bermaksud untuk menyoroti beberapa kriteria yang dapat dipertimbangkan saat membuat pilihan di antara berbagai pendekatan, topik yang sebagian juga dibahas dalam Bab 9.

## 9.2 APLIKASI PRAKTIS: PROSES REFERENSI DASAR

### Contoh: Fungsi ROW dan COLUMN

Gambar 9.1 menyediakan contoh fungsi ROW dan COLUMN (yang pada dasarnya dapat dijelaskan sendiri) (lihat Gambar 9.1). Perhatikan bahwa penggunaan praktis yang paling sering adalah dalam bentuk inti, yang menentukan nomor baris atau kolom dari satu sel.

	A	B	C	D
1				
2		Result		
3		<b>CORE FORM</b>		
4			4 =ROW(B4)	
5			2 =COLUMN(B5)	
6				
7		<b>ARRAY FORMULAE</b>		
8			4 {=ROW(B4:C5)}	
9			5 {=ROW(B4:C5)}	
10				
11			2	3
12			{=COLUMN(B5:D5)}	{=COLUMN(B5:D5)}

**Gambar 9.1 Penggunaan Rumus Sel Tunggal Dan Array Dengan Fungsi Row Dan Column**

Namun, fungsi tersebut dapat digunakan sebagai rumus array untuk menentukan nilai terkait untuk setiap sel dari rentang multisel. Perlu dicatat juga (untuk penggunaan nanti dalam bab ini) bahwa fungsi tersebut dapat dimasukkan dalam sel yang merujuk dirinya sendiri tanpa membuat lingkaran (misalnya di Sel B4, fungsi yang dimasukkan adalah ROW(B4)).

### Contoh: Fungsi ROWS dan COLUMNS

Gambar 9.2 menyediakan contoh fungsi ROWS dan COLUMNS, yang pada dasarnya juga dapat dijelaskan sendiri (lihat Gambar 9.2). Perlu dicatat (untuk digunakan nanti dalam teks ini) bahwa, sementara ROWS(B3:D7) mengembalikan nilai 5 (yaitu jumlah baris dalam rentang), dalam VBA pernyataan Range("B3:D7").Rows merujuk pada baris sebenarnya dalam rentang (bukan pada jumlahnya). Properti Count dari kumpulan baris inilah yang akan digunakan untuk mengetahui jumlah baris dalam suatu rentang: NRows=Range("B3:D7").Rows.Count

	A	B	C	D
1				
2		DATA		
3		22	14	21
4		11	9	11
5		2	1	7
6		14	14	19
7		5	6	18
8				
9		5	=ROWS(B3:D7)	
10		3	=COLUMNS(B3:D7)	
11				

**Gambar 9.2 Contoh Fungsi Rows Dan Columns**

### Contoh: Penggunaan Fungsi ADDRESS dan Perbandingan dengan CELL

Fungsi ADDRESS mengembalikan alamat sel dalam lembar kerja, dengan nomor baris dan kolom yang ditentukan. Perhatikan bahwa dalam Bab 22, kita melihat bahwa fungsi CELL juga dapat memberikan informasi terkait alamat (misalnya alamat, nomor baris, atau kolom) serta informasi lainnya (misalnya format atau jenis kontennya) tentang sel. Gambar 9.3 menunjukkan contoh fungsi ADDRESS dan hasil analog yang dihasilkan menggunakan bentuk alamat fungsi CELL (lihat Gambar 9.3).

	A	B	C
1			
2		Result	Formula
3		\$B\$3	=ADDRESS(ROW(B3),COLUMN(B3))
4		\$B\$4	=CELL("address",B4)
5			

**Gambar 9.3 Penggunaan Alamat Dan Perbandingan Dengan Sel**

Perhatikan (untuk referensi selanjutnya) bahwa setiap fungsi dapat dimasukkan dalam sel yang merujuk dirinya sendiri tanpa membuat kalkulasi melingkar. Perlu dicatat juga bahwa fungsi CELL memiliki properti Excel yang bersifat Volatile. Ini berarti bahwa fungsi tersebut dievaluasi pada setiap perhitungan ulang lembar kerja bahkan ketika argumennya tidak berubah, yang mengurangi efisiensi komputasi. Dengan demikian, fungsi ADDRESS dapat dipilih sebagai pengganti fungsi CELL dalam beberapa kasus. Gambar 9.4 berisi contoh penggunaan fungsi ADDRESS (lihat Gambar 9.4).

	A	B	C
1	Feb		
2		Result	Formula
3		Feb!\$B\$3	=ADDRESS(ROW(B3),COLUMN(B3),,\$A\$1)
4		Feb!\$B\$4	=\$A\$1&"!"&CELL("address",B4)
5			

**Gambar 9.4 Menemukan Alamat Sel Yang Sesuai Di Lembar Kerja Lain**

Fungsi tersebut menggunakan argumen opsional terakhir dari fungsi tersebut untuk menemukan alamat sel di lembar kerja lain dari buku kerja. Dengan kata lain, fungsi ADDRESS menyediakan (di Sel B3) alamat lengkap Sel B3 dari lembar kerja "Feb". Hasil serupa dapat diperoleh menggunakan bentuk alamat fungsi CELL dengan menggabungkan string teks secara eksplisit. (Pendekatan ini akan penting untuk beberapa contoh model multilembar yang ditunjukkan nanti dalam bab ini, dan dibahas dalam Bab 6.)

### 9.3 APLIKASI PRAKTIS: PROSES REFERENSI LEBIH LANJUT

#### Contoh: Membuat Skenario Menggunakan INDEX, OFFSET atau CHOOSE

Penggunaan teknik skenario pada dasarnya berarti bahwa nilai beberapa input diubah secara bersamaan. Ini biasanya merupakan perluasan dari analisis sensitivitas, yang pada intinya melibatkan perubahan nilai hanya satu variabel. Teknik skenario berguna untuk mulai menangkap serangkaian kemungkinan hasil yang luas, dan untuk menangkap ketergantungan antara variabel yang diyakini ada tetapi sulit untuk direpresentasikan melalui hubungan matematis yang lengkap.

	A	B	C	D	E	F	G	H	I	J
1										
2		Revenues Scenarios	2016	2017	2018	2019	2020	2021	2022	
3		Low			387	407	439	483	520	
4		Base			431	452	488	537	577	
5		High			474	497	537	591	635	
6	3	To Use			474	497	537	591	635	=CHOOSE(\$A6,3,4,5)
7										
8		Model ... Income Statement	2016	2017	2018	2019	2020	2021	2022	
9										
10		Revenues	400	410	474	497	537	591	635	=I6
11										
12		Variable Costs	140	160	170.5	179.0	193.3	212.7	222.3	
13		% revenues	35.0%	39.0%	36.0%	36.0%	36.0%	36.0%	35.0%	
14										
15		Fixed Costs (incl. depreciation)	70	72	72.7	74.2	76.4	78.7	81.1	
16		% Growth		2.9%	1.0%	2.0%	3.0%	3.0%	3.0%	
17										
18		EBIT	190.0	178.0	230.4	244.1	267.3	299.4	331.7	
19										
20		...REST OF MODEL								

Gambar 25.5 Penggunaan Fungsi CHOOSE Untuk Memilih Data Skenario Yang Relevan

Setelah skenario didefinisikan dengan data eksplisit, untuk skenario apa pun, nilai yang akan digunakan dapat dicari dari kumpulan data ini. Penggunaan proses pencarian merupakan alternatif untuk operasi "salin dan tempel" (di mana input model akan diganti secara manual dengan nilai untuk skenario yang diinginkan), dengan fungsi yang menciptakan tautan dinamis antara skenario input dan output model.

Gambar 9.5 menunjukkan contoh penggunaan fungsi CHOOSE (Baris 6) untuk memilih nilai yang berlaku untuk skenario pendapatan yang dipilih (lihat Gambar 9.5). Nomor skenario yang diinginkan dimasukkan di Sel A6, dan nilai referensi ditautkan ke dalam kalkulasi model berikutnya (Baris 10 ditautkan ke Baris 6). Perhatikan bahwa pada prinsipnya kalkulasi di Baris

6 dapat ditempatkan langsung di Baris 10.

Namun, untuk model besar, pendekatan seperti itu berarti bahwa fungsi CHOOSE akan merujuk ke data yang secara fisik terpisah dari inputnya dengan cara yang lebih signifikan, dan dengan demikian menjadi kurang transparan dan lebih rentan terhadap kesalahan. Perhatikan juga bahwa fungsi CHOOSE memerlukan referensi eksplisit dari data setiap skenario individual. Jika data skenario disajikan dalam rentang yang berdekatan (seperti pada contoh di atas), akan lebih efisien (dalam hal pembuatan model dan fleksibilitas untuk menambahkan skenario baru) untuk menggunakan fungsi INDEX guna mencari nilai yang relevan.

(Alternatif lain adalah fungsi OFFSET, meskipun, sebagai fungsi Volatile, fungsi ini kurang efisien secara komputasi.) gambar 9.6 menunjukkan bagian pemilihan skenario dari contoh di atas, yang diimplementasikan menggunakan masing-masing fungsi CHOOSE, INDEX, dan OFFSET (lihat Gambar 9.6).

	A	B	C	D	E	F	G	H	I	J
1										
2		Revenues Scenarios	2016	2017	2018	2019	2020	2021	2022	
3		Low			367	407	439	483	520	
4		Base			431	452	488	537	577	
5		High			474	497	537	591	635	
6										
7	3	To Use: With CHOOSE			474	497	537	591	635	=CHOOSE(\$A7,3,4,5)
8		To Use: With INDEX			474	497	537	591	635	=INDEX(B:5,\$A7,1)
9		To Use: With OFFSET			474	497	537	591	635	=OFFSET(2,\$A7,0)

**Gambar 9.6 Penggunaan Fungsi CHOOSE, INDEX, Dan OFFSET Untuk Memilih Data Skenario**

Perhatikan bahwa fungsi CHOOSE secara eksplisit merujuk ke sel yang terpisah, fungsi INDEX secara eksplisit merujuk ke rentang yang berdekatan, dan fungsi OFFSET secara implisit merujuk ke rentang yang berdekatan (dengan memilih nilai yang diimbangi dari tajuk tanggal dengan jumlah baris yang sama dengan nomor skenario).

Dengan demikian, penambahan skenario baru mungkin lebih mudah dicapai jika fungsi INDEX atau OFFSET digunakan. Gambar 9.7 menunjukkan perluasan metode pemilihan skenario di atas, yang di dalamnya terdapat skenario untuk biaya dan pendapatan. Pada prinsipnya (misalnya jika tanda dolar diterapkan dengan tepat), rumus pemilihan skenario tersebut dapat disalin untuk setiap variabel yang diinginkan. Dalam kasus praktis, data skenario sering kali dapat dihasilkan dengan salah satu dari dua cara:

- Dengan mencantumkan terlebih dahulu semua skenario pendapatan, lalu memperoleh skenario biaya terkait.
- Dengan mengerjakan skenario per skenario (misalnya, pertama skenario rendah, lalu skenario dasar, lalu skenario tinggi), menentukan data pendapatan dan biaya untuk masing-masing skenario.

	A	B	C	D	E	F	G	H	I	J
1										
2		Revenues Scenarios	2016	2017	2018	2019	2020	2021	2022	
3		Low			387	407	439	483	520	
4		Base			431	452	488	537	577	
5		High			474	497	537	591	635	
6										
7	3	To Use: With CHOOSE			474	497	537	591	635	=CHOOSE(\$A7,3,4,5)
8		To Use: With INDEX			474	497	537	591	635	=INDEX(3:5,\$A7,1)
9		To Use: With OFFSET			474	497	537	591	635	=OFFSET(2,\$A7,0)
10										
11										
12		Cost Scenarios	2016	2017	2018	2019	2020	2021	2022	
13		Low			354	364	430	480	428	
14		Base			395	387	488	476	563	
15		High			385	482	505	511	527	
16										
17	3	To Use: With CHOOSE			385	482	505	511	527	=CHOOSE(\$A17,113,114,115)
18		To Use: With INDEX			385	482	505	511	527	=INDEX(113:115,\$A17,1)
19		To Use: With OFFSET			385	482	505	511	527	=OFFSET(112,\$A17,0)
20										
21										

Gambar 9.7 Penggunaan Skenario Untuk Beberapa Variabel Model

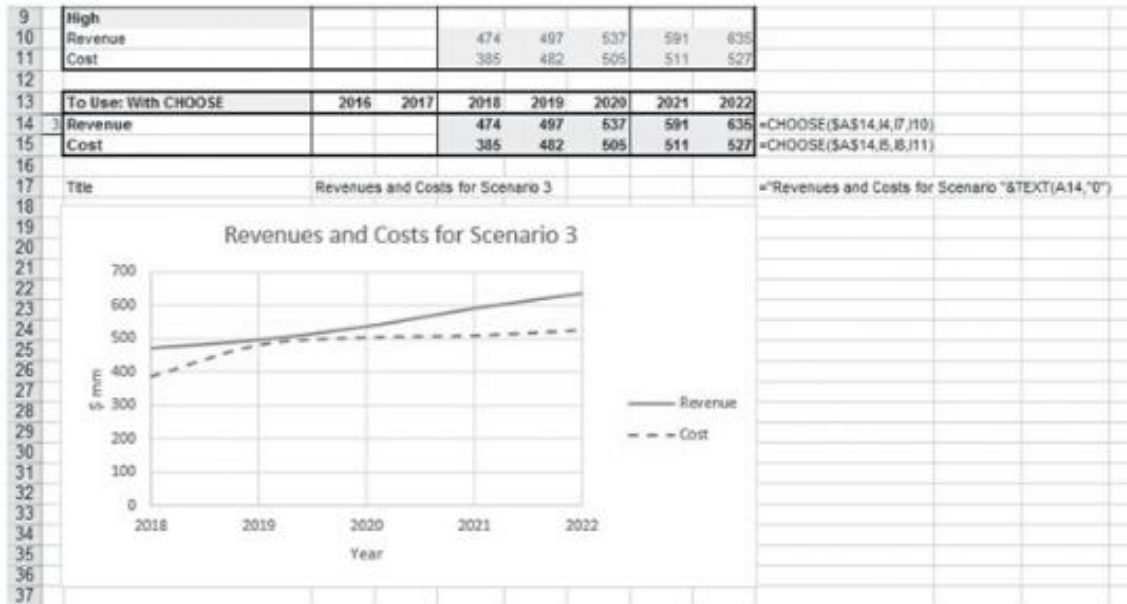
Gambar 9.8 menunjukkan contoh di mana struktur data skenario mengikuti pendekatan kedua (sedangkan Gambar 9.7 mengikuti pendekatan pertama). Dalam kasus seperti itu, penggunaan fungsi CHOOSE sering kali merupakan cara paling sederhana dan paling transparan untuk mencapai proses pemilihan skenario.

#### Contoh: Bagan yang Dapat Menggunakan Beberapa Sumber Data atau Fleksibel

Seseorang juga dapat menggunakan fungsi CHOOSE, INDEX, atau OFFSET untuk memilih data relevan yang kemudian ditautkan ke bagan, dan fungsi TEXT (atau ADDRESS atau lainnya) dapat digunakan untuk membuat label pembaruan untuk elemen bagan tertentu.

	A	B	C	D	E	F	G	H	I	J
1										
2		Scenarios	2016	2017	2018	2019	2020	2021	2022	
3		Low								
4		Revenue			387	407	439	483	520	
5		Cost			354	364	430	480	428	
6		Base								
7		Revenue			431	452	488	537	577	
8		Cost			395	387	488	476	563	
9		High								
10		Revenue			474	497	537	591	635	
11		Cost			385	482	505	511	527	
12										
13		To Use: With CHOOSE	2016	2017	2018	2019	2020	2021	2022	
14	3	Revenue			474	497	537	591	635	=CHOOSE(\$A\$14,4,7,10)
15		Cost			385	482	505	511	527	=CHOOSE(\$A\$14,5,8,11)
16										

Gambar 9.8 Penggunaan Skenario Untuk Set Data Yang Tidak Bersebelahan



**Gambar 9.9 Pendekatan Skenario Untuk Membuat Grafik**

Gambar 9.9 menunjukkan contoh di mana data yang masuk ke grafik, dan judul grafik, diperbarui secara otomatis saat nomor skenario diubah (lihat Gambar 9.9).

**Contoh: Membalikkan dan Mentransposisi Data Menggunakan INDEX atau OFFSET**

Ada banyak situasi di mana seseorang mungkin perlu membalikkan atau mentransposisi kumpulan data. Ini termasuk kasus di mana data deret waktu telah diimpor dari sumber eksternal, dalam perhitungan "segitiga" yang muncul dalam depresiasi dan konteks serupa lainnya (lihat Bab 2), atau saat menggunakan hasil fungsi array LINESST dalam regresi berganda untuk membuat prediksi (karena koefisien yang dikembalikan oleh fungsi tersebut berada dalam urutan terbalik dengan kumpulan data; lihat Bab 5).

Fungsi INDEX dan OFFSET dapat digunakan untuk membuat tautan dinamis antara kumpulan data asli (atau perhitungan) dan yang dibalik atau ditransposisi. Gambar 9.10 menunjukkan sebuah contoh. Set data asli ditampilkan di Kolom B dan C, dengan Kolom E yang dibuat untuk menyediakan referensi indeksasi. Fungsi OFFSET menggunakan indeksasi ini dalam artian subtraktif; semakin jauh ke bawah baris rumus disalin, semakin sedikit hasil yang diimbangi dari data asli (sehingga menciptakan efek pembalikan).

Hasil serupa dapat dicapai dengan menggunakan INDEX sebagai ganti OFFSET; sekali lagi, jika ada pilihan antara kedua fungsi tersebut, umumnya INDEX cenderung lebih disukai, karena meskipun OFFSET mungkin memiliki nama yang lebih menarik atau tampak transparan, fakta bahwa itu adalah fungsi Volatile mengurangi efektivitas komputasi, yang dapat menjadi sangat penting dengan set data dan model yang lebih besar. (Perhatikan bahwa untuk kesederhanaan penyajian prinsip-prinsip inti, contoh-contoh ditampilkan hanya dengan set data kecil, tetapi dalam banyak kasus praktis, jumlah item tentu saja akan jauh lebih besar.)

	A	B	C	D	E	F	G	H	I
1									
2		Count	15						
3							Using OFFSET		
4		Date	Adj Close				Date	Adj Close	
5		30/10/2008	4,869.30		1		10/10/2008	4,544.31	=OFFSET(C\$5,C\$2-E5,0)
6		29/10/2008	4,808.69		2		13/10/2008	5,062.45	=OFFSET(C\$5,C\$2-E6,0)
7		28/10/2008	4,823.45		3		14/10/2008	5,199.19	=OFFSET(C\$5,C\$2-E7,0)
8		27/10/2008	4,334.64		4		15/10/2008	4,861.63	=OFFSET(C\$5,C\$2-E8,0)
9		24/10/2008	4,295.67		5		16/10/2008	4,622.81	=OFFSET(C\$5,C\$2-E9,0)
10		23/10/2008	4,519.70		6		17/10/2008	4,781.33	=OFFSET(C\$5,C\$2-E10,0)
11		22/10/2008	4,571.07		7		20/10/2008	4,835.01	=OFFSET(C\$5,C\$2-E11,0)
12		21/10/2008	4,784.41		8		21/10/2008	4,784.41	=OFFSET(C\$5,C\$2-E12,0)
13		20/10/2008	4,835.01		9		22/10/2008	4,571.07	=OFFSET(C\$5,C\$2-E13,0)
14		17/10/2008	4,781.33		10		23/10/2008	4,519.70	=OFFSET(C\$5,C\$2-E14,0)
15		16/10/2008	4,622.81		11		24/10/2008	4,295.67	=OFFSET(C\$5,C\$2-E15,0)
16		15/10/2008	4,861.63		12		27/10/2008	4,334.64	=OFFSET(C\$5,C\$2-E16,0)
17		14/10/2008	5,199.19		13		28/10/2008	4,823.45	=OFFSET(C\$5,C\$2-E17,0)
18		13/10/2008	5,062.45		14		29/10/2008	4,808.69	=OFFSET(C\$5,C\$2-E18,0)
19		10/10/2008	4,544.31		15		30/10/2008	4,869.30	=OFFSET(C\$5,C\$2-E19,0)
20									

**Gambar 9.10 Penggunaan OFFSET Untuk Membalikkan Data Deret Waktu**

	A	B	C	D	E	F	G	H	I
1									
2		Count	15						
3							Using INDEX		
4		Date	Adj Close				Date	Adj Close	
5		30/10/2008	4,869.30		1		10/10/2008	4,544.31	=INDEX(C\$5:C\$19,C\$2-E5+1,1)
6		29/10/2008	4,808.69		2		13/10/2008	5,062.45	=INDEX(C\$5:C\$19,C\$2-E6+1,1)
7		28/10/2008	4,823.45		3		14/10/2008	5,199.19	=INDEX(C\$5:C\$19,C\$2-E7+1,1)
8		27/10/2008	4,334.64		4		15/10/2008	4,861.63	=INDEX(C\$5:C\$19,C\$2-E8+1,1)
9		24/10/2008	4,295.67		5		16/10/2008	4,622.81	=INDEX(C\$5:C\$19,C\$2-E9+1,1)
10		23/10/2008	4,519.70		6		17/10/2008	4,781.33	=INDEX(C\$5:C\$19,C\$2-E10+1,1)
11		22/10/2008	4,571.07		7		20/10/2008	4,835.01	=INDEX(C\$5:C\$19,C\$2-E11+1,1)
12		21/10/2008	4,784.41		8		21/10/2008	4,784.41	=INDEX(C\$5:C\$19,C\$2-E12+1,1)
13		20/10/2008	4,835.01		9		22/10/2008	4,571.07	=INDEX(C\$5:C\$19,C\$2-E13+1,1)
14		17/10/2008	4,781.33		10		23/10/2008	4,519.70	=INDEX(C\$5:C\$19,C\$2-E14+1,1)
15		16/10/2008	4,622.81		11		24/10/2008	4,295.67	=INDEX(C\$5:C\$19,C\$2-E15+1,1)
16		15/10/2008	4,861.63		12		27/10/2008	4,334.64	=INDEX(C\$5:C\$19,C\$2-E16+1,1)
17		14/10/2008	5,199.19		13		28/10/2008	4,823.45	=INDEX(C\$5:C\$19,C\$2-E17+1,1)
18		13/10/2008	5,062.45		14		29/10/2008	4,808.69	=INDEX(C\$5:C\$19,C\$2-E18+1,1)
19		10/10/2008	4,544.31		15		30/10/2008	4,869.30	=INDEX(C\$5:C\$19,C\$2-E19+1,1)
20									

**Gambar 9.11 Penggunaan INDEX untuk Membalikkan Data Deret Waktu**

Gambar 9.11 menunjukkan implementasi fungsi INDEX untuk contoh yang sama. Perhatikan bahwa nomor indeksasi yang dibuat di Kolom E tidak perlu ditempatkan secara eksplisit di lembar kerja (hal ini dilakukan di atas untuk memaksimalkan transparansi penyajian).



	A	B	C	D	E	F	G	H	I
1									
2		Count	15						
3									
4		Date	Adj Close				Using INDEX		
5		30/10/2008	4,869.30				Date	Adj Close	
6		29/10/2008	4,808.69				10/10/2008	4,544.31	=INDEX(C\$5:C\$19,C\$2-(ROW(C5)-ROW(C\$5)),1)
7		28/10/2008	4,823.45				13/10/2008	5,062.45	=INDEX(C\$5:C\$19,C\$2-(ROW(C6)-ROW(C\$5)),1)
8		27/10/2008	4,334.64				14/10/2008	5,199.19	=INDEX(C\$5:C\$19,C\$2-(ROW(C7)-ROW(C\$5)),1)
9		24/10/2008	4,295.67				15/10/2008	4,861.63	=INDEX(C\$5:C\$19,C\$2-(ROW(C8)-ROW(C\$5)),1)
10		23/10/2008	4,519.70				16/10/2008	4,622.81	=INDEX(C\$5:C\$19,C\$2-(ROW(C9)-ROW(C\$5)),1)
11		22/10/2008	4,571.07				17/10/2008	4,781.33	=INDEX(C\$5:C\$19,C\$2-(ROW(C10)-ROW(C\$5)),1)
12		21/10/2008	4,784.41				20/10/2008	4,835.01	=INDEX(C\$5:C\$19,C\$2-(ROW(C11)-ROW(C\$5)),1)
13		20/10/2008	4,835.01				21/10/2008	4,784.41	=INDEX(C\$5:C\$19,C\$2-(ROW(C12)-ROW(C\$5)),1)
14		17/10/2008	4,781.33				22/10/2008	4,571.07	=INDEX(C\$5:C\$19,C\$2-(ROW(C13)-ROW(C\$5)),1)
15		16/10/2008	4,622.81				23/10/2008	4,519.70	=INDEX(C\$5:C\$19,C\$2-(ROW(C14)-ROW(C\$5)),1)
16		15/10/2008	4,861.63				24/10/2008	4,295.67	=INDEX(C\$5:C\$19,C\$2-(ROW(C15)-ROW(C\$5)),1)
17		14/10/2008	5,199.19				27/10/2008	4,334.64	=INDEX(C\$5:C\$19,C\$2-(ROW(C16)-ROW(C\$5)),1)
18		13/10/2008	5,062.45				28/10/2008	4,823.45	=INDEX(C\$5:C\$19,C\$2-(ROW(C17)-ROW(C\$5)),1)
19		10/10/2008	4,544.31				29/10/2008	4,808.69	=INDEX(C\$5:C\$19,C\$2-(ROW(C18)-ROW(C\$5)),1)
20							30/10/2008	4,869.30	=INDEX(C\$5:C\$19,C\$2-(ROW(C19)-ROW(C\$5)),1)

**Gambar 9.12 Menggunakan ROW untuk Membuat Bidang Pengindeksan Tertanam**

Sebagai gantinya, fungsi ROW dapat disematkan dalam rumus sehingga pengindeksan dihitung untuk setiap elemen (fungsi ROWS juga dapat digunakan sebagai pengganti fungsi COUNT di Sel C2).

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
1																	
2										1	2	3	4	5	6	7	8
3										100	105	110	115	120	125	130	135
4																	
5			INDEX				OFFSET										
6	1		100	=INDEX(\$J\$3:\$Q\$3,1,B6)			100	=OFFSET(\$J\$3,0,B6-1)									
7	2		105	=INDEX(\$J\$3:\$Q\$3,1,B7)			105	=OFFSET(\$J\$3,0,B7-1)									
8	3		110	=INDEX(\$J\$3:\$Q\$3,1,B8)			110	=OFFSET(\$J\$3,0,B8-1)									
9	4		115	=INDEX(\$J\$3:\$Q\$3,1,B9)			115	=OFFSET(\$J\$3,0,B9-1)									
10	5		120	=INDEX(\$J\$3:\$Q\$3,1,B10)			120	=OFFSET(\$J\$3,0,B10-1)									
11	6		125	=INDEX(\$J\$3:\$Q\$3,1,B11)			125	=OFFSET(\$J\$3,0,B11-1)									
12	7		130	=INDEX(\$J\$3:\$Q\$3,1,B12)			130	=OFFSET(\$J\$3,0,B12-1)									
13	8		135	=INDEX(\$J\$3:\$Q\$3,1,B13)			135	=OFFSET(\$J\$3,0,B13-1)									

**Gambar 9.13 Menggunakan Fungsi Pencarian untuk Mengubah Posisi Data**

Gambar 9.12 menunjukkan implementasi fungsi INDEX untuk contoh yang sama. Mengenai transposisi data, sebagai alternatif penggunaan fungsi array TRANSPOSE (Bab 18), seseorang dapat menggunakan fungsi INDEX atau OFFSET. Ini mirip dengan yang di atas, kecuali bahwa (untuk mentransposisi data bentuk baris ke bentuk kolom), rumus pencarian ditulis sehingga saat fungsi disalin ke bawah, proses pencarian bergerak melintasi baris. Gambar 9.13 menunjukkan implementasi ini (sekali lagi, menggunakan pendekatan manual eksplisit untuk pengindeksan di Kolom B, sedangkan fungsi ROW dan COLUMNS atau COUNT dapat digunakan untuk membuat pengindeksan, baik di Kolom B atau tertanam dalam fungsi).

	A	B	C	D	E	F	G	H	I	J
1										
2		Period Number		1	2	3	4	5	6	
3		Volume: without shift		100	110	115	120	125	125	
4										
5		Methods to shift in time through a single delay amount i.e. spread across two future periods								
6										
7				1	2	3	4	5	6	
8		Delay is anything less than 1	0.3	70	107	114	119	124	125	=H3*\$C8+I3*(1-\$C8)
9		Delay in periods: whole numbers	2	0	0	100	110	115	120	=IF(AND(I\$7-\$C9>=1,I\$7-\$C9<=10),INDEX(\$D\$3:\$I\$3,1,I\$7-\$C9),0)
10		Delay in periods: whole numbers	2	0	0	100	110	115	120	=IF(I2<=\$C10,0,OFFSET(I3,0,-\$C10))
11		Delay is any number	1.6	0	40	104	112	117	122	=IF(I\$2<=(ROUNDDOWN(\$C11,0)+1),0,OFFSET(I\$3,0,-(ROUNDDOWN

Gambar 9.14 Berbagai Metode untuk Menggeser Arus Kas dari Waktu ke Waktu

### Contoh: Menggeser Arus Kas atau Item Lain dari Waktu ke Waktu

Ada banyak keadaan di mana seseorang mungkin ingin memodelkan dampak penundaan pada produksi, arus kas, atau pada kuantitas lainnya. Dalam kasus seperti itu, granularitas sumbu waktu dapat berdampak pada kompleksitas rumus yang diperlukan. Berikut ini, kami membahas kasus-kasus berikut:

- Penundaan yang panjangnya paling banyak satu periode model (misalnya setengah periode model).
- Penundaan yang panjangnya merupakan sejumlah periode model.
- Penundaan yang panjangnya dapat berupa jumlah positif apa pun.

Gambar 9.14 menunjukkan contoh yang pertama dapat dicapai dengan rumus pembobotan sederhana, yang kedua dan ketiga dengan menggunakan fungsi INDEX atau OFFSET. Fungsi Excel sederhana lainnya (misalnya IF, AND pada kasus kedua dan ROUNDDOWN pada kasus ketiga) juga diperlukan; pembaca dapat melihat rumus lengkap dalam berkas; untuk sel I11, rumus yang diterapkan adalah:

$$IF(I\$2 < (ROUNDDOWN(\$C11,0) + 1), 0, OFFSET(I\$3, 0, - (ROUNDDOWN(\$C11,0) + 1)) * (\$C11 - INT(\$C11)) + OFFSET(I\$3, 0, -ROUNDDOWN(\$C11,0)) * (1 - (\$C11 - INT(\$C11))))$$

Karena kompleksitasnya, sering kali lebih baik menerapkannya sebagai fungsi yang ditentukan pengguna dalam VBA.

### Contoh: Jadwal Penyusutan dengan Perhitungan Segitiga

Gambar 9.15 menunjukkan contoh di mana fungsi INDEX digunakan dalam perhitungan tipe segitiga untuk mentransposisi dan menggeser waktu data. SUMPRODUCT digunakan untuk menghitung penyusutan periodik (lihat Gambar 25.15). Perhatikan bahwa pernyataan IF dan AND digunakan untuk memastikan validitas periode waktu, dan juga bahwa seseorang tidak dapat menggunakan fungsi IFERROR secara kuat dalam konteks ini (karena fungsi INDEX terkadang dapat mengembalikan elemen pertama dari suatu rentang ketika angka indeksasi ditetapkan ke nol).

	A	B	C	D	E	F	G	H	I	J	K	L
1												
2		Years		2015	2016	2017	2018	2019	2020			
3		Capex	575	100	105	120	80	50	120			
4												
5		Depreciation Schedule		1	2	3	4	5				
6		Percentage	100%	40%	30%	20%	10%	0%				
7												
8		Years	Capex	2015	2016	2017	2018	2019	2020	2021	2022	2023
9	1	2015	100	40%	30%	20%	10%					
10	2	2016	105		40%	30%	20%	10%				
11	3	2017	120			40%	30%	20%	10%			
12	4	2018	80				40%	30%	20%	10%		
13	5	2019	50					40%	30%	20%	10%	
14	6	2020	120						40%	30%	20%	10%
15												
16		Depreciation		40	72	100	99	79	91	54	29	12
17												

Gambar 9.15 Menggunakan INDEKS dalam Perhitungan Penyusutan Tipe Segitiga

#### 9.4 APLIKASI PRAKTIS: MENGGABUNGKAN PROSES PENCOCOKAN DAN REFERENSI

**Contoh: Menemukan Periode di Mana Suatu Kondisi Terpenuhi Menggunakan PENCOCOKAN**

Sering kali penting untuk dapat mengidentifikasi periode waktu atau posisi dalam suatu model di mana suatu kondisi terpenuhi untuk pertama kalinya, misalnya:

- Pertama kali pendapatan suatu produk lebih tinggi daripada pendapatan produk lain.
- Pertama kali pendapatan mencapai titik impas, atau ketika pendapatan bisnis yang menurun turun di bawah titik tersebut.
- Tanggal di mana ladang minyak yang berproduksi harus ditinggalkan karena produksi menurun seiring waktu karena ladang tersebut habis, dan NPV masa depan akan negatif untuk pertama kalinya.
- Pertama kali kondisi terpenuhi yang memungkinkan pinjaman untuk dibiayai kembali pada tingkat yang lebih rendah, seperti ketika kondisi perjanjian tertentu terpenuhi.

Fungsi MATCH sangat ampuh dalam konteks seperti itu. Sintaks umumnya adalah sebagai berikut:

*MATCH (ValueToFind, RowOrColRangeToLookIn, [OptionalMatchType])*

Biasanya, akan lebih mudah untuk memastikan bahwa rentang pencarian berisi nilai variabel "bendera", yaitu yang menggunakan pernyataan IF untuk mengembalikan TRUE (atau 1) ketika kondisi terpenuhi dan FALSE (atau 0) jika tidak. Selain itu, biasanya penting untuk menggunakan argumen terakhir opsional dari fungsi tersebut di mana tipe pencocokan nol mengembalikan pencocokan persis pertama; jika ini dihilangkan, data dalam rentang pencarian harus dalam urutan menaik dan nilai terbesar yang kurang dari atau sama dengan nilai pencarian dikembalikan (sesuatu yang sering diabaikan, sehingga menghasilkan nilai yang

tidak konsisten).

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
1																
2						2018	2019	2020	2021	2022	2023	2024	2025	2026	2027	
3		Product 1	100			103	106	109	113	116	119	123	127	130	134	=N3*(1+O4)
4		Growth (% p.a.)	3.0%			3.0%	3.0%	3.0%	3.0%	3.0%	3.0%	3.0%	3.0%	3.0%	3.0%	=SC4
5		Product 2	70			76	82	88	95	103	111	120	130	140	151	=N5*(1+O6)
6		Growth (% p.a.)	8.0%			8.0%	8.0%	8.0%	8.0%	8.0%	8.0%	8.0%	8.0%	8.0%	8.0%	=SC8
7		Test Prod2>=Prod1		=MATCH(1,F7:O7,0)		0	0	0	0	0	0	0	1	1	1	=F(O5)>=O3,1,0)
8																
9		Default	10	=MATCH(1,F7:O7)												
10																
11		Year of Match	2025	=INDEX(F2:O2,1,C7)												

**Gambar 9.16 Menggunakan MATCH untuk Menemukan Waktu di Mana Pendapatan Mencapai Target**

Gambar 9.16 menunjukkan contoh yang diterapkan untuk mencari tahu kapan pendapatan dari satu produk akan melampaui pendapatan produk lain, menurut perkiraan pertumbuhan pendapatan perusahaan (lihat Gambar 9.16). Perhatikan bahwa rentang F7:O7 berisi kalkulasi variabel flag, dan Sel C7 berisi fungsi MATCH. Perhatikan juga bahwa, jika parameter opsional (terakhir) dibiarkan kosong, maka fungsi akan mengembalikan posisi kecocokan terakhir, bukan yang pertama (Sel C9).

Terakhir, perlu ditegaskan kembali bahwa fungsi mengembalikan posisi (relatif) (atau nomor indeks) di mana kondisi ini pertama kali terpenuhi, sehingga nilai di Sel C7 bukanlah nomor tahun (di Baris 2) tetapi justru menunjukkan bahwa kecocokan terjadi di posisi kedelapan dari kumpulan data; nilai tahun sebenarnya dapat dicari menggunakan fungsi INDEX (Sel C11).

### Contoh: Menemukan Data Skenario yang Tidak Bersebelahan Menggunakan Kunci Pencocokan

Sebelumnya di bab ini, kita menggunakan fungsi CHOOSE untuk mencari data untuk skenario dalam set data yang tidak bersebelahan, serta menggunakan INDEX atau OFFSET saat data bersebelahan. Umumnya, pendekatan ini adalah yang paling dapat diandalkan. Meskipun demikian, seseorang mungkin menemukan model yang menggunakan pendekatan yang lebih kompleks (atau tidak ideal), dan ada baiknya untuk menyadari kemungkinan tersebut.

	A	B	C	D	E	F	G	H	I	J	K
1											
2		Scenarios	KEY			2018	2019	2020	2021	2022	
3		Low									
4		Revenue	LowRevenue			307	407	438	483	520	
5		Cost	LowCost			354	364	430	480	428	
6		Base									
7		Revenue	BaseRevenue			431	452	488	537	577	
8		Cost	BaseCost			395	367	488	476	563	
9		High									
10		Revenue	HighRevenue			474	497	537	591	638	
11		Cost	HighCost			355	482	505	511	527	
12											
13		Scenarios to Use	Position			2018	2019	2020	2021	2022	
14		BaseRevenue		=MATCH(B14,5C5:5C511,0)		431	452	488	537	577	=INDEX(55:511,5C14,1)
15		BaseCost		=MATCH(B15,5C5:5C511,0)		395	367	488	476	563	=INDEX(55:511,5C15,1)

**Gambar 9.17 Menggunakan MATCH untuk Memilih Data dari Skenario yang Tidak Bersebelahan**

Gambar 9.17 menunjukkan contoh di mana setiap skenario didefinisikan secara unik oleh kunci teks dan ini digunakan untuk memilih data skenario yang sesuai. Fungsi MATCH menemukan posisi skenario dalam set data, dan fungsi INDEX digunakan untuk mencari data ini (lihat Gambar 9.17).

### Contoh: Membuat dan Menemukan Kolom Teks atau Kunci yang Cocok

Saat memanipulasi set data, termasuk menggabungkannya atau menggunakan satu di dalam yang lain, seseorang mungkin perlu menggunakan beberapa teknik secara bersamaan. Misalnya, fungsi Teks dapat digunakan untuk membuat kunci unik (seperti dalam Bab 8) yang dapat dicocokkan bersama. Gambar 9.18 menunjukkan sebuah contoh (lihat Gambar 9.18). Untuk set data pertama, fungsi Teks digunakan untuk memanipulasi data nilai tukar yang mendasarinya (dalam rentang B4:C15) untuk membuat kunci unik untuk setiap item (G4:G15), dan proses serupa digunakan dalam basis data utama. Akhirnya, posisi kunci yang cocok diidentifikasi. (Langkah logis terakhir dalam analisis, di mana nilai tukar yang berlaku dicari, dibahas dalam contoh berikut.)

	A	B	C	D	E	F	G	H	I
1									
2		EXCHANGE RATE TABLE							
3		XrateK	Orig Key	Search for Q	LEFT	RIGHT	Key to Match		
4		1.50	DollarQ_1	7 DollarQ	1	DollarQ_1			
5		1.50	DollarQ_2	7 DollarQ	2	DollarQ_2			
6		1.55	DollarQ_3	7 DollarQ	3	DollarQ_3			
7		1.62	DollarQ_4	7 DollarQ	4	DollarQ_4			
8		1.20	EuroQ_1	5 EuroQ	1	EuroQ_1			
9		1.16	EuroQ_2	5 EuroQ	2	EuroQ_2			
10		1.22	EuroQ_3	5 EuroQ	3	EuroQ_3			
11		1.25	EuroQ_4	5 EuroQ	4	EuroQ_4			
12		1.00	PoundQ_1	6 PoundQ	1	PoundQ_1			
13		1.00	PoundQ_2	6 PoundQ	2	PoundQ_2			
14		1.00	PoundQ_3	6 PoundQ	3	PoundQ_3			
15		1.00	PoundQ_4	6 PoundQ	4	PoundQ_4			
16									
17		MAIN ("LARGE") CUSTOMER DATABASE							
18		Location	Customer no.	Quarter	Amount (bcs Currency	Key to Match	Position of Matching Item		
19		Germany	C314	1	32,143 Euro	EuroQ_1	5	=MATCH(G19,\$G\$4:\$G\$15,0)	
20		London	C159	3	17,203 Pound	PoundQ_3	11	=MATCH(G20,\$G\$4:\$G\$15,0)	
21		France	C265	2	16,993 Euro	EuroQ_2	6	=MATCH(G21,\$G\$4:\$G\$15,0)	
22		US	C358	2	78,888 Dollar	DollarQ_2	2	=MATCH(G22,\$G\$4:\$G\$15,0)	
23		London	C979	3	43,957 Pound	PoundQ_3	11	=MATCH(G23,\$G\$4:\$G\$15,0)	
24		US	C323	4	25,898 Dollar	DollarQ_4	4	=MATCH(G24,\$G\$4:\$G\$15,0)	
25		London	C486	1	26,858 Pound	PoundQ_1	9	=MATCH(G25,\$G\$4:\$G\$15,0)	
26		London	C729	2	63,652 Dollar	DollarQ_2	2	=MATCH(G26,\$G\$4:\$G\$15,0)	
27		US	C266	3	28,180 Dollar	DollarQ_3	3	=MATCH(G27,\$G\$4:\$G\$15,0)	
28		Germany	C357	4	47,153 Euro	EuroQ_4	8	=MATCH(G28,\$G\$4:\$G\$15,0)	
29		France	C989	1	51,737 Euro	EuroQ_1	5	=MATCH(G29,\$G\$4:\$G\$15,0)	
30		London	C323	2	36,332 Pound	PoundQ_2	10	=MATCH(G30,\$G\$4:\$G\$15,0)	
31		Germany	C486	3	33,363 Euro	EuroQ_3	7	=MATCH(G31,\$G\$4:\$G\$15,0)	
32		Germany	C729	1	17,635 Euro	EuroQ_1	5	=MATCH(G32,\$G\$4:\$G\$15,0)	
33									

**Gambar 9.18 Menggunakan Fungsi Teks dan MATCH untuk Menemukan Data dalam Basis Data Mata Uang**

### Contoh: Menggabungkan INDEX dengan MATCH

Seperti disebutkan di atas, karena fungsi MATCH hanya menemukan posisi relatif suatu item (yaitu mengembalikan nomor indeksasi); fungsi ini umumnya perlu digunakan sebagai input untuk proses pencarian lebih lanjut. Berkas Gambar 9.19 menunjukkan langkah terakhir dalam analisis untuk contoh sebelumnya: fungsi INDEX digunakan untuk mencari nilai tukar yang berlaku dan perhitungan dilakukan untuk mengetahui angka total dalam Poundsterling (£) (lihat Gambar 9.19).

	H	I	J	K	L	M	N	O
16								
17								
18		Position of Matching Item		Xrate:€			Amount (€)	
19		5	=MATCH(G19,\$G\$4:\$G\$15,0)	1.20	=INDEX(\$B\$4:\$B\$15,H19,1)		26786	=E19/K19
20		11	=MATCH(G20,\$G\$4:\$G\$15,0)	1.00	=INDEX(\$B\$4:\$B\$15,H20,1)		17203	=E20/K20
21		6	=MATCH(G21,\$G\$4:\$G\$15,0)	1.16	=INDEX(\$B\$4:\$B\$15,H21,1)		14649	=E21/K21
22		2	=MATCH(G22,\$G\$4:\$G\$15,0)	1.58	=INDEX(\$B\$4:\$B\$15,H22,1)		49929	=E22/K22
23		11	=MATCH(G23,\$G\$4:\$G\$15,0)	1.00	=INDEX(\$B\$4:\$B\$15,H23,1)		43957	=E23/K23
24		4	=MATCH(G24,\$G\$4:\$G\$15,0)	1.82	=INDEX(\$B\$4:\$B\$15,H24,1)		15987	=E24/K24
25		9	=MATCH(G25,\$G\$4:\$G\$15,0)	1.00	=INDEX(\$B\$4:\$B\$15,H25,1)		26856	=E25/K25
26		2	=MATCH(G26,\$G\$4:\$G\$15,0)	1.58	=INDEX(\$B\$4:\$B\$15,H26,1)		40286	=E26/K26
27		3	=MATCH(G27,\$G\$4:\$G\$15,0)	1.55	=INDEX(\$B\$4:\$B\$15,H27,1)		18831	=E27/K27
28		8	=MATCH(G28,\$G\$4:\$G\$15,0)	1.25	=INDEX(\$B\$4:\$B\$15,H28,1)		37723	=E28/K28
29		5	=MATCH(G29,\$G\$4:\$G\$15,0)	1.20	=INDEX(\$B\$4:\$B\$15,H29,1)		43114	=E29/K29
30		10	=MATCH(G30,\$G\$4:\$G\$15,0)	1.00	=INDEX(\$B\$4:\$B\$15,H30,1)		36332	=E30/K30
31		7	=MATCH(G31,\$G\$4:\$G\$15,0)	1.22	=INDEX(\$B\$4:\$B\$15,H31,1)		27347	=E31/K31
32		5	=MATCH(G32,\$G\$4:\$G\$15,0)	1.20	=INDEX(\$B\$4:\$B\$15,H32,1)		14696	=E32/K32
33							413696	=SUM(N19:N32)
34								

Gambar 25.19 Menggabungkan Proses INDEX dan MATCH

### Contoh: Membandingkan INDEX-MATCH dengan V- dan HLOOKUP

Contoh-contoh berikut bertujuan untuk menyoroti bahwa penggunaan kombinasi INDEX-MATCH hampir selalu lebih disukai daripada penggunaan fungsi V- atau HLOOKUP. Hal ini karena alasan fleksibilitas, ketahanan, efisiensi komputasi, dan kemudahan audit model yang lebih unggul. Gambar 9.20 menunjukkan contoh penggunaan fungsi VLOOKUP untuk menyediakan data nilai tukar yang tepat (Gambar 25.20). Perhitungannya benar dan tampaknya menawarkan solusi yang memadai untuk situasi yang dihadapi.

	A	B	C	D	E	F	G	H	I
1									
2		Currency	Xrate:€						
3		Dollar	1.60						
4		Euro	1.20						
5		Pound	1.00						
6									
7		Location	Customer no.	Date	Amount (local currency)	Currency	Xrate:€		
8		Germany	C314	Jan	32,143	Euro	1.20	=VLOOKUP(F8,\$B\$3:\$C\$5,2)	
9		London	C159	Jan	17,203	Pound	1.00	=VLOOKUP(F9,\$B\$3:\$C\$5,2)	
10		France	C265	Feb	16,993	Euro	1.20	=VLOOKUP(F10,\$B\$3:\$C\$5,2)	
11		US	C358	Feb	78,888	Dollar	1.60	=VLOOKUP(F11,\$B\$3:\$C\$5,2)	
12		London	C979	Feb	43,957	Pound	1.00	=VLOOKUP(F12,\$B\$3:\$C\$5,2)	
13		US	C323	Mar	25,898	Dollar	1.60	=VLOOKUP(F13,\$B\$3:\$C\$5,2)	
14		London	C486	Mar	28,856	Pound	1.00	=VLOOKUP(F14,\$B\$3:\$C\$5,2)	
15		London	C729	Mar	63,652	Dollar	1.60	=VLOOKUP(F15,\$B\$3:\$C\$5,2)	
16		US	C266	April	29,188	Dollar	1.60	=VLOOKUP(F16,\$B\$3:\$C\$5,2)	
17		Germany	C357	April	47,153	Euro	1.20	=VLOOKUP(F17,\$B\$3:\$C\$5,2)	
18		France	C989	May	51,737	Euro	1.20	=VLOOKUP(F18,\$B\$3:\$C\$5,2)	
19		London	C323	June	36,332	Pound	1.00	=VLOOKUP(F19,\$B\$3:\$C\$5,2)	
20		Germany	C486	July	33,363	Euro	1.20	=VLOOKUP(F20,\$B\$3:\$C\$5,2)	
21		Germany	C729	August	17,635	Euro	1.20	=VLOOKUP(F21,\$B\$3:\$C\$5,2)	
22									

Gambar 9.20 Menggunakan VLOOKUP untuk Menemukan Nilai Tukar yang Relevan

Gambar 9.21 menyoroti bahwa fungsi VLOOKUP mengharuskan nilai yang akan dicocokkan ditempatkan di sebelah kiri kumpulan data. Hal ini membatasi fleksibilitas dalam hal reposisi kumpulan data (menyalin kumpulan data awal beberapa kali merupakan alternatif yang tidak efisien dan rawan kesalahan) (lihat Gambar 9.21). Gambar 9.22 menyoroti bahwa fungsi VLOOKUP awalnya menggunakan nomor kolom yang dikodekan secara keras (misalnya Kolom 2 dalam contoh ini) (lihat Gambar 9.22).

Hal ini dapat menyebabkan pemodel (atau pengguna lain) membuat kesalahan yang tidak disengaja dengan menambahkan kolom baru tanpa mengadaptasi rumus untuk menyesuaikan referensi kolom. Terutama dalam basis data yang lebih besar dan yang mungkin perlu diperbarui di beberapa titik, kesalahan seperti itu sering terjadi dan sering tidak teramati; kemungkinan menyadari kesalahan berkurang lebih jauh jika item data memiliki nilai yang sama (seperti kolom data gaji dalam periode yang berbeda), karena kesalahan tidak akan mudah diidentifikasi melalui nilai yang ditampilkan yang jelas salah.

	A	B	C	D	E	F	G	H	I
1									
2			Xrate:E	Currency					
3			1.60	Dollar					
4			1.20	Euro					
5			1.00	Pound					
6									
7									
8		Location	Customer no.	Date	Amount (local currency)	Currency	Xrate:E		
9		Germany	C314	Jan	32,143	Euro	#N/A	=VLOOKUP(F8,\$C\$3:\$D\$5,2)	
10		London	C159	Jan	17,203	Pound	#N/A	=VLOOKUP(F9,\$C\$3:\$D\$5,2)	
11		France	C265	Feb	16,993	Euro	#N/A	=VLOOKUP(F10,\$C\$3:\$D\$5,2)	
12		US	C358	Feb	78,888	Dollar	#N/A	=VLOOKUP(F11,\$C\$3:\$D\$5,2)	
13		London	C979	Feb	43,957	Pound	#N/A	=VLOOKUP(F12,\$C\$3:\$D\$5,2)	
14		US	C323	Mar	25,898	Dollar	#N/A	=VLOOKUP(F13,\$C\$3:\$D\$5,2)	
15		London	C486	Mar	26,856	Pound	#N/A	=VLOOKUP(F14,\$C\$3:\$D\$5,2)	
16		London	C729	Mar	63,652	Dollar	#N/A	=VLOOKUP(F15,\$C\$3:\$D\$5,2)	
17		US	C266	April	29,188	Dollar	#N/A	=VLOOKUP(F16,\$C\$3:\$D\$5,2)	
18		Germany	C357	April	47,153	Euro	#N/A	=VLOOKUP(F17,\$C\$3:\$D\$5,2)	
19		France	C989	May	51,737	Euro	#N/A	=VLOOKUP(F18,\$C\$3:\$D\$5,2)	
20		London	C323	June	36,332	Pound	#N/A	=VLOOKUP(F19,\$C\$3:\$D\$5,2)	
21		Germany	C486	July	33,363	Euro	#N/A	=VLOOKUP(F20,\$C\$3:\$D\$5,2)	
22		Germany	C729	August	17,635	Euro	#N/A	=VLOOKUP(F21,\$C\$3:\$D\$5,2)	

**Gambar 9.21 Keterbatasan Struktur Data Saat Menggunakan VLOOKUP**

	A	B	C	D	E	F	G	H	I	J
1										
2			Currency	Xrate:E						
3			Dollar	1.60						
4			Euro	1.20						
5			Pound	1.00						
6										
7										
8		Location	Customer no.	Date	Amount (local currency)	Currency	Xrate:E			
9		Germany	C314	Jan	32,143	Euro	0.00	=VLOOKUP(G8,\$B\$3:\$D\$5,2)		
10		London	C159	Jan	17,203	Pound	0.00	=VLOOKUP(G9,\$B\$3:\$D\$5,2)		
11		France	C265	Feb	16,993	Euro	0.00	=VLOOKUP(G10,\$B\$3:\$D\$5,2)		
12		US	C358	Feb	78,888	Dollar	0.00	=VLOOKUP(G11,\$B\$3:\$D\$5,2)		
13		London	C979	Feb	43,957	Pound	0.00	=VLOOKUP(G12,\$B\$3:\$D\$5,2)		
14		US	C323	Mar	25,898	Dollar	0.00	=VLOOKUP(G13,\$B\$3:\$D\$5,2)		
15		London	C486	Mar	26,856	Pound	0.00	=VLOOKUP(G14,\$B\$3:\$D\$5,2)		
16		London	C729	Mar	63,652	Dollar	0.00	=VLOOKUP(G15,\$B\$3:\$D\$5,2)		
17		US	C266	April	29,188	Dollar	0.00	=VLOOKUP(G16,\$B\$3:\$D\$5,2)		
18		Germany	C357	April	47,153	Euro	0.00	=VLOOKUP(G17,\$B\$3:\$D\$5,2)		
19		France	C989	May	51,737	Euro	0.00	=VLOOKUP(G18,\$B\$3:\$D\$5,2)		
20		London	C323	June	36,332	Pound	0.00	=VLOOKUP(G19,\$B\$3:\$D\$5,2)		
21		Germany	C486	July	33,363	Euro	0.00	=VLOOKUP(G20,\$B\$3:\$D\$5,2)		
22		Germany	C729	August	17,635	Euro	0.00	=VLOOKUP(G21,\$B\$3:\$D\$5,2)		

**Gambar 9.22 Sifat Nomor Kolom yang Dikodekan Secara Keras yang Rawan Kesalahan Saat Menggunakan VLOOKUP**

Gambar 9.23 menunjukkan bagaimana keterbatasan penambahan kolom di atas dengan cara yang kuat dapat diatasi dengan menggunakan fungsi MATCH untuk menentukan nomor kolom, sehingga tidak lagi dikodekan secara kaku (lihat Gambar 9.23). Perhatikan bahwa melakukan hal itu berarti bahwa kombinasi VLOOKUP-MATCH telah digunakan untuk mengatasi satu

keterbatasan, yang lain masih tetap ada; yaitu, bahwa kumpulan data perlu disusun dengan kolom pencarian utama di sebelah kiri.

Currency	Xrate:€
Dollar	1.00
Euro	1.20
Pound	1.00

Location	Customer no.	Date	Amount (local currency)	Currency	Xrate:€
Germany	C314	Jan	32,143	Euro	1.20
London	C159	Jan	17,203	Pound	1.00
France	C265	Feb	16,993	Euro	1.20
US	C358	Feb	78,888	Dollar	1.60
London	C979	Feb	43,957	Pound	1.00
US	C323	Mar	25,898	Dollar	1.60
London	C486	Mar	26,856	Pound	1.00
London	C729	Mar	63,652	Dollar	1.60
US	C266	April	29,188	Dollar	1.60
Germany	C357	April	47,153	Euro	1.20
France	C989	May	51,737	Euro	1.20
London	C323	June	36,332	Pound	1.00
Germany	C486	July	33,363	Euro	1.20
Germany	C729	August	17,635	Euro	1.20

**GAMBAR 9.23 Menggunakan MATCH untuk Membuat Nomor Kolom Fleksibel dalam VLOOKUP**

Gambar 9.24 menunjukkan bagaimana kombinasi INDEX-MATCH dapat digunakan untuk menciptakan situasi di mana data kolom dapat berada dalam urutan apa pun dan kolom baru juga dapat disisipkan tanpa menyebabkan kesalahan (lihat Gambar 9.24). Perhatikan bahwa proses pencocokan dan pencarian dapat dilakukan secara terpisah (Kolom G dan H) atau sebagai fungsi tertanam tunggal (Kolom K).

Selain keterbatasan pada struktur set data dan sifat rawan kesalahan dalam penyisipan kolom, VLOOKUP (dan juga proses HLOOKUP) memiliki kelemahan lain:

- Jika kunci (di sisi kiri basis data) digunakan untuk menemukan nilai dalam beberapa kolom, pencocokan tersirat dari kunci ini dilakukan setiap waktu, sehingga efisiensi komputasi berkurang. Dengan melakukan satu langkah pencocokan (menggunakan MATCH) yang dirujuk oleh beberapa proses pencarian terpisah, keseluruhan proses menjadi lebih efisien secara komputasi.

Xrate:€	Currency
1.00	Dollar
1.20	Euro
1.00	Pound

Location	Customer no.	Date	Amount (local currency)	Currency	Posn	Xrate:€	Xrate:€
Germany	C314	Jan	32,143	Euro	2	1.20	=INDEX(\$B\$3:\$B\$5,MATCH(F8,\$D\$3:\$D\$5,0),1)
London	C159	Jan	17,203	Pound	3	1.00	=INDEX(\$B\$3:\$B\$5,MATCH(F9,\$D\$3:\$D\$5,0),1)
France	C265	Feb	16,993	Euro	2	1.20	=INDEX(\$B\$3:\$B\$5,MATCH(F10,\$D\$3:\$D\$5,0),1)
US	C358	Feb	78,888	Dollar	1	1.60	=INDEX(\$B\$3:\$B\$5,MATCH(F11,\$D\$3:\$D\$5,0),1)
London	C979	Feb	43,957	Pound	3	1.00	=INDEX(\$B\$3:\$B\$5,MATCH(F12,\$D\$3:\$D\$5,0),1)
US	C323	Mar	25,898	Dollar	1	1.60	=INDEX(\$B\$3:\$B\$5,MATCH(F13,\$D\$3:\$D\$5,0),1)
London	C486	Mar	26,856	Pound	3	1.00	=INDEX(\$B\$3:\$B\$5,MATCH(F14,\$D\$3:\$D\$5,0),1)
London	C729	Mar	63,652	Dollar	1	1.60	=INDEX(\$B\$3:\$B\$5,MATCH(F15,\$D\$3:\$D\$5,0),1)
US	C266	April	29,188	Dollar	1	1.60	=INDEX(\$B\$3:\$B\$5,MATCH(F16,\$D\$3:\$D\$5,0),1)
Germany	C357	April	47,153	Euro	2	1.20	=INDEX(\$B\$3:\$B\$5,MATCH(F17,\$D\$3:\$D\$5,0),1)
France	C989	May	51,737	Euro	2	1.20	=INDEX(\$B\$3:\$B\$5,MATCH(F18,\$D\$3:\$D\$5,0),1)
London	C323	June	36,332	Pound	3	1.00	=INDEX(\$B\$3:\$B\$5,MATCH(F19,\$D\$3:\$D\$5,0),1)
Germany	C486	July	33,363	Euro	2	1.20	=INDEX(\$B\$3:\$B\$5,MATCH(F20,\$D\$3:\$D\$5,0),1)
Germany	C729	August	17,635	Euro	2	1.20	=INDEX(\$B\$3:\$B\$5,MATCH(F21,\$D\$3:\$D\$5,0),1)

**Gambar 9.24 Menggunakan INDEX-MATCH sebagai Pengganti VLOOKUP**



- Model yang memuat fungsi-fungsi ini sering kali sulit diaudit. Hal ini sebagian disebabkan oleh rentang preseden untuk setiap sel yang memuat fungsi VLOOKUP adalah seluruh rentang pencarian (dua dimensi), yang sering kali sangat besar. Dengan demikian, penelusuran mundur potensi kesalahan sering kali menjadi sangat sulit, dan ukuran berkas (digital) model juga sangat besar.
- Kumpulan data sulit direstrukturisasi, karena harus berupa rentang dua dimensi yang bersebelahan. Saat mengembangkan model, biasanya penting untuk memiliki fleksibilitas dalam memindahkan data (misalnya untuk meminimalkan panjang jalur audit dengan menempatkan data cukup dekat dengan rumus yang digunakan). Selain itu, banyak model memiliki satu sumbu yang secara logis dominan (seperti sumbu waktu dalam model tradisional, atau daftar pengenalan unik dalam model basis data); dalam kasus seperti itu pendekatan satu dimensi untuk mencari proses sering kali lebih disukai, lebih fleksibel, dan lebih tangguh.

Gambar 9.25 memberikan contoh di mana setiap fungsi VLOOKUP secara implisit mencocokkan kunci skenario (daripada pencocokan tunggal yang dilakukan untuk setiap kunci, diikuti oleh proses untuk mencari nilai, seperti yang ditunjukkan sebelumnya); seperti yang disebutkan di atas, pencarian skenario yang sama beberapa kali akan mengurangi efisiensi komputasi (kecepatan perhitungan ulang) (lihat Gambar 9.25). Gambar 9.26 dan 9.27 menunjukkan contoh sederhana dari proses penelusuran dependen dan preseden, yang menunjukkan bahwa setiap sel dalam rentang pencarian adalah preseden untuk setiap fungsi VLOOKUP yang merujuk ke rentang tersebut.

	A	B	C	D	E	F	G	H	I
1									
2		Revenue Scenari	2018	2019	2020	2021	2022		
3		A	387	407	439	483	520		
4		B	431	452	488	537	577		
5		C	474	497	537	591	635		
6									
7		Scenarios to Use	2018	2019	2020	2021	2022		
8		B	431	452	488	537	577	=VLOOKUP(\$B8,\$B\$3:\$G\$5,\$G\$7-\$C\$7+2)	
9									

**Gambar 9.25 Ketidakefisienan Komputasi VLOOKUP Ketika Beberapa Item Dicari yang Memerlukan Kunci Dasar yang Sama**

	A	B	C	D	E	F	G	H	I
1									
2		Revenue Scenari	2018	2019	2020	2021	2022		
3		A	387	407	439	483	520		
4		B	431	452	488	537	577		
5		C	474	497	537	591	635		
6									
7		Scenarios to Use	2018	2019	2020	2021	2022		
8		B	431	452	488	537	577	=VLOOKUP(\$B8,\$B\$3:\$G\$5,\$G\$7-\$C\$7+2)	

**Gambar 9.26 Penelusuran Ketergantungan saat Fungsi VLOOKUP Digunakan**

	A	B	C	D	E	F	G	H	I
1									
2		Revenue Scenari	2018	2019	2020	2021	2022		
3		A	387	407	439	483	520		
4		B	431	452	488	537	577		
5		C	474	497	537	591	635		
6									
7		Scenarios to Use	2018	2019	2020	2021	2022		
8		B	431	452	488	537	577	=VLOOKUP(\$B8,\$B\$3:\$G\$5,\$G7-\$C\$7+2)	
9									

**Gambar 9.27** Penelusuran Preseden saat Fungsi VLOOKUP Digunakan

	A	B	C	D	E	F	G	H
1								
2		Revenue Scenari	2018	2019	2020	2021	2022	
3		A	387	407	439	483	520	
4		B	431	452	488	537	577	
5		C	474	497	537	591	635	
6								
7		Scenarios to Use	2018	2019	2020	2021	2022	
8		B	431	452	488	537	577	=INDEX(\$G\$3:\$G\$4,\$A8,1)
9								

**Gambar 9.28** Transparansi dan Efisiensi Komputasi yang Lebih Besar dari Pendekatan INDEX/MATCH

Gambar 9.28 menunjukkan implementasi alternatif, yang pertama-tama membuat sel yang berisi fungsi MATCH untuk menemukan baris relevan tempat data harus diambil (Sel A8), dan fungsi INDEX dijalankan dari kolom data individual. Struktur dependensi jauh lebih jelas dan lebih efisien secara komputasi (lihat Gambar 9.28). Perlu dicatat bahwa meskipun fungsi MATCH dapat disematkan dalam setiap fungsi INDEX, di mana item pencocokan yang sama (mis. skenario B yang sesuai dengan kolom 2) digunakan untuk menjalankan beberapa proses pencarian (mis. fungsi di Sel C8:G8), akan lebih efisien secara komputasi untuk memiliki fungsi MATCH secara eksplisit dalam satu sel, sehingga hanya satu proses pencocokan untuk setiap item yang akan terjadi.

Meskipun contoh-contoh di atas telah ditunjukkan dengan mengacu pada fungsi VLOOKUP, komentar analog berlaku untuk HLOOKUP, jadi kami membahas ini hanya dengan satu contoh spesifik. Gambar 9.29 menunjukkan contoh sifat HLOOKUP yang rawan kesalahan jika baris baru ditambahkan. Gambar 9.29 menunjukkan rangkaian rumus awal yang memilih nilai kolom biaya dengan benar, dan Gambar 9.30 menunjukkan bagaimana penggunaan nomor baris yang dikodekan secara kaku akan menyebabkan kesalahan jika baris dimasukkan tanpa menyesuaikan nomor baris secara manual (yaitu menghasilkan Baris 17 yang menunjukkan data pendapatan, bukan informasi biaya).

	A	B	C	D	E	F	G	H
1								
2			2018	2019	2020	2021	2022	
3		Costs	800	840	882	926	972	
4		Capex	100	105	110	116	122	
5								
6			2018	2019	2020	2021	2022	
7		Costs	800	840	882	926	972	=HLOOKUP(G\$6,\$B\$2:\$G\$4,2)
8								

Gambar 9.29 Penggunaan HLOOKUP untuk Memilih Nilai dari Tabel

9								
10								
11			2018	2019	2020	2021	2022	
12		Revenues	1000	1050	1103	1158	1216	
13		Costs	800	840	882	926	972	
14		Capex	100	105	110	116	122	
15								
16			2018	2019	2020	2021	2022	
17		Costs	1000	1050	1103	1158	1216	=HLOOKUP(G\$16,\$B\$11:\$G\$14,2)
18								

Gambar 9.30 Potensi Kesalahan saat Baris Dimasukkan dalam Rentang HLOOKUP

### Contoh: Membandingkan INDEX-MATCH dengan LOOKUP

Fungsi LOOKUP tersedia dalam dua bentuk: bentuk vektor dan bentuk array. Meskipun menu Bantuan Excel menyarankan penggunaan VLOOKUP atau HLOOKUP sebagai ganti bentuk array, penulis umumnya menyarankan penggunaan pendekatan INDEX-MATCH sebagai ganti keduanya! Faktanya, bentuk vektor mirip dengan pendekatan INDEX-MATCH: fungsi mencari nilai tertentu dalam rentang satu dimensi (nilai yang harus dalam urutan menaik) dan mengembalikan nilai dari posisi terkait dalam rentang satu dimensi lainnya (jika nilai tertentu tidak dapat ditemukan dalam rentang pencarian pertama, cocok dengan nilai terbesar yang lebih kecil atau sama dengan nilai yang ditentukan; fungsi mengembalikan #N/A jika nilai yang ditentukan lebih kecil dari semua nilai dalam vektor pencarian).

Gambar 9.31 berisi contoh yang mirip dengan contoh di atas, di mana nama mata uang pertama kali dicari dalam satu bidang (E3:E5) yang menyediakan posisi jumlah mata uang yang relevan di bidang lainnya (C3:C5) (lihat Gambar 9.31). Singkatnya, pendekatan terbaik secara keseluruhan tampaknya adalah menggunakan pendekatan INDEX-MATCH pada dasarnya semua kasus di mana VLOOKUP (atau HLOOKUP) dapat dipertimbangkan, dan bahkan dalam kasus di mana bentuk vektor fungsi LOOKUP dapat digunakan.

Fungsi VLOOKUP (dan HLOOKUP) menciptakan kurangnya fleksibilitas, rawan kesalahan, sulit diaudit, dan sering kali tidak efisien secara komputasi. Tentu saja, dalam model sederhana (misalnya yang berisi dua kolom data dengan nilai pencarian ditempatkan di kolom kiri), penggunaan VLOOKUP akan sedikit lebih cepat diimplementasikan daripada pendekatan INDEX-MATCH. Namun, sering kali muncul model awal yang kemudian dikembangkan atau ditambahkan, sehingga pilihan fungsi yang tidak tepat dengan cepat tertanam dalam model yang lebih besar dan sulit diperbaiki tanpa pengerjaan ulang yang signifikan.

Pendekatan INDEX-MATCH juga sedikit lebih disukai daripada bentuk vektor fungsi LOOKUP: pertama, pemisahan eksplisit dari satu langkah pencocokan lebih efisien jika hasilnya akan digunakan dalam beberapa proses pencarian berikutnya (daripada secara implisit melakukan pencocokan tertanam dalam setiap proses). Kedua, pendekatan ini menyediakan pendekatan yang konsisten (dan hemat) untuk pemodelan. Ketiga, fungsi LOOKUP tampaknya diperlakukan oleh Microsoft sebagai fungsi lama.

Location	Customer no.	Date	Amount (local currency)	Currency	Xrate/E
Germany	C314	Jan	32,143	Euro	1.20
London	C159	Jan	17,203	Pound	1.00
France	C285	Feb	16,993	Euro	1.20
US	C358	Feb	78,888	Dollar	1.00
London	C979	Feb	43,957	Pound	1.00
US	C323	Mar	25,898	Dollar	1.60
London	C486	Mar	26,856	Pound	1.00
London	C729	Mar	63,852	Dollar	1.60
US	C266	April	29,188	Dollar	1.60
Germany	C357	April	47,153	Euro	1.20
France	C989	May	51,737	Euro	1.20
London	C323	June	36,332	Pound	1.00
Germany	C486	July	33,363	Euro	1.20
Germany	C729	August	17,635	Euro	1.20

Gambar 9.31 Menggunakan Bentuk Vektor Fungsi LOOKUP sebagai Alternatif untuk INDEX-MATCH

### Contoh: Menemukan Nilai yang Paling Cocok Menggunakan Kombinasi Array dan Fungsi Lainnya

Fungsi pencarian dapat digunakan untuk menemukan nilai terdekat dalam kumpulan data yang cocok dengan nilai yang dimasukkan, dengan menggabungkan penggunaan beberapa fungsi Excel. Selain itu, pendekatan menggunakan rumus array dapat dipertimbangkan.

Data	Diff	ABS	Min	Value
472	319	319	24	24
358	206	206	9	9
117	-36	36	177	177
715	562	562		
757	604	604		
364	211	211		
801	648	648		
691	538	538		
177	24	24		
865	712	712		
953	800	800		
806	653	653		
757	604	604		
637	484	484		
792	639	639		
744	591	591		

Gambar 9.32 Menemukan Nilai Pencocokan Terdekat Menggunakan Rumus Pencarian dan Array

Gambar 9.32 menunjukkan contoh (lihat Gambar 9.32). Persyaratannya adalah bagi pengguna untuk mengetikkan nilai ke dalam Sel C20, dengan urutan fungsi menentukan nilai dalam daftar (sel C3:C8) yang paling dekat dengan ini. Perhitungan di sisi kanan (Kolom E ke kanan) adalah langkah perhitungan eksplisit, di mana perbedaan absolut antara setiap titik data dan nilai input dihitung (Kolom F) dan nilai minimumnya kemudian ditemukan (sel I2), posisi ini dalam kumpulan data ditemukan (sel I3) dan nilai dicari (sel I4). Perhitungan yang setara dapat dilakukan langsung sebagai rumus array (tanpa harus membuat Kolom E di sebelah kanan), seperti yang ditunjukkan di Sel C21. (Sel C22 menunjukkan variasi untuk referensi, di mana OFFSET digunakan untuk melakukan tahap pencarian akhir.)

### 9.5 APLIKASI PRAKTIS: INFORMASI TENTANG FUNGSI OFFSET DAN RENTANG DINAMIS

Fungsi OFFSET dapat sangat berguna untuk membuat rumus yang merujuk ke rentang yang fleksibel, dalam arti bahwa ukuran atau lokasinya dapat disesuaikan berdasarkan masukan atau perhitungan. Dalam contoh berikut, kami menunjukkan tiga variasi utama:

- Di mana rentang yang direferensikan adalah sel dan referensi yang dikembalikan adalah sel.
- Di mana rentang yang direferensikan adalah rentang dan referensi yang dikembalikan adalah rentang (tanpa menggunakan argumen tinggi dan lebar opsional dari fungsi tersebut).
- Di mana rentang yang direferensikan adalah sel dan referensi yang dikembalikan adalah rentang (dengan menggunakan argumen tinggi dan lebar opsional dari fungsi tersebut).

Perhatikan bahwa fungsi INDEX juga ada dalam bentuk referensi (bukan bentuk array yang digunakan sebelumnya dalam bab ini), dan dengan demikian sering dapat digunakan untuk melakukan operasi yang mirip dengan beberapa operasi yang ditunjukkan di sini. Akan tetapi, kami menemukan bahwa penggunaan formulir tersebut biasanya kurang fleksibel dan kurang transparan dibandingkan formulasi lain, sehingga tidak dibahas lebih lanjut dalam teks ini.

#### Contoh: Rentang Fleksibel Menggunakan OFFSET (I)

Contoh sederhana penggunaan OFFSET untuk membuat rentang dinamis adalah menjumlahkan rentang dari satu titik ke titik lainnya. Misalnya, seseorang mungkin ingin menjumlahkan nilai di sepanjang baris antara dua sel yang lokasinya dapat bervariasi (misalnya, di mana sel awal dan akhir merupakan input model atau ditentukan dari perhitungan.) Gambar 9.33 menunjukkan contoh di mana pengguna menentukan bagian mana dari serangkaian arus kas yang akan dimasukkan dalam penjumlahan, dengan memberikan periode awal dan akhir sebagai input (lihat Gambar 9.33). Rumus di Sel C7 adalah:

$$SUM(OFFSET(D3,0,C6-1,1,C7-C6+1))$$

Dalam rumus ini, dua argumen terakhir dari fungsi OFFSET (yang merupakan argumen fungsi opsional) digunakan untuk membuat rentang tinggi satu dan lebar enam, dimulai dari sel keempat dalam rentang D3:M3 (yaitu berdasarkan masukan yang ditentukan pengguna di sel C6 dan C7).

	A	B	C	D	E	F	G	H	I	J	K	L	M
1													
2				1	2	3	4	5	6	7	8	9	10
3		Cash Flow/Model Results		-100	-50	-5	542	583	538	541	537	598	581
4													
5		Sum	3339	=SUM(OFFSET(D3,0,C6-1,1,C7-C6+1))									
6		Start Period	4										
7		Finish Period	9										

**Gambar 9.33 Menggunakan OFFSET untuk Menjumlahkan Antar Sel yang Ditentukan Pengguna**

Gambar 9.34 menunjukkan perluasan dari ini di mana titik awal untuk penjumlahan ditentukan dari perhitungan dalam model, khususnya untuk menjumlahkan arus kas dalam rentang enam tahun dimulai dari titik di mana arus kas positif pertama terdeteksi (dihitung oleh fungsi array di Sel C6).

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1														
2				1	2	3	4	5	6	7	8	9	10	
3		Cash Flow/Model Results		-100	-50	-5	542	583	538	541	537	598	581	
4														
5		Sum	3339	=SUM(OFFSET(D3,0,C6-1,1,C7-C6+1))										
6		Start Period	4	={MATCH(1,IF(D3:M3>0,1,0),0)}										
7		Finish Period	9	=C6+5										

**Gambar 25.34 Menggunakan OFFSET untuk Menjumlahkan Antar Sel yang Dihitung**

Dengan demikian, dalam model kehidupan nyata di mana arus kas di Baris 3 ditentukan melalui perhitungan berdasarkan masukan lain, titik awal untuk penjumlahan akan bergerak secara dinamis saat nilai masukan diubah (lihat Gambar 9.34). Perhatikan bahwa perhitungan serupa dapat dilakukan untuk menjumlahkan rentang secara mundur dari titik awal; ini mungkin diperlukan dalam perhitungan pajak di mana hanya sejumlah kerugian pajak tahun sebelumnya yang dapat dialihkan.

#### **Contoh: Rentang Fleksibel Menggunakan OFFSET (II)**

Contoh sederhana lain untuk membuat rentang fleksibel menggunakan OFFSET adalah rumus yang menjumlahkan item dalam baris di atasnya, sedemikian rupa sehingga jika baris dimasukkan, penjumlahannya akan tetap benar. Gambar 9.35 menunjukkan contoh (lihat Gambar 9.35). Perhatikan bahwa contoh ini bersifat warisan di Excel.

Dalam versi lama, pengenalan baris baru antara item terakhir dan rumus penjumlahan (yaitu antara Baris 10 dan 11 seperti yang ditunjukkan) akan menghasilkan rumus SUM (yang awalnya ditunjukkan di Sel C11, tetapi kemudian akan ada di Sel C12) yang tidak akan menyertakan baris baru ini. Di sisi lain, rumus di F11 secara otomatis menjumlahkan item hingga baris tepat di atas rumus, dan dengan demikian akan beradaptasi secara otomatis.

Perhatikan bahwa versi Excel terkini akan secara otomatis menyesuaikan rumus penjumlahan (C11) dengan memperluas rentangnya saat baris disisipkan dan nilai dimasukkan.

	A	B	C	D	E	F	G	H	I
1									
2			Standard			Using OFFSET			
3		Item 1	10			10			
4		Item 2	10			10			
5		Item 3	10			10			
6		Item 4	10			10			
7		Item 5	10			10			
8		Item 6	10			10			
9		Item 7	10			10			
10		Item 8	10			10			
11		Total	80	=SUM(C3:C10)		80	=SUM(F3:OFFSET(F11,-1,0))		

**Gambar 25.35 Menggunakan OFFSET untuk Menjumlahkan Baris di Atas**

### Contoh: Rentang Fleksibel Menggunakan OFFSET (III)

Sedangkan contoh di atas yang menggunakan OFFSET difokuskan pada perubahan ukuran rentang yang dirujuk, penggunaan lainnya terkait dengan perubahan set data yang dirujuk. Salah satu aplikasinya adalah dalam kalkulasi matriks korelasi yang di dalamnya terdapat beberapa variabel, atau yang datanya dari variabel tambahan dapat ditambahkan di beberapa titik mendarang.

Saat menggunakan rumus referensi, seperti CORREL(Range1, Range2), di mana Range1 dan Range2 adalah rentang input langsung, rumus yang dibuat dalam sel awal tidak dapat disalin ke seluruh matriks korelasi (karena tidak mungkin membuat rumus tunggal dengan struktur \$ yang benar (untuk referensi sel absolut) yang dapat disalin di kedua arah baris dan kolom). Di sisi lain, fungsi OFFSET dapat digunakan untuk membuat satu rumus yang rentangnya disesuaikan secara otomatis, dan karenanya dapat disalin ke seluruh rentang matriks.

Kecuali untuk matriks yang sangat kecil (seperti 2x2 atau 3x3), fungsi ini menghemat waktu dan memastikan bahwa penambahan berikutnya ke set data juga dapat dimasukkan dengan cepat. Gambar 9.36 menunjukkan sebuah contoh (lihat Gambar 9.36). Perhatikan bahwa rumus yang menggunakan fungsi OFFSET itu sendiri membuat rentang (dan bukan nilai). Oleh karena itu, rumus tersebut harus disematkan dalam fungsi lain (dan tidak dapat ditempatkan secara eksplisit dalam sel Excel). Dengan demikian, rumus di Sel F9 adalah: CORREL (OFFSET (\$C\$17 : \$C\$77,0, F\$8 1), OFFSET (\$C\$17 : \$C\$77,0,\$B9 1))

	1	2	3	4	
1	100%	24%	9%	60%	=CORREL(F17:F77,C17:C77)
2	24%	100%	34%	69%	
3	9%	34%	100%	53%	
4	60%	69%	53%	100%	

	1	2	3	4	
1	100%	24%	9%	60%	=CORREL(OFFSET(\$C\$17:\$C\$77,0,F\$8-1),OFFSET(\$C\$17:\$C\$77,0,\$B9-1))
2	24%	100%	34%	69%	
3	9%	34%	100%	53%	
4	60%	69%	53%	100%	

DATA	Daily returns			
Day	Asset 1	Asset 2	Asset 3	Asset 4
1	1.7%	1.0%	-2.5%	-0.9%
2	-2.1%	1.1%	1.1%	-0.5%
3	0.1%	2.9%	-1.3%	1.1%
4	0.0%	-1.7%	0.2%	-0.2%
5	-1.0%	-0.7%	-1.8%	-2.1%
6	1.1%	0.3%	0.9%	1.7%
7	-1.3%	-3.6%	-1.9%	-4.4%
8	0.0%	-1.1%	-1.3%	-1.3%
9	1.4%	-1.7%	-0.4%	-1.2%
10	-2.1%	-0.8%	0.3%	-1.7%

Gambar 9.36 Pembuatan Matriks Korelasi Menggunakan Satu Rumus

**Contoh: Rentang Fleksibel Menggunakan OFFSET (IV)**

Gambar 9.34 menunjukkan contoh akhir penggunaan fungsi OFFSET untuk menerjemahkan fungsi Excel dari satu bahasa ke bahasa lain. Sebagai titik awal pembahasan, Gambar 9.37 menunjukkan kemungkinan penggunaan fungsi VLOOKUP jika seseorang menerjemahkan dari bahasa yang selalu ada di kolom paling kiri dari kumpulan data. Seperti yang dibahas sebelumnya, pendekatan INDEX-MATCH memberikan lebih banyak fleksibilitas terlepas dari bagaimana data tersebut ditata. Namun, contoh-contoh ini didasarkan pada gagasan bahwa kedua bahasa tersebut bersifat tetap dan bahwa penerjemahan dilakukan dalam urutan yang diketahui (misalnya, dari bahasa Inggris ke bahasa Jerman dalam contoh-contoh ini).

	A	B	C	D	E	F	G
1							
2		Using VLOOKUP			Using MATCH, then INDEX		
3							
4		English	German		English	German	
5		COUNTA	ANZAHL2		COUNTA	ANZAHL2	
6			=VLOOKUP(B5,B9:C377,2)			=INDEX(E10:E377,MATCH(E5,F10:F377,0),1)	
7							
8							
9		English	German		German	English	
10		ABS	ABS		ABS	ABS	
11		ACCRINT	AUFGELZINS		AUFGELZINS	ACCRINT	
12		ACCRINTM	AUFGELZINSF		AUFGELZINSF	ACCRINTM	
13		ACOS	ARCCOS		ARCCOS	ACOS	
14		ACOSH	ARCCOSHYP		ARCCOSHYP	ACOSH	
15		ADDRESS	ADRESSE		ADRESSE	ADDRESS	
16		AMORDEGIF	AMORDEGRK		AMORDEGRK	AMORDEGRC	
17		AMORLINC	AMORLINEARK		AMORLINEAR	AMORLINC	
18		AND	UND		UND	AND	
19		AREAS	BEREICHE		BEREICHE	AREAS	
20		ASC	ASC		ASC	ASC	

Gambar 25.37 Penerjemahan Antara Dua Bahasa Tetap



Pada Gambar 9.38, kami menunjukkan contoh dari berkas yang sama, di mana penerjemahan dapat dilakukan dari bahasa apa pun ke bahasa lain. Dalam kasus ini, fungsi MATCH digunakan untuk menentukan posisi kolom relatif dari setiap bahasa dalam kumpulan data (misalnya, bahasa Jerman adalah kolom kedua). Rumus kuncinya adalah yang ada di sel I6 (yang menunjukkan nilai 60), yaitu:

$$\text{MATCH}(I5, \text{OFFSET}(H10 : H377, 0, H6 - 1), 0)$$

Dalam rumus ini, posisi kata yang akan diterjemahkan (di sini: Anzahl2) dicari dalam rentang kolom, yang ditentukan sebagai posisi yang diimbangi dari kolom pertama set data sesuai dengan bahasa tempat kata ini berada.

	H	I	J	K	L	M	N
	Using MATCH then OFFSET						
	Starting Language	Function	Ending Language	Translated Function			
	German	ANZAHL2	English	COUNTA			
		2	60	1	=OFFSET(H9,I6,I6-1)		
		=MATCH(H5,H9:L9,0)	=MATCH(I5,OFFSET(H:	=MATCH(J5,H9:L9,0)			
	English	German	French	Spanish	Italian		
	ABS	ABS	ABS	ABS	ASS		
	ACCRINT	AUFGEZINS	INTERET.ACC	INT.ACUM	INT.MATURATO.PER		
	ACCRINTM	AUFGEZINSF	INTERET.ACC.MAT	INT.ACUM.V	INT.MATURATO.SCAD		
	ACOS	ARCCOS	ACOS	ACOS	ARCCOS		
	ACOSH	ARCCOSHYP	ACOSH	ACOSH	ARCCOSH		
	ADDRESS	ADRESSE	ADRESSE	DIRECCION	INDIRIZZO		

Gambar 9.38 Penerjemahan Antar Dua Bahasa

## 9.6 APLIKASI PRAKTIS: FUNGSI INDIRECT DAN STRUKTUR BUKU KERJA ATAU DATA FLEKSIBEL

**Contoh: Contoh Sederhana Penggunaan INDIRECT untuk Merujuk ke Sel dan Lembar Kerja Lainnya**

Jika kolom teks dapat diartikan oleh Excel sebagai alamat sel atau rentang sel, fungsi INDIRECT dapat digunakan untuk menemukan nilai dalam sel atau rentang tersebut. Misalnya, pernyataan:

$$= \text{INDIRECT}("C2")$$

akan merujuk ke Sel C2, seolah-olah seseorang telah menggunakan rumus referensi sel langsung:

$$= C2$$

	A	B	C	D	E	F
1						
2		Revenue	100			
3						
4					100 =C2	
5					100 =INDIRECT("C2")	

**Gambar 9.39 Aplikasi Dasar Fungsi INDIRECT**

Gambar 9.39 menunjukkan beberapa contohnya. Gambar 9.39 menunjukkan contoh sederhana di mana nilai dalam Sel C2 diperlukan di bagian model selanjutnya (Kolom E). Pada Gambar 9.40, kami menunjukkan contoh yang diperluas, di mana referensi sel C2 dikodekan secara kaku (Sel E6) atau ditentukan menggunakan berbagai bentuk fungsi ADDRESS atau CELL. Pada Gambar 9.41, kami menunjukkan contoh lebih lanjut, di mana data diambil dari lembar kerja lain, menggunakan referensi langsung ke sel di lembar kerja lain dan referensi tidak langsung. Terakhir, Gambar 9.42 menunjukkan contoh fungsi yang diterapkan di mana bidang teks input mengidentifikasi rentang, bukan sel tunggal. Pendekatan ini digunakan dalam praktik pada contoh berikutnya.

6				C2	
7				100	=INDIRECT(E6)
8				\$C\$2	=ADDRESS(2,3)
9				100	=INDIRECT(E8)
10				\$C\$2	=ADDRESS(ROW(C2),COLUMN(C2))
11				100	=INDIRECT(E10)
12				\$C\$2	=CELL("address",C2)
13				100	=INDIRECT(E12)

**Gambar 9.40 Menggabungkan Fungsi INDIRECT dengan ADDRESS atau CELL**

15				100	=Data!C2
16				100	=INDIRECT("Data"&"!"&"C2")

**Gambar 9.41 Referensi Langsung dan Tidak Langsung Data pada Lembar Kerja Lain**

18		100	200	=SUM(INDIRECT("C18:C19"))
19		100		

**Gambar 9.42 Penggunaan Argumen Teks sebagai Rentang dalam Fungsi INDIRECT**

### Contoh: Menggabungkan Data dari Beberapa Model Lembar Kerja dan Pemodelan Skenario Fleksibel

Salah satu aplikasi paling ampuh dari penggunaan fungsi INDIRECT adalah untuk membuat model "berdasarkan data", yang di dalamnya terdapat beberapa lembar kerja yang berisi data dasar, dan yang di dalamnya pengguna menentukan lembar kerja tempat data akan

diambil. Fungsi ini digunakan untuk merujuk ke nilai-nilai pada lembar kerja yang ditentukan. Pendekatan semacam itu memungkinkan lembar kerja data baru untuk ditambahkan (atau yang lama dihapus) dengan upaya atau penyesuaian minimal, selama lembar kerja data tersebut memiliki struktur yang sama.

Aplikasi spesifik penting dari pendekatan ini adalah dalam pemodelan skenario, yang mana jumlah skenarionya tidak diketahui sebelumnya. Jika jumlahnya tetap (misalnya, tiga atau lima), maka lembar kerja data terpisah untuk setiap skenario dapat dibuat dalam model, dan data dipilih menggunakan pendekatan referensi sel langsung (seperti dengan fungsi CHOOSE atau INDEX). Jika jumlah skenario tidak diketahui, maka pendekatan multi-lembar ini memungkinkan lembar skenario untuk ditambahkan (atau dihapus). Gambar 9.36 berisi contohnya.

Perhatikan bahwa ada empat lembar kerja data yang mendasarinya dalam model asli, serta lembar kerja pemilihan data (bernama Intermediate). Pengguna menentukan lembar kerja tempat data akan diambil (dengan memasukkan namanya di Sel A1 lembar kerja Intermediate). Rumus dalam lembar kerja Intermediate masing-masing menemukan referensi selnya sendiri, sebelum fungsi INDIRECT digunakan untuk menemukan nilai dalam sel dengan referensi yang sama tetapi dalam lembar kerja data yang dipilih (lihat Gambar 9.43). (Untuk tujuan ilustrasi, baik CELL maupun ADDRESS telah digunakan.)

	A	B	C	D
1	Jan			
2		Revenue	100	=INDIRECT(\$A\$1&"I"&CELL("address",C2))
3		Cost	70	=INDIRECT(ADDRESS(ROW(C3),COLUMN(C3),,\$A\$1))
4				

**Gambar 9.43 Menggunakan INDIRECT untuk Memasukkan Data dari Lembar yang Ditentukan**

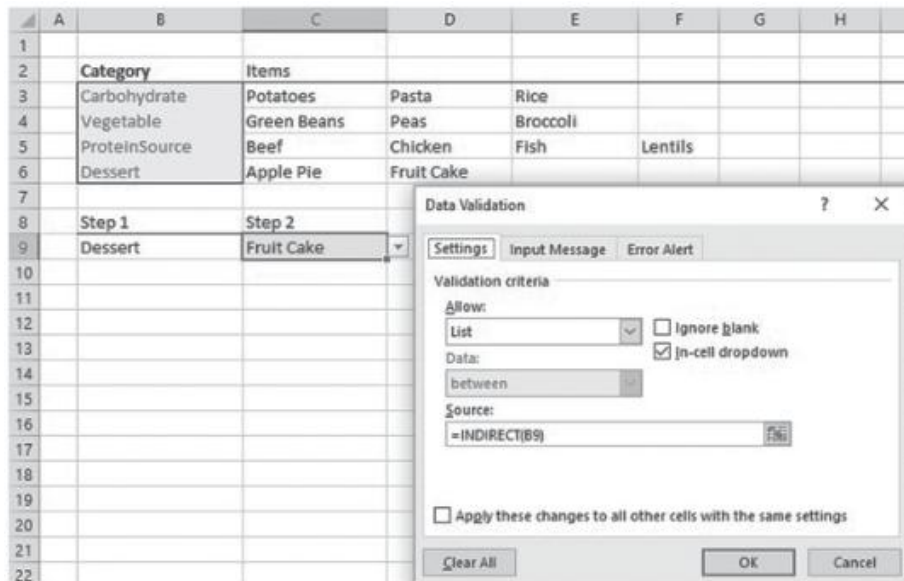
Perhatikan bahwa:

- Secara umum, dalam model kehidupan nyata, lembar kerja Intermediate sendiri akan digunakan untuk membuat tautan (langsung) ke (atau "umpan") lembar kerja model. Lembar kerja model tidak perlu memiliki struktur yang sama dengan lembar kerja Menengah, maupun lembar kerja data. Penggunaan langkah menengah ini (yaitu membawa data secara tidak langsung dari lembar data ke lembar kerja Menengah dan dari sana ke lembar model) memungkinkan seseorang untuk memastikan konsistensi yang ketat antara struktur sel lembar kerja Menengah dan lembar kerja data yang mendasarinya; ini diperlukan untuk memiliki proses referensi tidak langsung yang sederhana dan kuat.
- Meskipun penting secara umum bahwa lembar kerja data memiliki struktur yang mirip satu sama lain, dalam beberapa kasus praktis yang lebih kompleks, lembar kerja data juga dapat berisi perhitungan yang khusus untuk masing-masing lembar kerja (dan berbeda dari lembar kerja ke lembar kerja). Namun, asalkan ada area ringkasan di

setiap lembar yang memiliki struktur yang sama dengan lembar kerja data lainnya, maka pendekatan di atas masih dapat digunakan.

### Contoh: Penggunaan Lain INDIRECT – Daftar Drop-down Berjenjang

Tentu saja ada banyak kemungkinan penggunaan fungsi INDIRECT selain yang disebutkan di atas. Gambar 9.44 berisi contoh daftar drop-down yang peka terhadap konteks (atau menu berjenjang). Pengguna pertama-tama memilih kategori utama makanan, setelah itu fungsi INDIRECT digunakan untuk membuat drop-down baru, yang hanya mencantumkan item yang relevan dalam kategori yang dipilih (lihat Gambar 9.44). Perhatikan bahwa hal ini dicapai dengan menggunakan nama kategori sebagai Rentang Bernama Excel yang masing-masing merujuk ke daftar item dalam kategori tersebut (misalnya ProteinSource adalah rentang C5:F5), dan menggunakan fungsi INDIRECT dalam daftar Validasi Data (untuk Sel C9), sehingga daftar tersebut hanya merujuk ke item dalam kategori tersebut.



**Gambar 9.44 Drop-down Berurutan Menggunakan INDIRECT**

## 9.7 CONTOH PRAKTIS: PENGGUNAAN HYPERLINK UNTUK NAVIGASI MODEL DAN LINK DATA

Di bagian ini, kami sebutkan secara singkat beberapa fungsi yang dapat digunakan untuk menyediakan tautan ke set data:

- HYPERLINK membuat pintasan atau lompatan yang dapat digunakan untuk menyisipkan hyperlink dalam dokumen sebagai fungsi. Tautan dapat berupa rentang bernama dalam file Excel, atau lebih umum lagi tautan ke dokumen yang disimpan di server jaringan, intranet, atau Internet. Fungsi ini memiliki parameter opsional sehingga tautan yang ditampilkan juga dapat diberi "nama yang mudah diingat". Secara keseluruhan, konsep ini mirip dengan penggunaan operasi Sisipkan/Hyperlink, kecuali

bahwa operasi terakhir ini menghasilkan tautan langsung (bukan fungsi yang mengembalikan tautan).

- GETPIVOTDATA mengembalikan data yang disimpan dalam laporan PivotTable.
- RTD mengambil data waktu nyata dari program yang mendukung otomatisasi COM.

#### Contoh: Navigasi Model Menggunakan Rentang Bernama dan Hyperlink

Gambar menunjukkan contoh penggunaan fungsi HYPERLINK dan hyperlink yang disisipkan menggunakan operasi Sisipkan/Hyperlink, dalam setiap kasus untuk merujuk ke bagian model (Sel A1) yang didefinisikan dengan rentang bernama (DataAreaStart) (lihat Gambar 9.45).

	A	B
1	Area of Model Where Data Starts	
2		
529		
530		
531	<a href="#">Data Start Area</a>	=HYPERLINK(DataAreaStart,"Data Start Area")
532	<a href="#">Data Start Area</a>	#N/A

**Gambar 9.45 Perbandingan dan Penggunaan Fungsi HYPERLINK atau Menu Insert/Hyperlink**

## **BAB 10**

### **FILTER, FUNGSI BASIS DATA, DAN PIVOTTABLE**

#### **10.1 PENDAHULUAN**

Bab ini membahas beberapa fungsi dan fungsionalitas Excel yang dapat membantu dalam analisis kumpulan data, termasuk:

- Filter dan Filter Lanjutan, yang menyajikan atau mengekstrak kumpulan data yang difilter yang elemennya memenuhi kriteria tertentu.
- Fungsi basis data, yang mengembalikan kalkulasi nilai dalam basis data yang memenuhi kriteria tertentu (tanpa mengekstraksi titik data yang relevan secara eksplisit), dan yang hasilnya ditautkan langsung ke kumpulan data. Meskipun kalkulasi yang setara umumnya dapat dilakukan dengan fungsi Excel biasa (seperti SUMIFS atau MAXIFS), fungsi Basis Data memungkinkan identitas kriteria diubah lebih cepat. Sebagai fungsi, fungsi ini umumnya mengharuskan kesalahan atau nilai spesifik lainnya dalam kumpulan data diperbaiki atau ditangani secara eksplisit melalui kriteria baru, untuk menghilangkannya dari proses kalkulasi.
- PivotTable, yang membuat laporan ringkasan menurut kategori dan tabulasi silang. Meskipun tidak terhubung langsung ke set data (jadi perlu disegarkan jika data berubah), PivotTable memungkinkan pembuatan laporan yang sangat cepat, perubahan identitas kriteria, dan "penelusuran" atau analisis data yang lebih terperinci yang dapat dilakukan dengan sangat cepat. PivotTable juga memungkinkan kesalahan atau nilai spesifik lainnya dalam data yang mendasarinya diabaikan (atau difilter) saat menerapkan kriteria atau filter.

Bab ini tidak membahas tentang penautan Excel ke set data eksternal, seperti database Access atau server SQL, atau opsi lain yang terdapat pada tab Data; teknik tersebut berada di luar cakupan teks ini.

#### **10.2 MASALAH UMUM YANG BERLAKU DALAM BEKERJA DENGAN SET DATA**

##### **Membersihkan dan Memanipulasi Data Sumber**

Dalam praktiknya, seseorang sering kali harus memanipulasi atau membersihkan data awal sebelum dapat digunakan untuk keperluan analisis dan pelaporan. Ada banyak teknik yang mungkin perlu diterapkan dalam kasus seperti itu, termasuk:

- Memisahkan item menjadi komponen terpisah menggunakan fungsi Data/Teks-ke-Kolom atau fungsi Excel (seperti fungsi Teks, lihat Bab 7).
- Menggabungkan kembali atau menggabungkan item berdasarkan pencarian (atau pembuatan) kunci dalam satu set yang secara unik cocok dengan kunci di set lain (lihat Bab 7, Bab 8, dan Bab 9 untuk banyak teknik dasar yang sering diperlukan).
- Membalikkan atau mentransposisi data, menggunakan fungsi Salin dan

Tempel/PasteSpecial atau fungsi (misalnya fungsi array atau pencarian, lihat Bab 2 dan Bab 9).

- Mengidentifikasi kesalahan ejaan atau pengenalan yang tidak jelas. Misalnya, mungkin nama negara (seperti Italia) tercantum dengan beberapa pengenalan (seperti Italy, Italie, Italia, Repubblica Italiana, atau cukup It). Demikian pula, mungkin ada bagian kosong atau item lain yang tidak diinginkan dalam kumpulan data. Sebagian besar dapat diidentifikasi dengan penerapan Filter, diikuti dengan pemeriksaan menu drop-down, yang akan mencantumkan nilai unik. Atau, kolom data lengkap dapat disalin ke bidang terpisah dan teknik seperti Data/Remove Duplicates dan/atau Data/Sort diterapkan. Seseorang juga dapat menggunakan opsi Conditional Formatting Excel untuk menyorot nilai tertentu, seperti kesalahan, duplikat, item positif atau negatif, dll.
- Memperbaiki kesalahan ejaan atau pengenalan yang tidak jelas. Ini dapat dilakukan secara manual ketika hanya ada beberapa entri, atau menggunakan operasi seperti Find/Replace.
- Menghapus item yang tidak diinginkan. Beberapa item, seperti baris kosong atau entri yang tidak relevan untuk analisis apa pun, mungkin perlu dihapus. Hal ini dapat dilakukan secara manual jika hanya ada beberapa item, atau menggunakan Filter jika ada lebih banyak item (lihat contoh di bawah). Dalam kasus yang lebih luas, proses ini sebagian besar dapat diotomatisasi menggunakan makro. Alternatif untuk menghapus item yang tidak diinginkan adalah mengekstraknya; ini dapat dicapai menggunakan fungsi Filter Lanjutan (lihat di bawah).
- Untuk mengidentifikasi daftar item unik dalam suatu kategori (seperti nama negara dan pelanggan), atau daftar unik item gabungan (seperti kombinasi unik negara-pelanggan). Sekali lagi, fungsi Data/Hapus Duplikat dapat membantu dalam hal ini.

### Kueri Statis atau Dinamis

Secara umum saat menganalisis kumpulan data, seseorang memiliki dua pendekatan umum untuk membuat laporan ringkasan:

- Laporan yang sepenuhnya terhubung langsung ke data serta kriteria, yaitu sebagai fungsi. Penggunaan fungsi Database (atau mungkin fungsi Excel biasa seperti SUMIFS atau MAXIFS) termasuk dalam kategori ini, dan umumnya lebih sesuai jika:
- Struktur kumpulan data pada dasarnya sudah ditetapkan dan dipahami dengan baik, dengan sedikit analisis eksploratif (“drill-down”) yang diperlukan.
- Nilai data akan diperbarui secara berkala, sehingga perhitungan dinamis akan otomatis diperbarui dengan sedikit atau tanpa pengerjaan ulang.
- Identitas kriteria yang digunakan relatif stabil dan sederhana.
- Data mengandung sedikit kesalahan atau, jika ada kesalahan, kesalahan tersebut dapat dengan mudah (idealnya otomatis) diidentifikasi, dihilangkan, atau ditangani dengan tepat.
- Laporan merupakan langkah sementara dalam proses pemodelan, dengan hasil yang

digunakan untuk melakukan perhitungan lebih lanjut, dan diharapkan (dari perspektif pemodelan) memiliki tautan dinamis penuh dari data yang mendasarinya hingga ke hasil model.

- Laporan yang merupakan ringkasan numerik dari data, tetapi tidak sepenuhnya terhubung secara langsung. Penggunaan Filter dan PivotTable termasuk dalam kategori ini. Pendekatan semacam itu sering kali sesuai jika seseorang ingin mengeksplorasi hubungan antara item, melakukan analisis drill-down, mengubah kriteria yang digunakan untuk menyajikan laporan, atau mengubah penyajian laporan (misalnya mengubah presentasi baris menjadi kolom).

### **Pembuatan Kolom Baru atau Filter Kompleks?**

Saat melakukan kueri basis data kompleks (berdasarkan beberapa kriteria dasar atau kriteria yang tidak biasa), seseorang memiliki pilihan untuk menerapkan kriteria tersebut hanya pada tingkat pelaporan (sambil membiarkan set data tidak berubah), atau menambahkan kolom terhitung baru ke set data, dengan kolom tersebut digunakan untuk mengidentifikasi apakah set kriteria yang lebih kompleks telah terpenuhi (misalnya dengan menghasilkan kolom bendera yang bernilai 0 atau 1). Pendekatan terakhir ini sering kali lebih sederhana dan lebih transparan daripada membuat kombinasi kriteria kompleks pada tingkat kueri.

Tentu saja, pendekatan ini meningkatkan ukuran set data, sehingga jika banyak kombinasi tersebut diperlukan, akan lebih efisien (atau hanya mungkin) untuk membuatnya dalam struktur pelaporan. Dalam beberapa kasus, pendekatan gabungan mungkin diperlukan. Misalnya, jika salah satu kriteria kueri adalah panjang nama negara, atau huruf ketiga dari nama tersebut, maka kemungkinan besar perlu membuat kolom baru (misalnya menggunakan fungsi LEN atau MID) untuk membuat kolom ini secara eksplisit dalam basis data.

### **10.3 BASIS DATA DAN TABEL EXCEL**

Basis data Excel adalah rentang sel yang bersebelahan, yang setiap barisnya berisi data yang berkaitan dengan properti suatu item (misalnya tanggal lahir, alamat, nomor telepon orang), dengan serangkaian tajuk kolom yang mengidentifikasi properti tersebut. Struktur kolom basis data membedakannya dari kumpulan data Excel umum, yang rumusnya dapat diterapkan pada baris atau kolom.



	A	B	C	D	E	F	G	H	I
1									
2		<b>Nama</b>		<b>Kompensasi 2022</b>			<b>Kompensasi 2023</b>		
3		<b>First</b>	<b>Last</b>	<b>Gaji</b>	<b>Bonus</b>	<b>Pensiun</b>	<b>Gaji</b>	<b>Bonus</b>	<b>Pensiun</b>
4		data	data	data	data	data	data	data	data
5		data	data	data	data	data	data	data	data
6		data	data	data	data	data	data	data	data
7		data	data	data	data	data	data	data	data
8		data	data	data	data	data	data	data	data
9		data	data	data	data	data	data	data	data
10		data	data	data	data	data	data	data	data
11		data	data	data	data	data	data	data	data
12		data	data	data	data	data	data	data	data
13		data	data	data	data	data	data	data	data

**Gambar 10.1** Pengidentifikasi Bidang Beberapa Baris Harus Dihindari Saat Menggunakan Basis Data dan Tabel

	A	B	C	D	E	F	G	H	I
1									
2									
3		<b>Nama.First</b>	<b>Nama.Last</b>	<b>Komp.Gaji.2022</b>	<b>Komp.Bonus.2022</b>	<b>Komp.Pensiun.2022</b>	<b>Komp.Gaji.2023</b>	<b>Komp.Bonus.2023</b>	<b>Komp.Pensiun.2023</b>
4		data	data	data	data	data	data	data	data
5		data	data	data	data	data	data	data	data
6		data	data	data	data	data	data	data	data
7		data	data	data	data	data	data	data	data
8		data	data	data	data	data	data	data	data
9		data	data	data	data	data	data	data	data
10		data	data	data	data	data	data	data	data
11		data	data	data	data	data	data	data	data
12		data	data	data	data	data	data	data	data
13		data	data	data	data	data	data	data	data
14									
15									

**Gambar 10.2** Pengidentifikasi Bidang Baris Tunggal Harus Diutamakan Saat Menggunakan Basis Data dan Tabel

Perlu dicatat juga bahwa pada umumnya penting bagi nama bidang (tajuk) untuk berada dalam satu sel, dan pendekatan terstruktur harus digunakan untuk membuatnya (terutama untuk basis data yang lebih besar). Pendekatan yang sering digunakan, yang menggunakan kategori dan subkategori untuk menentukan tajuk menggunakan dua baris atau lebih, umumnya bukan pendekatan yang paling tepat. Sebaliknya, tajuk bidang harus berada dalam satu baris. Misalnya, alih-alih memiliki satu baris "tajuk" dan baris "subtajuk" di bawahnya, bidang tajuk dan subtajuk dapat digabungkan secara terstruktur. Hal ini diilustrasikan dalam Gambar 26.1, di mana rentang B2:I3 berisi pendekatan dua baris (yang tidak sesuai), dan dalam Gambar 26.2, di mana rentang B3:I3 berisi pendekatan baris tunggal yang diadaptasi. Tabel Excel adalah basis data yang telah ditetapkan secara eksplisit sebagai Tabel, menggunakan salah satu dari:

- Sisipkan/Tabel.
- Beranda/Format sebagai Tabel.
- Pintasan Ctrl+T (atau Ctrl+L).

Selama proses pembuatan Tabel, ada kemungkinan untuk menetapkan nama secara eksplisit untuknya; jika tidak, Excel akan secara otomatis menentukannya. Properti utama (dan manfaat) Tabel adalah:

- Ekstensi data saat baris data baru ditambahkan dalam baris yang bersebelahan di bagian bawah set data. Rumus apa pun yang merujuk ke tabel (menggunakan fungsi dan sintaksis terkait Tabel) akan secara otomatis beradaptasi untuk menyertakan data baru.
- Ekstensi rumus saat rumus dimasukkan ke dalam sel data pertama kolom baru; rumus secara otomatis disalin ke semua baris segera setelah dimasukkan.
- Pemformatan warna baris alternatif (secara default) untuk kemudahan membaca.
- Penamaan default berdasarkan indeks numerik dapat memfasilitasi otomatisasi beberapa prosedur saat menggunakan makro VBA.

(Setelah Tabel didefinisikan, tabel tersebut dapat diubah kembali menjadi rentang dengan mengklik kanan tabel tersebut dan menggunakan Alat Tabel/Ubah ke Rentang, atau dengan menggunakan tombol Ubah ke Rentang pada tab Desain yang muncul saat memilih titik mana pun dalam tabel.)

### **Otomatisasi Menggunakan Makro**

Dalam banyak aplikasi di dunia nyata, teknik dalam bab ini (dan teknik umum dalam manipulasi dan analisis data) menjadi jauh lebih efektif saat diotomatisasi dengan makro VBA. Misalnya:

- Meskipun filter individual (yang diaktifkan secara manual) dapat diterapkan untuk menghapus item dengan properti (seperti kosong), rutinitas VBA yang menghapus item tertentu dengan mengklik tombol dapat lebih efektif saat operasi berulang tersebut diperlukan.
- Untuk mengotomatiskan proses (dengan mengklik tombol) pembuatan daftar item unik untuk nama bidang (misalnya dengan mengotomatiskan dua langkah manual menyalin data dan menghapus duplikat).
- Untuk mengekstrak banyak subset data secara eksplisit dan menempatkan masing-masing di lembar kerja atau buku kerja baru.
- Untuk melakukan beberapa kueri menggunakan fungsi Basis Data untuk serangkaian kriteria yang strukturnya berubah (karena bidang kosong atau kolom yang dihitung tidak benar-benar kosong).
- Beberapa basis data dapat dibuat menggunakan bank rumus tunggal yang disalin ke sel individual dengan menggunakan makro. Proses penyalinan dapat dilakukan secara bertahap, setelah setiap rumus diganti dengan nilai, untuk mengurangi ruang memori dan intensitas komputasi.

Contoh teknik ini dibahas dalam Bab 32.

#### 10.4 APLIKASI PRAKTIS: FILTER

Contoh: Menerapkan Filter dan Memeriksa Data untuk Kesalahan atau Kemungkinan Koreksi Filter dapat diterapkan hanya dengan memilih sel mana pun dalam rentang basis data dan mengklik ikon Data/Filter. Perhatikan bahwa saat melakukannya, penting untuk memastikan bahwa tajuk kolom disertakan (atau baris teratas secara default akan dianggap berisi tajuk). Lebih jauh, seseorang harus mendefinisikan set data dalam rentang tunggal yang terisolasi dengan (selain dari tajuk) tanpa label yang berdampingan, dan tidak memiliki baris atau kolom yang sepenuhnya kosong dalam set data yang dimaksud.

Dalam hal ini, set data default akan menjadi Wilayah Saat Ini dari titik yang dipilih, yaitu wilayah persegi panjang berukuran maksimum yang didefinisikan dengan mengikuti semua jalur kesinambungan dari titik asal. Dalam istilah praktis, semua baris di atas dan di bawah disertakan hingga titik di mana baris kosong pertama ditemukan (dan serupa untuk kolom), sehingga penting untuk memastikan bahwa tidak ada baris kosong dalam set data, dan tidak ada label tambahan tepat di bawah set data atau di atas tajuk bidang (dan serupa dalam arah kolom).

Perhatikan bahwa Wilayah Saat Ini dari titik yang dipilih dapat dengan cepat dilihat dengan menggunakan pintasan Ctrl+Shift+\*. Gambar 10.3 menunjukkan contoh filter yang diterapkan pada kumpulan data. Perhatikan bahwa menu drop-down ditambahkan secara otomatis ke header, dengan drop-down ini menyediakan opsi pengurutan serta opsi pemfilteran yang sensitif terhadap bidang.

	A	B	C	D	E
1					
2					
3					
4		Customer	Country	Amount £	Due Date
5		Cust02	UK	12232	20-Mar-17
6		Cust06	Italy	4749	16-Mar-17
7		Cust07	Italy	7282	12-Apr-17
8		Cust03	Italy	12759	14-Jun-17
9		Cust10	UK	12334	24-May-17
10			Italy	4283	24-Mar-17
11		Cust06	Germany	7992	5-May-17
12		Cust06	Italy	13202	
13		Cust04	Germany	12684	4-Jun-17
14		Cust10	UK	11862	13-Jun-17
15		Cust10	Ita	13630	21-May-17
16		Cust07	UK	14593	20-Jan-17
17		Cust07		4394	4-May-17
18		Cust09	Italy	15712	8-Apr-17
19		Cust10	UK	6503	28-Mar-17
20		Cust05	France		8-Apr-17
21		Cust02	Germany	9274	17-Jun-17
22		Cust05	Italy	7919	27-Jun-17
23		Cust05	Italy	6402	14-Jun-17
24		Cust04	France	9100	6-Jun-17
25		Cust08	Spain	14120	2-Jul-17
26		Cust06	Spain	8889	8-Jun-17
27		Cust04	France	9547	1-Jun-17
28		Cust10	UK	14001	13-Jun-17
29		Cust01	Spain	4486	27-Apr-17
30		Cust05	Germany	9832	18-Apr-17
31		Cust08	Spain	9022	14-Apr-17

**Gambar 10.3** Bagian dari Kumpulan Data Awal Sebelum Memperbaiki Kesalahan Ejaan atau Bagian yang Kosong

Daftar opsi pemfilteran untuk bidang tertentu (misalnya untuk bidang Negara) menyediakan kemampuan untuk memeriksa kesalahan dengan cepat (misalnya kesalahan ejaan, atau angka negatif dalam konteks di mana semua angka seharusnya positif, serta melihat apakah ada item kosong). Gambar 10.3 menunjukkan contoh kumpulan data awal, dan Gambar 10.4 menunjukkan hasil penerapan, dalam bidang Negara, filter untuk memilih hanya bagian yang kosong atau ejaan yang salah dari negara Italia. Perhatikan bahwa potensi kerugian dari pendekatan filter standar ini adalah:

- Kriteria yang digunakan untuk memfilter kumpulan data tertanam dalam menu drop-down dan karenanya tidak eksplisit kecuali seseorang memeriksa filter.
- Untuk kumpulan data dengan banyak kolom, tidak selalu mudah untuk melihat filter mana yang telah diterapkan.

	A	B	C	D	E
1					
2					
3					
4		Customer	Country	Amount £	Due Date
15		Cust10	Ita	13630	21-May-17
17		Cust07		4394	4-May-17
34		Cust10	Italia	7250	13-May-17
61		Cust06		15842	1-Jul-17
105					

**Gambar 10.4 Data yang Difilter untuk Hanya Menampilkan Entri yang Salah atau Kosong dalam Kolom Negara**

	A	B	C	D	E
1					
2					
3					
4		Customer	Country	Amount £	Due Date
15		Cust10	Ita	13630	21-May-17
17		Cust07		4394	4-May-17
34		Cust10	Italia	7250	13-May-17
61		Cust06		15842	1-Jul-17
105					
106				1042744	=SUM(D5:D104)
107				41116	=SUBTOTAL(9,D5:D104)
108				41116	=SUBTOTAL(109,D5:D104)

**Gambar 10.5 Fungsi yang Diterapkan pada Set Data yang Difilter**

Cara mudah untuk menghapus semua filter adalah dengan pintasan Alt-D-F-S.

Perlu diperhatikan bahwa seseorang perlu ekstra hati-hati saat menerapkan sebagian besar fungsi Excel pada set data yang difilter, karena hasilnya mungkin tidak terduga atau tidak intuitif. Misalnya, untuk sebagian besar fungsi Excel biasa (misalnya SUM, COUNT), hasilnya

sama baik data terlihat atau tidak dan apakah difilter atau tidak.

Di sisi lain, fungsi SUBTOTAL memiliki opsi untuk menyertakan atau mengabaikan data tersembunyi (lihat Bab 17), tetapi opsi ini mungkin memiliki efek yang tidak terduga saat data difilter. Setelah filter diterapkan, baris yang disembunyikan (baik sebelum atau setelah penerapan filter) diperlakukan seolah-olah telah difilter (bukannya tidak difilter tetapi disembunyikan).

Gambar 10.5 menunjukkan fungsi yang diterapkan pada kumpulan data yang difilter, dan Gambar 10.6 menunjukkan hal yang sama, kecuali bahwa salah satu baris telah disembunyikan. Dalam setiap kasus, fungsi SUBTOTAL mengembalikan hasil yang sama, yaitu berdasarkan data yang terlihat, sehingga opsi fungsi mengenai data yang disembunyikan tampaknya tidak berpengaruh pada Gambar 10.6.

Pembaca dapat memverifikasi dalam berkas Excel bahwa dengan filter dihapus, penyembunyian baris akan menyebabkan dua fungsi SUBTOTAL memberikan hasil yang berbeda. Jadi, secara umum disarankan untuk tidak menerapkan fungsi Excel pada data yang difilter. Sebaliknya, penggunaan fungsi Basis Data atau pendekatan lain di mana kriteria dibuat eksplisit dalam fungsi (seperti menggunakan SUMIFS) lebih disukai.

	A	B	C	D	E
1					
2					
3					
4		Customer	Country	Amount £	Due Date
15		Cust10	Ita	13630	21-May-17
17		Cust07		4394	4-May-17
61		Cust05		15842	1-Jul-17
105					
106				1042744	=SUM(D5:D104)
107				33866	=SUBTOTAL(9,D5:D104)
108				33866	=SUBTOTAL(109,D5:D104)
109					

Gambar 10.6 Fungsi yang Diterapkan pada Set Data yang Difilter dengan Baris Tersembunyi

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1														
2														
3														
4	Customer	Country	Amount £	Due Date		Country								
5	Cust02	UK	12232	20-Mar-17		UK								
6	Cust06	Italy	4749	16-Mar-17		Italy								
7	Cust07	Italy	7282	12-Apr-17		Italy								
8	Cust03	Italy	12759	14-Jun-17		Italy								
9	Cust10	UK	13394	14-May-17		UK								
10	Cust05	Italy	4280	24-Mar-17		Italy								
11	Cust08	Germany	7992	5-May-17		Germany								
12	Cust06	Italy	13202	16-Apr-17		Italy								
13	Cust04	Germany	12684	4-Jun-17		Germany								
14	Cust10	UK	11882	13-Jun-17		UK								
15	Cust10	Italy	13630	21-May-17		Italy								
16	Cust07	UK	14593	20-Jan-17		UK								
17	Cust07	Italy	4394	4-May-17		Italy								
18	Cust09	Italy	15732	8-Apr-17		Italy								
19	Cust10	UK	6503	18-Mar-17		UK								
20	Cust05	France	10544	8-Apr-17		France								
21	Cust02	Germany	9274	17-Jun-17		Germany								
22	Cust06	Italy	7919	27-Jun-17		Italy								
23	Cust09	Italy	6402	14-Jun-17		Italy								
24	Cust04	France	9100	6-Jun-17		France								

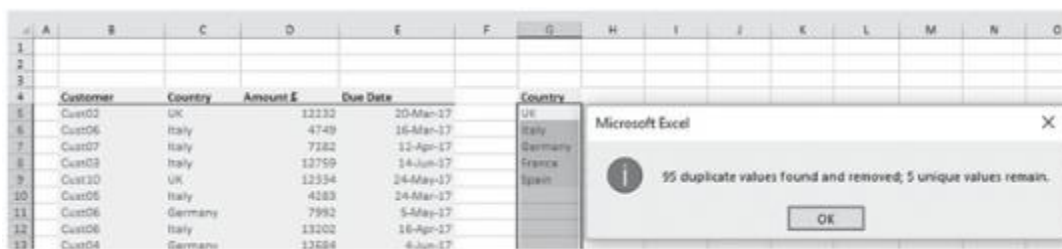
Gambar 10.7 Menerapkan Remove Duplicates pada Data yang Disalin

### Contoh: Identifikasi Item Unik dan Kombinasi Unik

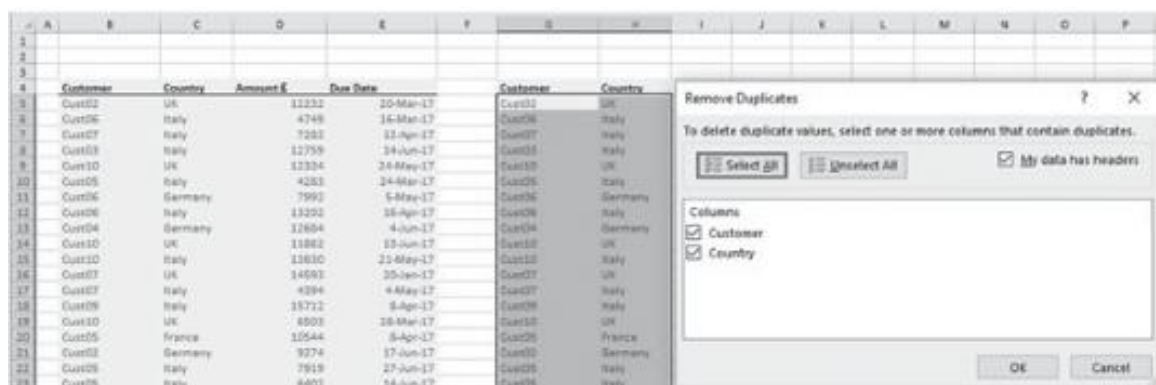
Untuk menganalisis set data, seseorang mungkin juga perlu mengidentifikasi daftar item unik dalam suatu kategori (seperti nama negara dan pelanggan), atau daftar unik item gabungan (seperti kombinasi unik negara-pelanggan). Penggunaan Remove Duplicates dari Excel (pada tab Data) merupakan fungsi penting dalam hal ini, yang memungkinkan seseorang untuk membuat daftar item unik yang sebenarnya. Gambar 10.7 menunjukkan sebuah contoh. Pada langkah pertama, seseorang menyalin semua data dalam data umum yang relevan (seperti nama negara) ke area lain yang tidak bersebelahan dengan set data (mungkin menggunakan pintasan Ctrl+Shift+↓ untuk memilih data).

Menu Remove Duplicates kemudian digunakan untuk membuat daftar item unik (dengan memperhatikan apakah kolom header telah disalin atau tidak). Gambar 10.7 menunjukkan langkah awal dari proses ini, menerapkan Remove Duplicates ke kumpulan data Negara yang disalin, dengan hasil yang ditunjukkan pada Gambar 10.8. Identifikasi kombinasi unik dapat dilakukan dengan salah satu dari dua cara:

- Membuat kolom teks yang menggabungkan (menyambung atau menggabungkan) item-item ini, dan menggunakan Remove Duplicates pada kolom item yang digabungkan.
- Menggunakan menu Remove Duplicates yang diterapkan pada kumpulan data yang disalin dari dua kolom.



Gambar 10.8 Item Unik yang Dihasilkan dari Penerapan Remove Duplicates



Gambar 10.9 Penerapan Remove Duplicates untuk Mengidentifikasi Semua Kombinasi Unik

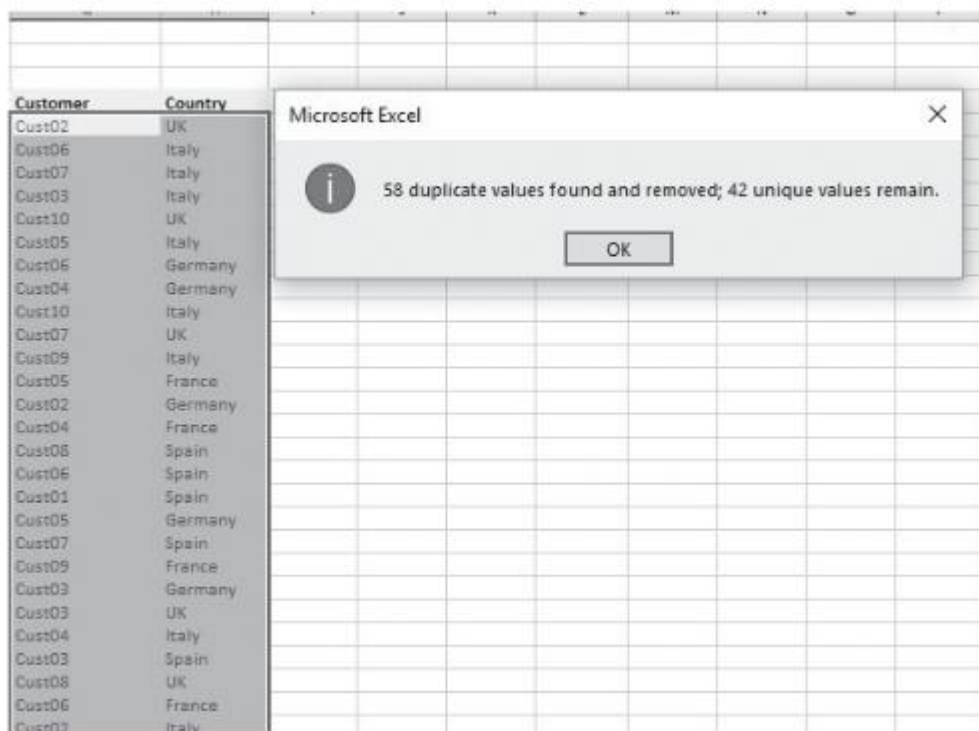
Gambar 10.9 menunjukkan hasil penggunaan pendekatan kedua. Gambar 10.9 menunjukkan langkah awal dalam penerapannya (yaitu menyalin set data yang relevan dan penerapan menu Remove Duplicated). Perhatikan bahwa jika bekerja dengan set data yang lebih besar (seperti jika seluruh set data telah disalin sehingga kolom untuk tanggal atau jumlah tidak relevan untuk definisi item unik), kotak centang yang ditampilkan dalam dialog Remove Duplicates untuk item ini akan dibiarkan tidak dicentang. Gambar 10.10 menunjukkan hasil dari proses tersebut.

**Contoh: Menggunakan Filter untuk Menghapus Item Kosong atau Item Tertentu Lainnya**

Saat membersihkan atau merapikan set data, seseorang mungkin ingin menghapus beberapa baris, termasuk:

- Jika data tidak lengkap.
- Jika terdapat kesalahan.
- Jika item spesifik lainnya tidak relevan atau diinginkan (seperti semua entri untuk negara Italia).
- Jika terdapat baris kosong.

Seperti yang disebutkan di atas, penggunaan pintasan Ctrl+Shift+\* untuk memilih data tidak akan memilih apa pun di bawah baris kosong pertama, jadi jika terdapat baris kosong dalam kumpulan data, pintasan ini tidak boleh digunakan; sebaliknya, kumpulan data harus dipilih secara eksplisit secara keseluruhan sebelum menerapkan filter.



**Gambar 10.10 Kombinasi Unik yang Dihasilkan dari Penerapan Hapus Duplikat**

Gambar 10.10 menunjukkan sebuah contoh. Langkah-langkah utamanya adalah:

- Terapkan filter untuk mengidentifikasi dan memilih item yang ingin dihapus (umumnya, sebaiknya simpan salinan kumpulan data asli, untuk berjaga-jaga jika terjadi kesalahan selama beberapa bagian proses dan langkah-langkah perlu diulang).
- Pilih rentang penuh baris yang difilter (ini dapat dilakukan dengan cara biasa, seperti memilih rentang sel yang bersebelahan, meskipun sel tersebut sebenarnya tidak bersebelahan).
- Hapus baris-baris ini dengan cara biasa (misalnya menggunakan Beranda/Hapus Sel atau klik kanan untuk mendapatkan menu yang peka terhadap konteks).
- Hapus filter.

Mari kita asumsikan bahwa diinginkan untuk menghapus semua rekaman yang bidang Negaranya kosong atau berlaku untuk Italia (yang diasumsikan keduanya dianggap tidak relevan untuk semua tujuan analisis selanjutnya). Gambar 10.11 menunjukkan bagian dari proses yang telah memfilter data, dan telah memilih baris filter sebagai persiapan untuk penghapusannya. Gambar 10.12 menunjukkan hasil ketika baris-baris ini telah dihapus dan filter dihapus, sehingga memberikan kumpulan data yang dibersihkan.

	A	B	C	D	E	F
1						
2						
3						
4	Customer	Country	Amount £	Due Date		
6	Cust06	Italy	4749	16-Mar-17		
7	Cust07	Italy	7282	12-Apr-17		
8	Cust03	Italy	12759	14-Jun-17		
9	Cust10		12334	24-May-17		
10	Cust05	Italy	4283	24-Mar-17		
12	Cust06	Italy	13202	16-Apr-17		
13	Cust04		12684	4-Jun-17		
15	Cust10	Italy	13630	21-May-17		
16	Cust07		14593	20-Jan-17		
17	Cust07	Italy	4394	4-May-17		
18	Cust09	Italy	15712	8-Apr-17		
21	Cust02		9274	17-Jun-17		
22	Cust05	Italy	7919	27-Jun-17		
23	Cust05	Italy	6402	14-Jun-17		
34	Cust10	Italy	7250	13-May-17		
40	Cust04	Italy	7768	31-May-17		
41	Cust05	Italy	11381	15-Apr-17		
45	Cust02	Italy	4449	14-Jul-17		
47	Cust10	Italy	11433	4-Jul-17		
55	Cust08	Italy	4876	10-May-17		
57	Cust05	Italy	14322	25-May-17		
61	Cust06	Italy	15842	1-Jul-17		
65	Cust10	Italy	13334	15-May-17		
69	Cust07	Italy	13911	13-Jul-17		
71	Cust10	Italy	14647	15-Aug-17		
74	Cust05	Italy	11079	31-Jul-17		
78	Cust05	Italy	4711	26-May-17		
91	Cust08	Italy	9057	29-Aug-17		
103	Cust06	Italy	8726	3-Sep-17		
105						

**Gambar 10.11 Pemfilteran dan Pemilihan Baris yang Ingin Dihapus**



Perlu dicatat bahwa proses ini dapat diotomatisasi dengan relatif mudah dengan makro VBA. Misalnya, saat makro dijalankan, pengguna dapat diminta untuk mengklik sel tertentu yang berisi pengenal item yang akan dihapus dalam bidang tersebut (misalnya sel kosong, atau sel yang berisi kata Italia), dengan makro yang menangani semua langkah lain di balik layar. Dengan cara ini, kumpulan data dapat dibersihkan dengan cepat: penyalinan, pemfilteran, dan penghapusan filter tertanam sebagai bagian dari keseluruhan proses, dan seseorang dapat menjalankan proses tersebut berkali-kali dengan sangat cepat.

#### Contoh: Ekstraksi Data Menggunakan Filter

Dalam beberapa kasus, seseorang mungkin ingin menggunakan filter untuk mengekstrak item daripada menghapusnya. Dalam kasus seperti itu, bagian awal dari proses dapat dilakukan seperti di atas (hingga yang ditunjukkan pada Gambar 10.11), sedangkan bagian terakhir akan melibatkan penyalinan data ke rentang baru, daripada menghapusnya. Ini pada prinsipnya merupakan operasi yang sangat mudah dan standar jika elemen dasar Excel dipatuhi (seperti memastikan bahwa operasi penyalinan apa pun tidak menimpa data asli). Perhatikan bahwa Excel akan mengganti item yang dihitung dengan nilainya untuk memastikan integritas proses.

	A	B	C	D	E
1					
2					
3					
4		Customer	Country	Amount £	Due Date
5		Cust02	UK	12232	20-Mar-17
6		Cust06	Germany	7992	5-May-17
7		Cust10	UK	11862	13-Jun-17
8		Cust10	UK	6503	28-Mar-17
9		Cust05	France	10544	8-Apr-17
10		Cust04	France	9100	6-Jun-17
11		Cust08	Spain	14120	2-Jul-17
12		Cust06	Spain	8889	8-Jun-17
13		Cust04	France	9547	1-Jun-17
14		Cust10	UK	14001	13-Jun-17
15		Cust01	Spain	4486	27-Apr-17
16		Cust05	Germany	9832	18-Apr-17
17		Cust08	Spain	9022	14-Apr-17
18		Cust02	Germany	14290	1-Jun-17
19		Cust07	Spain	10007	29-May-17
20		Cust09	France	15080	7-Apr-17
21		Cust03	Germany	9373	3-May-17
22		Cust07	UK	9006	26-Apr-17
23		Cust03	UK	8302	17-May-17
24		Cust03	Germany	10855	30-Jun-17

Gambar 10.12 Hasil Penghapusan Baris Tertentu

#### Contoh: Menambahkan Perhitungan Kriteria ke Set Data

Meskipun Excel menyediakan beberapa opsi bawaan mengenai filter, sering kali lebih

mudah untuk menambahkan kriteria baru ke set data. Gambar 10.13 menunjukkan contoh di mana Anda ingin mengidentifikasi dan menghapus semua baris yang dua atau lebih elemen kolomnya kosong. Kolom baru telah ditambahkan yang menggunakan fungsi COUNTBLANK untuk menghitung jumlah kolom kosong (lihat Gambar 10.13). Untuk memudahkan penyajian, Pemformatan Bersyarat telah diterapkan untuk menyorot sel-sel yang jumlah kolom kosongnya di barisnya sama dengan angka ambang batas (didefinisikan di Sel F2).

Dengan kata lain, pemformatan ini mengidentifikasi baris-baris yang mungkin ingin dihapus. Perhatikan bahwa (karena kumpulan data asli dalam contoh tersebut tidak didefinisikan secara eksplisit sebagai tabel Excel), setelah kolom ini ditambahkan, menu Data/Filter perlu diterapkan kembali untuk memastikan bahwa filter disertakan dalam kolom baru. Prosedur yang sama seperti dalam contoh di atas kemudian dapat digunakan untuk memfilter dan menghapus (atau menyalin) baris-baris ini.

#### Contoh: Penggunaan Tabel

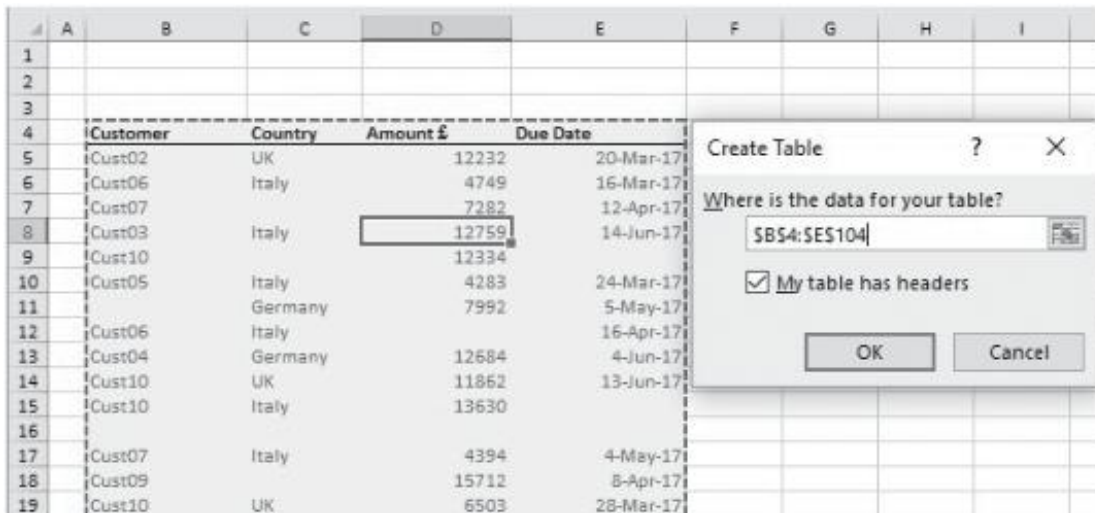
Secara umum, lebih baik untuk mendefinisikan kumpulan data sebagai Tabel Excel. Perhatikan bahwa seseorang perlu berhati-hati untuk memastikan bahwa, jika kumpulan data asli berisi baris kosong, maka rentang Tabel perlu diperluas untuk menyertakan semua data (karena operasi Sisipkan/Tabel hanya akan memilih Wilayah Saat Ini dari titik data secara default). Gambar 10.14 menunjukkan contoh proses untuk membuat Tabel (lihat Gambar 10.14), dengan memperhatikan bahwa rentang default hanya akan sampai Baris 15 (karena Baris 16 kosong), sehingga rentang diperluas secara manual untuk menyertakan baris hingga Baris 104.

	A	B	C	D	E	F
1						
2					Threshold	2
3						
4		Customer	Country	Amount £	Due Date	No. of Blan
5		Cust02	UK	12232	20-Mar-17	0
6		Cust06	Italy	4749	16-Mar-17	0
7		Cust07		7282	12-Apr-17	1
8		Cust03	Italy	12759	14-Jun-17	0
9		Cust10		12334		2
10		Cust05	Italy	4283	24-Mar-17	0
11			Germany	7992	5-May-17	1
12		Cust06	Italy		16-Apr-17	1
13		Cust04	Germany	12684	4-Jun-17	0
14		Cust10	UK	11862	13-Jun-17	0
15		Cust10	Italy	13630		1
16						4
17		Cust07	Italy	4394	4-May-17	0
18		Cust09		15712	8-Apr-17	1

**Gambar 10.13 Penambahan Kolom Baru untuk Menangkap Kriteria Tertentu atau Kompleks**

Setelah ditetapkan sebagai Tabel, setiap baris atau kolom baru (bersebelahan) yang ditambahkan akan secara otomatis menjadi bagian dari Tabel. Selain itu, rumus yang

dimasukkan dalam sel data pertama dari suatu kolom akan disalin ke semua baris tabel. Memang, entri sederhana dalam sel tersebut (seperti dalam Sel F5 dari Tabel asli yang tidak diperluas) akan mengakibatkan kolom baru ditambahkan secara otomatis. Gambar 10.15 menunjukkan proses yang sama seperti di atas (di mana fungsi COUNTBLANK digunakan sebagai kolom baru), saat diterapkan ke Tabel.



**Gambar 10.14 Menetapkan Rentang Data untuk Tabel**

	A	B	C	D	E	F
1						
2						
3						
4		Customer	Country	Amount £	Due Date	No. of Bills
5		Cust02	UK	12232	20-Mar-17	0
6		Cust06	Italy	4749	16-Mar-17	0
7		Cust07		7282	12-Apr-17	1
8		Cust03	Italy	12759	14-Jun-17	0
9		Cust10		12334		2
10		Cust05	Italy	4283	24-Mar-17	0
11			Germany	7992	5-May-17	1
12		Cust06	Italy		16-Apr-17	1
13		Cust04	Germany	12684	4-Jun-17	0
14		Cust10	UK	11862	13-Jun-17	0
15		Cust10	Italy	13630		1
16						4
17		Cust07	Italy	4394	4-May-17	0
18		Cust09		15712	8-Apr-17	1
19		Cust10	UK	6503	28-Mar-17	0

**Gambar 10.15 Penambahan Kolom Baru ke Tabel**

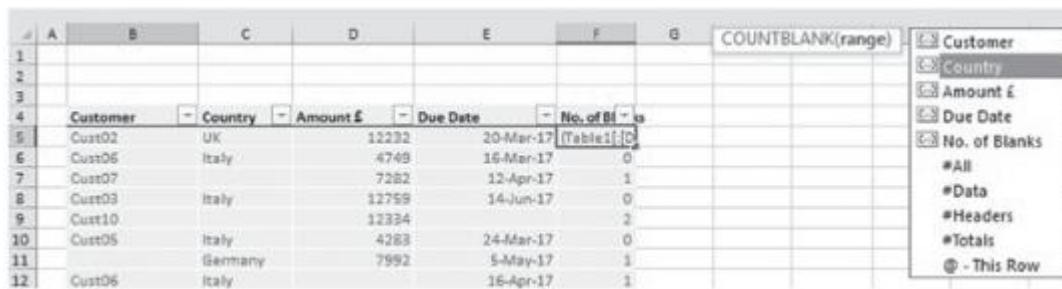
Perhatikan bahwa sintaks rumus di setiap sel dalam rentang Tabel kolom F adalah sama (dan tidak bergantung pada baris):

*COUNTBLANK(Tabel 1[@[Pelanggan]:[Tanggal Jatuh Tempo]])*

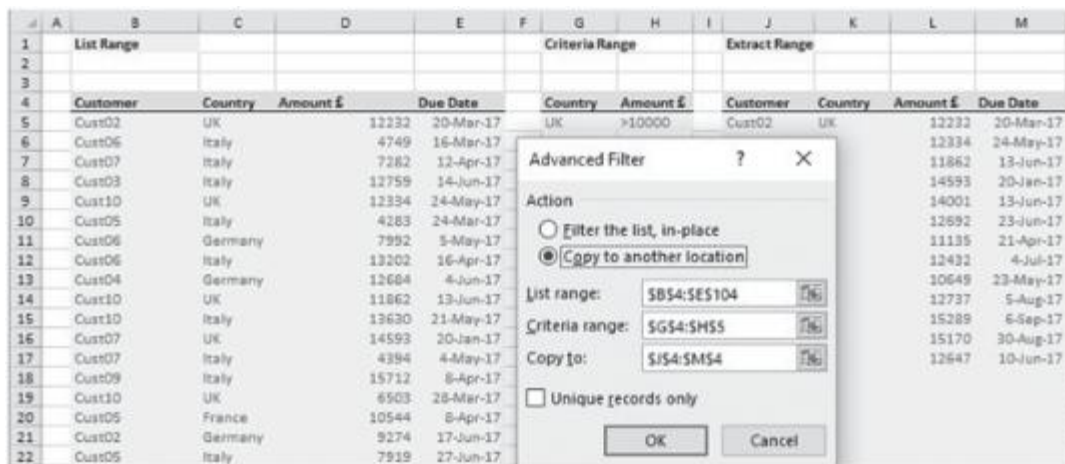
Jika seseorang ingin mengubah rumus, seperti menghitung kolom kosong dari titik yang berbeda, maka penghapusan nama bidang asli (seperti yang ditunjukkan pada Gambar 10.16) akan menghasilkan menu tarik-turun nama bidang yang dapat dipilih sebagai gantinya untuk membuat rumus yang diperlukan. Perhatikan juga bahwa, saat menggunakan Tabel, seseorang dapat membuat rumus menggunakan Nama Header yang dihasilkan Tabel, seperti:

$SUM(\text{Tabel 1}[\text{Jumlah } \text{£}])$

Terakhir, perhatikan juga bahwa saat seseorang telah memilih sel di dalam Tabel, Tab Desain Alat Tabel akan muncul. Tab ini berisi ikon untuk Menghapus Duplikat dan membuat ringkasan PivotTable. Tab ini juga memungkinkan seseorang untuk mengganti nama Tabel, meskipun pengalaman menunjukkan bahwa proses ini paling baik dilakukan saat Tabel pertama kali dibuat, dan bukan saat rumus telah dibuat yang merujuk ke nama tersebut.



Gambar 10.16 Modifikasi Rumus Menggunakan Nama Header yang Dihasilkan dalam Tabel



Gambar 10.17 Contoh Implementasi Filter Lanjutan

### Contoh: Ekstraksi Data Menggunakan Filter Lanjutan

Dalam beberapa contoh sebelumnya, kami menggunakan proses penyaringan yang tertanam dalam menu drop-down untuk menghapus data yang tidak diinginkan dan mengekstrak data yang mungkin diinginkan untuk analisis terpisah. Dalam praktiknya, pendekatan tersebut umumnya paling tepat hanya untuk penghapusan data, sedangkan

ekstraksi item yang diinginkan ke rentang lain lebih efektif diimplementasikan menggunakan alat Data/Filter Lanjutan.

Gambar 10.17 menunjukkan sebuah contoh (lihat Gambar 10.17). Perhatikan bahwa ada rentang terpisah untuk menentukan kriteria untuk ekstraksi dan rentang lain untuk menyalin data yang diekstraksi (rentang terakhir ini bersifat opsional; sesuai kotak dialog, daftar tersebut dapat difilter di tempat, sehingga set data lengkap asli akan difilter seolah-olah dilakukan dengan menu drop-down).

Perhatikan bahwa:

- Rentang Kriteria terdiri dari beberapa tajuk dan setidaknya satu baris di bawah yang bersebelahan dengan tajuk. Bila beberapa item dimasukkan dalam baris (non-header), diperlakukan sebagai representasi kriteria "And". Bila baris non-header tambahan digunakan, setiap baris diperlakukan sebagai kriteria "Or" (yaitu hasil kriteria pada satu baris ditambahkan ke hasil kriteria pada baris lain); kesalahan yang sering terjadi adalah memasukkan baris kosong dalam rentang kriteria, yang berarti semua elemen akan disertakan dalam proses ekstraksi atau penyaringan.
- Kolom header yang digunakan dalam rentang Criteria harus identik dengan kolom header pada set data utama. Bila ejaannya sedikit berbeda, proses tidak akan berjalan sebagaimana mestinya.
- Untuk rentang Extract, header penting hanya untuk tujuan kosmetik; set data yang diekstrak akan ditempatkan dalam berkas dalam urutan yang sama seperti pada set data asli, meskipun header tidak ada dalam rentang Extract. Tentu saja, dalam praktiknya, yang paling mudah adalah mendefinisikan rentang Extract sebagai yang berisi salinan header kolom (seperti dalam contoh kita).

## 10.5 APLIKASI PRAKTIS: FUNGSI BASIS DATA

Fungsi Basis Data Excel mengembalikan kalkulasi yang berkaitan dengan bidang tertentu dalam basis data, saat serangkaian kriteria diterapkan untuk menentukan rekaman mana yang akan disertakan. Fungsi yang tersedia adalah DAVERAGE, DCOUNT, DCOUNTA, DGET, DMIN, DMAX, DPRODUCT, DSTDEV, DSTDEVP, DSUM, DVAR, dan DVARP. Misalnya, DSUM menyediakan penjumlahan bersyarat dalam basis data, yaitu penjumlahan bidang yang hanya menyertakan item yang memenuhi kriteria yang ditentukan.

Dalam arti tertentu, DSUM mirip dengan fungsi SUMIFS (demikian pula, DAVERAGE mirip dengan AVERAGEIFS, dan DMIN atau DMAX mirip dengan MINIFS atau MAXIFS). Fungsi Basis Data lainnya umumnya tidak memiliki padanan non-basis data bersyarat sederhana (meskipun penggunaan fungsi array memungkinkan fungsi tersebut untuk direproduksi pada prinsipnya, seperti yang dibahas dalam Bab 18).

Untuk Filter Lanjutan, fungsi Basis Data mengharuskan seseorang untuk menentukan Basis Data (yang mungkin telah ditetapkan sebagai Tabel Excel), Rentang Kriteria (tajuk dan setidaknya satu baris, dengan baris bersebelahan tambahan yang mewakili kriteria ATAU) dan bidang tempat perhitungan bersyarat akan dilakukan. Fungsi basis data dan padanannya yang

bukan basis data menyediakan perhitungan yang dinamis atau terhubung langsung ke set data (berbeda dengan Tabel Pivot). Masing-masing memiliki kelebihan dibandingkan yang lain:

- Fungsi non-basis data memiliki kelebihan karena dapat berfungsi tanpa memandang apakah data ditata dalam bentuk baris atau kolom, dan dapat disalin ke dalam baris bersebelahan (sehingga fungsi dapat menyediakan serangkaian nilai yang terkait dengan kueri yang melibatkan nilai berbeda dari jenis kriteria dasar yang sama, misalnya jumlah semua item dalam bulan terakhir, atau dalam dua bulan terakhir, dst.)
- Fungsi basis data mengharuskan data dalam format kolom. Keunggulannya antara lain adalah memungkinkan identitas kriteria diubah dengan sangat cepat: dengan memasukkan nilai di area rentang kriteria yang sesuai, kriteria yang berbeda (bukan hanya nilai yang berbeda untuk kriteria yang sama) digunakan. Misalnya, seseorang mungkin ingin menghitung jumlah semua item dalam bulan lalu, lalu jumlah semua item senilai lebih dari Rp.10.000.000, atau semua item yang berkaitan dengan negara tertentu, dan seterusnya. Saat menggunakan padanan non-basis data (SUMIFS dalam contoh ini), seseorang harus mengedit fungsi untuk setiap kueri baru, untuk memastikan bahwa fungsi tersebut merujuk ke rentang yang benar dalam set data dan juga dalam rentang kriteria. Perhatikan juga bahwa fungsi Basis Data memerlukan penggunaan nama tajuk bidang (seperti halnya PivotTable), sedangkan fungsi non-basis data merujuk ke set data secara langsung, dengan tajuk dan label sebagai item kosmetik (seperti pada sebagian besar fungsi Excel lainnya).

### Contoh: Menghitung Jumlah Bersyarat dan Nilai Maksimum Menggunakan DSUM dan DMAX

Gambar 10.18 berisi sebuah contoh. Gambar 10.18 memperlihatkan bagian dari kumpulan data (yang telah ditetapkan sebagai Tabel, dan diberi nama DataSet1) serta informasi parameter tambahan yang diperlukan untuk menggunakan fungsi Basis Data, yaitu nama bidang yang akan dianalisis (dalam kasus ini bidang "Jumlah £") dan Rentang Kriteria (sel I4:L5). Gambar 10.19 memperlihatkan implementasi beberapa fungsi.

1	A	B	C	D	E	F	G	H	I	J	K	L	M
2		Database					Field	Criteria					
3													
4		Customer	Country	Amount £	Due Date		Amount £	Customer	Country	Amount £	Due Date		
5		Cust02	UK	12232	20-Mar-17				UK	>£10000			
6		Cust06	Italy	4749	16-Mar-17								
7		Cust07	Italy	7282	12-Apr-17								
8		Cust03	Italy	12759	14-Jun-17								
9		Cust10	UK	12334	24-May-17								

**Gambar 10.18** Basis Data dan Parameter yang Diperlukan untuk Menggunakan Fungsi Basis Data

M	N	O	P	Q	R	S	T	U
Dfunctions								
DCOUNT		DSUM	DMIN	DMAX	=DMAX(DataSet1[#All], \$G\$4, \$I\$4:\$L\$5)			
13		167773	10649	15289				

**Gambar 10.19 Implementasi Fungsi Basis Data Terpilih**

**Contoh: Implementasi Kueri Antar**

Jika seseorang ingin mengimplementasikan kueri antar-jenis, cara paling sederhana biasanya adalah memperluas Rentang Kriteria dengan menambahkan kolom dan menerapkan dua pengujian ke bidang kriteria yang sama; ini serupa dengan prosedur saat menggunakan fungsi SUMIFS, seperti yang dibahas sebelumnya dalam teks. (Alternatifnya adalah mengimplementasikan dua kueri terpisah dan mengambil selisih di antara keduanya.) Gambar 10.20 berisi contoh, yang pada dasarnya sudah jelas; Gambar 10.20 menunjukkan Rentang Kriteria yang diperluas yang dapat digunakan dalam kasus ini.

**Contoh: Menerapkan Beberapa Kueri**

Saat menggunakan fungsi Basis Data, Rentang Kriteria terdiri dari judul bidang dan satu atau beberapa baris yang bersebelahan di bawahnya. Jadi, jika diperlukan serangkaian kriteria kedua, Rentang Kriteria baru akan diperlukan, dan jika banyak serangkaian kriteria tersebut diperlukan, akan ada beberapa rentang tersebut, sehingga ruang yang digunakan dalam buku kerja dapat menjadi sangat besar (yaitu setiap Rentang Kriteria akan memerlukan setidaknya tiga baris Excel judul, nilai kriteria, dan baris kosong sebelum Rentang Kriteria berikutnya).

Alternatifnya adalah memiliki satu Rentang Kriteria, dan menempatkan secara berurutan serangkaian nilai yang akan digunakan untuk setiap kueri dalam rentang ini, setiap kali juga mencatat hasil kueri (seolah-olah melakukan analisis sensitivitas). Sayangnya, proses seperti itu secara umum tidak dapat disederhanakan dengan menggunakan fungsi pencarian untuk menempatkan data dalam rentang kriteria: jika rentang kriteria berisi fungsi yang mengembalikan sel kosong, maka fungsi Basis Data tidak menganggap kekosongan yang dihitung tersebut sama seperti jika sel yang berisi nilai kriteria yang sama benar-benar kosong.

Dengan demikian, dalam praktiknya, proses semacam itu perlu dilakukan secara manual atau otomatis menggunakan makro VBA, karena penyalinan di Excel (atau penugasan di VBA) sel kosong ke dalam Rentang Kriteria menyebabkan nilainya diperlakukan sebagai benar-benar kosong.

G	H	I	J	K	L	M
Field	Criteria					
Amount £	Customer	Country	Amount £	Amount £	Due Date	
		UK	>£5000	<=£10000		

**Gambar 10.20 Rentang Kriteria Diperluas oleh Kolom untuk Menerapkan Kueri Between**

Field	Criteria	Amount £	Amount £	Due Date	DCOUNT	DSUM	DMIN	DMAX
	=IF(ISBLANK	<=£10000			0	0	0	0
	UK	<=£10000			10	72974	5096	9006
PASTED VALUES OF RESULTS:								
		<=£10000			43	304157	4095	9942
		>£10000			57	749131	10007	15851
	UK	<=£10000			10	72974	5096	9006
	UK	>£5000	<=£10000					
	Germany	<=£10000						
	Germany	>£5000	<=£10000					
	Italy	<=£10000						
	Italy	>£5000	<=£10000					
	France	<=£10000						
	France	>£5000	<=£10000					
	Spain	<=£10000						
	Spain	>£5000	<=£10000					

**Gambar 10.21 Fungsi Basis Data Tidak Memperlakukan Blank yang Dihitung sebagai Blank yang Sebenarnya**

Gambar 10.21 menunjukkan contoh-contohnya. Gambar 10.21 menunjukkan bahwa fungsi Basis Data akan mengembalikan nol (sel O5:R5) saat proses pencarian digunakan untuk menemukan nilai yang diinginkan untuk digunakan dalam rentang kriteria, bahkan saat langkah tambahan dibuat, yaitu saat kosong dalam rentang input dirujuk, nilai yang dikembalikan juga ditetapkan menjadi kosong, bukan sebagai nol; lihat Bilah Rumus. Karena kosong terhitung dalam Rentang Kriteria tidak diperlakukan seolah-olah itu adalah sel kosong sederhana, tidak ada rekaman yang cocok ditemukan.

Namun, saat kueri ditempelkan satu per satu ke dalam Rentang Kriteria (untuk tujuan demonstrasi, rentang baru telah disiapkan di sel I8:M9), tidak ada masalah seperti itu yang muncul. Seseorang dapat mengulangi proses untuk menyalin kueri individual dan setiap kali menempelkan hasilnya ke dalam rentang penyimpanan; dalam Gambar, tiga set kriteria pertama telah diselesaikan, dengan hasil ditempelkan di sel O13:O15). Jelas, proses semacam itu dapat diotomatisasi menggunakan makro VBA.

## 10.6 APLIKASI PRAKTIS: PIVOTTABLE

PivotTable dapat digunakan untuk menghasilkan laporan tabulasi silang yang meringkas berbagai aspek basis data menurut kategori. Keunggulan utamanya meliputi:

- PivotTable memungkinkan pembuatan laporan dengan sangat cepat, untuk mengubah identitas kriteria, dan untuk melakukan "drill-down" atau analisis data yang lebih terperinci dengan mudah dan cepat.
- Penyajian laporan dapat dengan mudah diubah, misalnya mengganti baris dan kolom



atau melaporkan rata-rata alih-alih total. Lebih jauh, filter dan "slicer" tambahan dapat ditambahkan atau dihapus, yang dapat memungkinkan eksplorasi dan analisis yang sangat cepat.

- PivotChart juga memungkinkan kesalahan atau nilai spesifik lainnya dalam data yang mendasarinya diabaikan (atau difilter) saat menerapkan kriteria atau filter.
- PivotChart mudah dibuat dari PivotTable.
- Tabel-tabel ini didukung oleh VBA, sehingga seseorang dapat merekam makro berdasarkan tindakan yang diterapkan pada PivotTable, dan menyesuaikannya sesuai kebutuhan.

Meskipun hebat, PivotTable memiliki beberapa properti yang berpotensi merugikan:

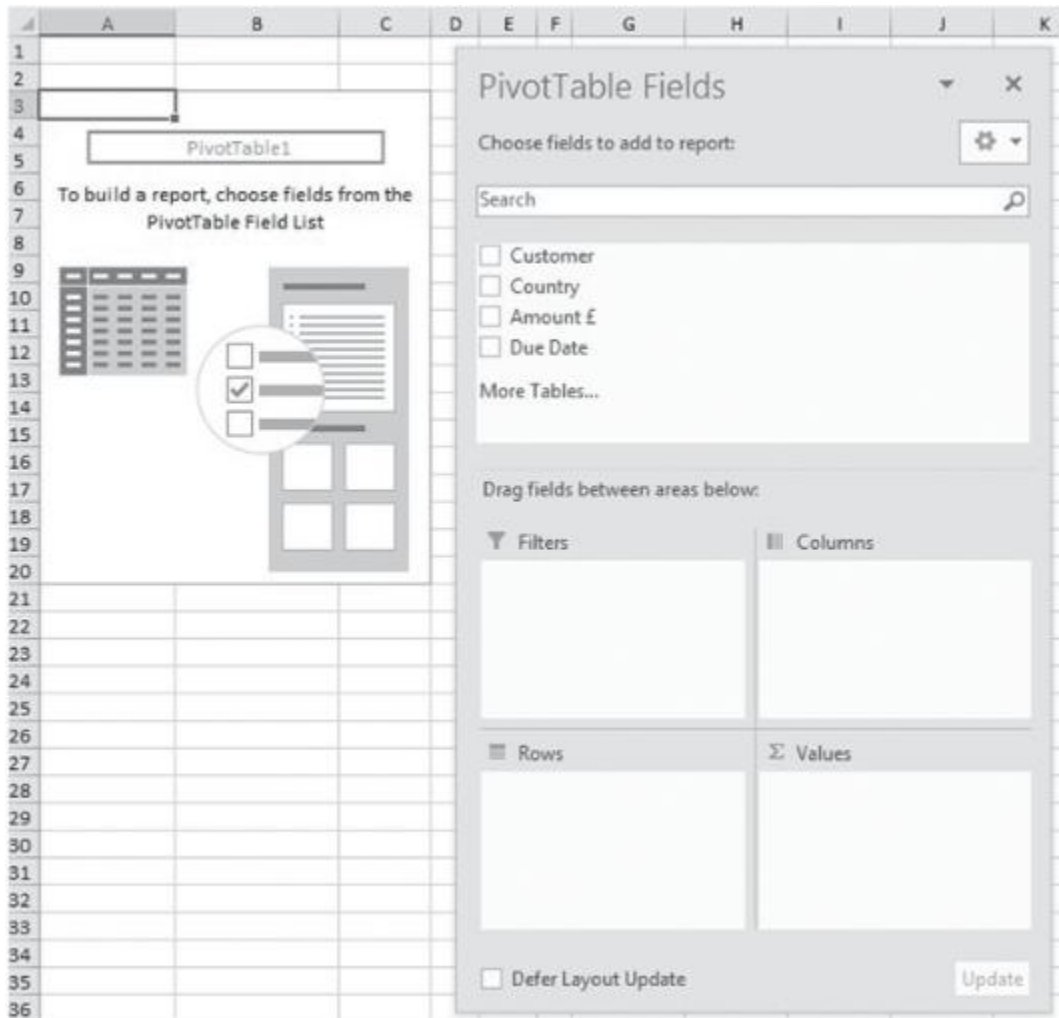
- Tabel-tabel ini tidak terhubung langsung ke set data, jadi perlu disegarkan jika data berubah. (Ini karena tabel-tabel ini menggunakan "cache" dari data asli. Faktanya, beberapa PivotTable dapat dibuat menggunakan data dasar yang sama, dan ini idealnya dilakukan dengan menyalin PivotTable pertama untuk memastikan bahwa data cache yang sama digunakan dan dengan demikian tidak mengurangi efisiensi komputasi saat menangani set data yang besar.)
- Tabel-tabel ini akan "rusak", dan perlu dibangun kembali, jika judul bidang berubah. Misalnya, PivotTable yang didasarkan pada judul bidang "Gaji 2016" perlu dibangun kembali jika - selama proses pembaruan tahunan - judul bidang berubah menjadi "Gaji 2017". Hal ini berbeda dengan fungsi Excel biasa (non-basis data) (yang hanya ditautkan ke data, dengan label bidang yang pada dasarnya hanya sebagai pelengkap), dan fungsi Basis Data (yang jika diterapkan dengan benar tidak akan rusak jika judul bidang berubah, atau setidaknya hanya memerlukan pengerjaan ulang minimal). Jadi, fungsi-fungsi tersebut mungkin bukan metode pelaporan yang paling efektif untuk kueri kompleks yang akan diperbarui secara berkala.
- Karena struktur PivotTable (misalnya tata letak baris-kolom, jumlah baris, dsb.) bergantung pada data yang mendasarinya, maka secara umum tidak tepat untuk menggunakannya sebagai langkah perhitungan sementara (misalnya rumus Excel secara umum harus menghindari referensi sel dalam PivotTable). Oleh karena itu, PivotTable paling baik digunakan baik untuk analisis eksplorasi umum, maupun sebagai bentuk laporan akhir.

### **Contoh: Menjelajahi Nilai Ringkasan Kumpulan Data**

Gambar 10.22 menunjukkan contoh PivotTable sederhana. Basis data yang mendasarinya telah ditetapkan sebagai Tabel Excel yang disebut Data-Set1 (meskipun hal itu tidak diperlukan untuk membuat PivotTable). PivotTable dibuat dengan memilih sel mana pun dalam basis data dan menggunakan Sisipkan/PivotTable, lalu mengisi data atau pilihan yang diperlukan oleh kotak dialog.

Saat melakukannya, seseorang secara umum dapat memilih untuk menempatkan

PivotTable pada lembar kerja baru untuk menjaga data tetap terpisah dari analisis, dan untuk memungkinkan adaptasi kumpulan data (penyisipan baris baru) tanpa berpotensi menimbulkan konflik. Gambar 10.22 menunjukkan struktur dasar yang dibuat secara otomatis. Setelah itu, pengguna dapat memilih nama bidang menggunakan kotak centang dan menyeretnya ke area Baris, Kolom, Nilai, atau Filter, sesuai keinginan. Gambar 26.23 menunjukkan contoh laporan yang merangkum angka-angka menurut negara dan pelanggan.



**Gambar 10.22 Struktur Dasar PivotTable**

Perhatikan bahwa laporan akan langsung ditransposisikan jika struktur baris-kolom item negara dan pelanggan ditukar. Gambar 10.24 menunjukkan bagaimana seseorang dapat memilih untuk menggunakan pendekatan filter untuk menyajikan hasil. Dalam contoh ini, bidang Negara dipindahkan dari area Kolom ke area Filter sehingga filter baru muncul di Baris 1; ini telah diterapkan untuk hanya menampilkan data yang berkaitan dengan Jerman.

**Beberapa poin penting tambahan meliputi:**

- Item dapat disusun ulang dalam presentasi menggunakan opsi Urutkan, atau secara manual dengan mengklik kanan pada item dan menggunakan menu peka konteks

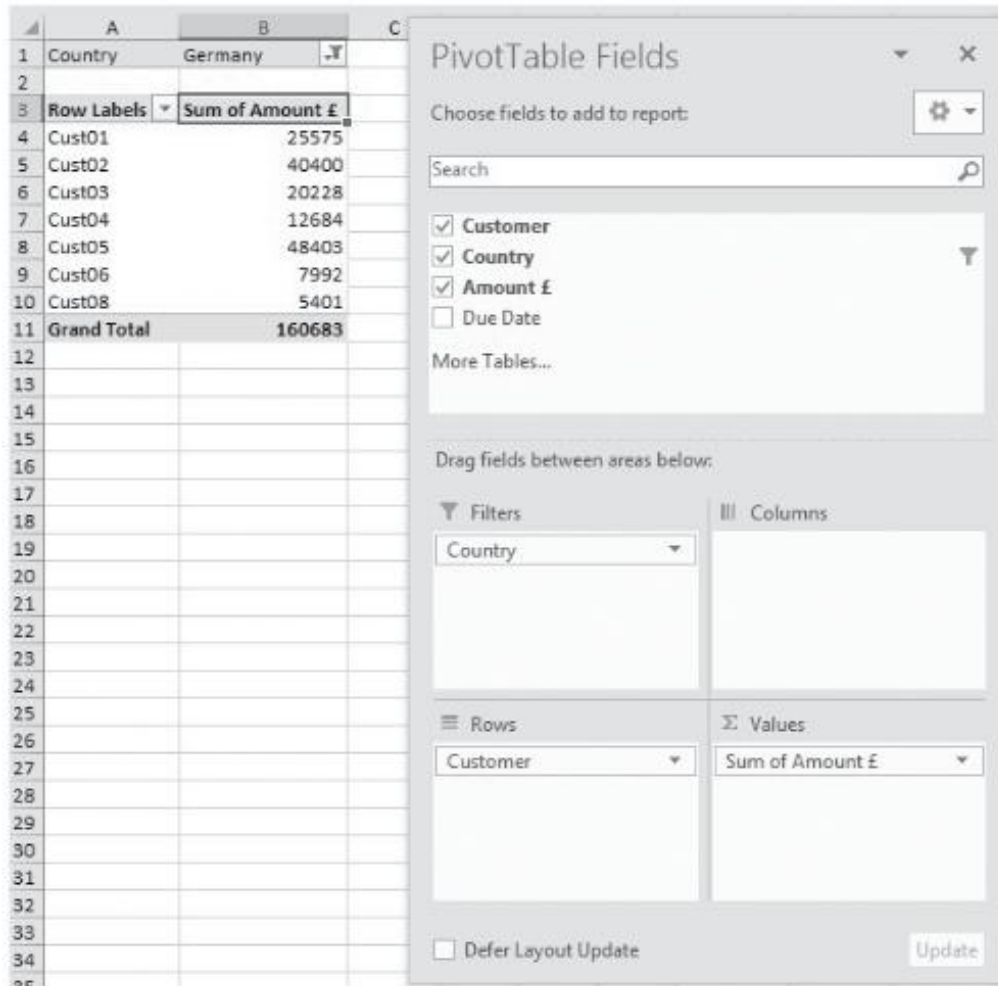
(Pindahkan), atau cukup dengan mengklik batas sel yang berisi label dan menyeret sel ini ke lokasi yang diinginkan.

- Tab Alat/Analisis PivotTable menyediakan berbagai macam opsi, termasuk penggunaan Pengaturan Bidang untuk menampilkan kalkulasi selain jumlah total, serta penggunaan kalkulasi yang disesuaikan.

Row Labels	France	Germany	Italy	Spain	UK	Grand Total
Cust01	19383	25575		4486		49444
Cust02	9386	40400	4449	15814	42662	112711
Cust03	11117	20228	12759	68569	37917	150590
Cust04	46494	12684	7768	20457		87403
Cust05	10544	48403	60097	5457	23012	147513
Cust06	13594	7992	42519	19950	22211	106266
Cust07	15264		25587	16446	59372	116699
Cust08	29188	5401	13933	36627	5777	90926
Cust09	25022		15712	13232		53966
Cust10			60294	27710	49796	137800
<b>Grand Total</b>	<b>179992</b>	<b>160683</b>	<b>243118</b>	<b>228748</b>	<b>240747</b>	<b>1053288</b>

**Gambar 10.23 Pembuatan Baris-kolom PivotTable dan Struktur Pelaporan**

- PivotTable kedua (berdasarkan cache data yang sama) dapat dibuat baik dengan menggunakan opsi Tabel Lainnya dalam dialog Bidang PivotTable, atau dengan memilih Seluruh PivotTable dalam opsi Pilih pada tab Alat/Analisis PivotTable, lalu menggunakan operasi Salin/Tempel standar. PivotTable dapat dihapus dengan proses pemilihan yang sama, lalu menggunakan menu Clear Contents biasa di Excel. Gambar 10.25 menunjukkan contoh penerapan beberapa poin ini: PivotTable kedua telah disisipkan, Filter digunakan sebagai ganti struktur kolom, dan urutan negara telah diubah secara manual. Perhatikan bahwa dialog PivotTable Fields hanya muncul satu kali, tetapi khusus untuk PivotTable yang aktif.



**Gambar 10.24 Penggunaan Dasar Filter**

### Contoh: Menjelajahi Elemen Dasar Item Ringkasan

Fitur penting PivotTable adalah kemampuan untuk melihat item individual yang menyusun gambar ringkasan dengan cepat, yaitu untuk mencantumkan catatan individual terperinci yang relevan dalam basis data dasar. Seseorang dapat membuat daftar tersebut dengan:

- Mengklik dua kali pada sel di PivotTable.
- Mengklik kanan dan menggunakan Tampilkan Detail.

Gambar 10.26 menunjukkan hasil penerapan proses ini ke Sel B23 (data ringkasan untuk Prancis) dalam contoh di atas (ditunjukkan dalam lembar kerja DrillDown dari file contoh).

### Contoh: Menambahkan Slicer

Dalam bentuk dasarnya, Slicer mirip dengan Filter, tetapi hanya menyediakan tampilan visual yang lebih baik, serta lebih fleksibel untuk bereksperimen dengan opsi dengan cepat.

Customer	Country	Amount £	Due Date
Cust01	France	19383	
Cust02	France	9386	
Cust03	France	11117	
Cust04	France	46494	
Cust05	France	10544	
Cust06	France	13594	
Cust07	France	15264	
Cust08	France	29188	
Cust09	France	25022	
Cust10	France		
Grand Total		179992	

Gambar 10.25 Representasi Laporan Berdasarkan PivotTable yang Disalin

Customer	Country	Amount £	Due Date
Cust03	France	11117	07/08/2017
Cust02	France	9386	23/07/2017
Cust08	France	13337	22/07/2017
Cust07	France	15264	29/05/2017
Cust04	France	8548	04/06/2017
Cust09	France	9942	01/08/2017
Cust04	France	12790	04/07/2017
Cust01	France	5853	05/07/2017
Cust04	France	6509	16/05/2017
Cust01	France	13530	17/06/2017
Cust08	France	15851	14/07/2017
Cust06	France	13594	23/05/2017
Cust09	France	15080	07/04/2017
Cust04	France	9547	01/06/2017
Cust04	France	9100	06/06/2017
Cust05	France	10544	08/04/2017

Gambar 10.26 Contoh Hasil Drill-down Item dalam PivotTable

Row Labels	France	Germany	Italy	Spain	UK	Grand Total
Cust01	19383	25575			4486	49444
Cust02	9386	40400	4449	15814	42662	112711
Cust03	11117	20228	12759	68569	37917	150590
Cust04	46494	12684	7768	20457		87403
Cust05	10544	48403	60097	5457	23012	147513
Cust06	13594	7992	42519	19990	22211	106266
Cust07	15264		25587	16446	59372	116669
Cust08	29188	5401	13933	36627	5777	90926
Cust09	25022		15712	13232		53966
Cust10			60294	27710	49796	137800
<b>Grand Total</b>	<b>179992</b>	<b>160683</b>	<b>243118</b>	<b>2E+05</b>	<b>240747</b>	<b>1E+06</b>

Country	Sum of Amount £
Germany	160683
France	179992
Italy	243118
<b>Grand Total</b>	<b>583793</b>

Gambar 26.27 Penggunaan Slicer Tunggal

Slicer dapat dibuat menggunakan menu pada tab PivotTable Tools/Analyze. Gambar 10.27 menunjukkan contoh penambahan Slicer pada kolom Country dari PivotTable kedua, lalu menampilkan hasil ringkasan untuk kumpulan negara yang dipilih (tahan tombol Ctrl untuk memilih beberapa item). Pada Gambar 10.28, kami menampilkan hasil penyertaan Slicer kedua (berdasarkan Pelanggan), di mana semua item untuk Slicer kedua dipilih.

Gambar 10.29 menunjukkan hasil penggunaan Slicer kedua untuk melaporkan hanya item yang terkait dengan pelanggan Cust01; terlihat bahwa Slicer Country menyesuaikan untuk menampilkan bahwa pelanggan tersebut tidak ada di Italia. Hal ini menyoroti salah satu keuntungan utama pemotong dibanding pendekatan filter (yaitu mencerminkan interaksi antara subset data dengan benar).

#### Contoh: Pemotong Garis Waktu

Pemotong Garis Waktu hanyalah bentuk pemotong khusus yang dapat digunakan (disisipkan menggunakan tab Alat/Analisis Tabel Pivot) saat kolom tanggal akan dianalisis. Daripada harus menyesuaikan data (seperti dengan menggunakan fungsi BULAN atau TAHUN), pemotong secara otomatis menyediakan opsi untuk menganalisis data menurut berbagai tingkat ketelitian waktu. Gambar 10.30 menunjukkan hasil penyisipan Pemotong Garis Waktu (di mana daftar Pelanggan lengkap digunakan), dengan menu tarik-turun pemotong diatur ke opsi "Bulan".

17		
18	Customer	(All)
19		
20	Row Labels	Sum of Amount £
21	Germany	160683
22	France	179992
23	Italy	243118
24	Grand Total	583793
25		
26	Customer	
27	Cust01	
28	Cust02	
29	Cust03	
30	Cust04	
31	Cust05	
32	Cust06	
33	Cust07	
34	Cust08	
35		
36		
37		
38		
39		
40		

Gambar 10.28 Penyertaan Slicer Kedua

18	Customer	Cust01
19		
20	Row Labels	Sum of Amount £
21	Germany	25575
22	France	19383
23	Grand Total	44958
24		
25		
26	Customer	
27	Cust01	
28	Cust02	
29	Cust03	
30	Cust04	
31	Cust05	
32	Cust06	
33	Cust07	
34	Cust08	
35		
36		
37		
38		
39		

Gambar 10.29 Interaksi Antar Slicer

Row Labels	Sum of Amount £
Germany	35723
France	21059
Italy	32430
<b>Grand Total</b>	<b>89212</b>

Gambar 10.30 Penggunaan Slicer Timeline

#### Contoh: Membuat Laporan yang Mengabaikan Kesalahan atau Item Tertentu Lainnya

Salah satu fitur PivotTable yang bermanfaat adalah kemampuan untuk dengan cepat menyertakan atau mengecualikan data dari laporan ringkasan (menggunakan filter atau slicer). Misalnya, kolom komentar dalam kumpulan data dapat menunjukkan bahwa beberapa item perlu diperlakukan dengan cara khusus. Gambar 10.31 menunjukkan sebuah contoh. Gambar 10.31 menunjukkan sebuah basis data yang di dalamnya terdapat kolom komentar.

Gambar 10.32 menunjukkan sebuah PivotTable yang di dalamnya terdapat Slicer, dengan analisis yang dimaksudkan untuk hanya menyertakan item-item di dalam kolom komentar yang kosong atau ditandai "OK" atau ditandai "Dicentang". Perhatikan bahwa jika seseorang akan melakukan analisis ringkasan tersebut menggunakan fungsi (basis data atau non-basis data), penanganan khusus terhadap item yang "dikoreksi" tersebut akan jauh lebih rumit: misalnya, seseorang biasanya perlu membuat bidang khusus dalam basis data yang bertindak sebagai kriteria untuk menentukan apakah item akan disertakan atau tidak dalam perhitungan. Namun, karena komentar yang dibuat pada dasarnya dapat berupa apa saja, akan sangat rumit, memakan waktu, dan rawan kesalahan untuk membuat rumus yang berlaku.

#### Contoh: Menggunakan Fungsi GETPIVOTDATA

Seseorang dapat memperoleh (atau mencari) nilai dalam PivotTable yang terlihat menggunakan fungsi GETPIVOT-DATA. Fungsi tersebut memiliki sejumlah parameter:

- Nama bidang data (diapit tanda kutip).
- Argumen Pivot\_table, yang harus menyediakan referensi ke sel mana pun di PivotTable tempat data ingin diambil (atau rentang sel, atau rentang bernama).



- Nama bidang dan item: ini adalah argumen opsional, dan mungkin ada beberapa, tergantung pada tingkat ketelitian dan jumlah variabel di PivotTable.

	A	B	C	D	E	F	G	H
1								
2								
3								
4		Customer	Country	Amount £	Due Date	Comments		
5		Cust02	UK	12232	20-Mar-17	OK		
6		Cust06	Italy	4749	16-Mar-17	exclude until have checked with Bob		
7		Cust07	Italy	7282	12-Apr-17	OK		
8		Cust03	Italy	12759	14-Jun-17	OK		
9		Cust10	UK	12334	24-May-17	OK		
10		Cust05	Italy	4283	24-Mar-17	Should be left out		
11		Cust06	Germany	7992	5-May-17	OK		
12		Cust06	Italy	13202	16-Apr-17			
13		Cust04	Germany	12684	4-Jun-17	Checked 31 August		
14		Cust10	UK	11862	13-Jun-17			
15		Cust10	Italy	13630	21-May-17	unsure if data is correct		
16		Cust07	UK	14593	20-Jan-17			
17		Cust07	Italy	4394	4-May-17			
18		Cust09	Italy	15712	8-Apr-17			
19		Cust10	UK	6503	28-Mar-17			
20		Cust05	France	10544	8-Apr-17			
21		Cust02	Germany	9274	17-Jun-17			

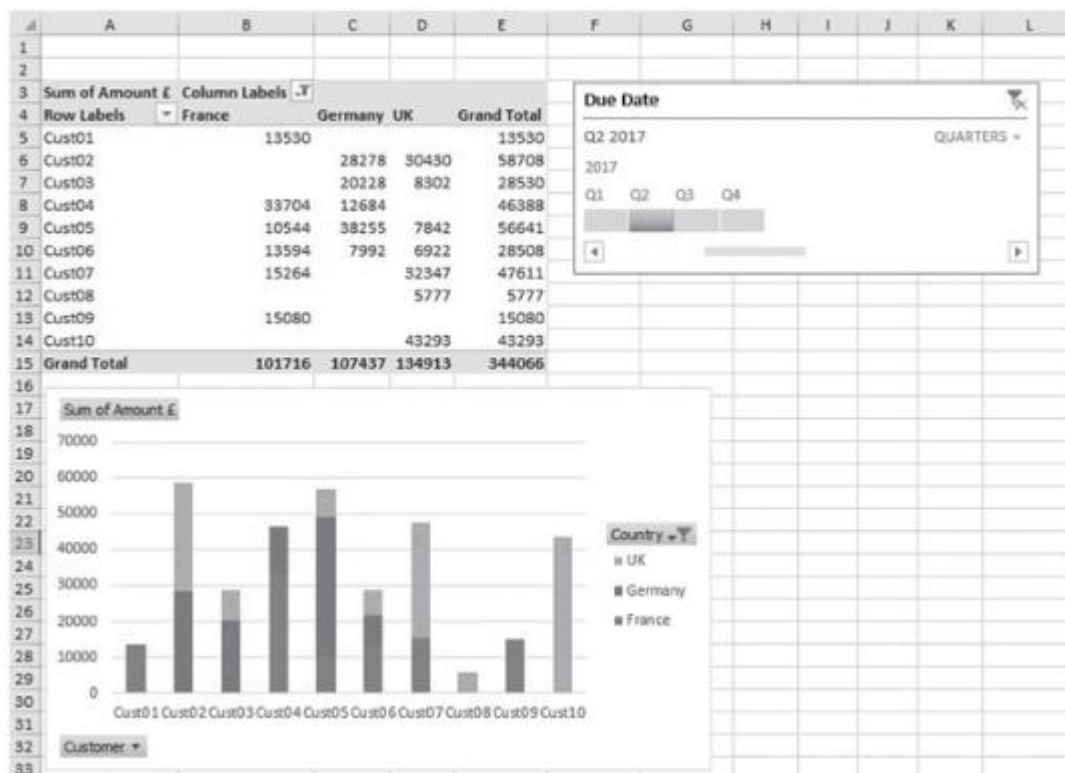
Gambar 10.31 Kumpulan Data dengan Kolom Komentar

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29
			Sum of Amount £	Column Labels																									
		Row Labels	France	Germany	Italy	Spain	UK	Grand Total																					
		Cust01	19383	25575	4449	15814	42662	112711																					
		Cust02	9386	40400	4449	15814	42662	112711																					
		Cust03	11117	20228	12759	68569	37917	150590																					
		Cust04	46494	12684	7768	20457		87403																					
		Cust05	10544	48403	55814	5457	23012	143230																					
		Cust06	13594	7992	37770	19950	22211	101517																					
		Cust07	15264		25587	16446	59372	116669																					
		Cust08	29188	5401	13933	36627	5777	90926																					
		Cust09	25022		15712	13232		53966																					
		Cust10			46664	27710	49796	124170																					
		Grand Total	179992	160683	220456	228748	240747	1030626																					
		Comments																											
		Checked 31 August																											
		exclude until have checke...																											
		OK																											
		Should be left out																											
		unsure if data is correct																											
		(blank)																											

Gambar 10.32 Menggunakan Slicer untuk Mengecualikan Item Tertentu yang Dikomentari

	A	B	C	D	E	F	G	H	I
1									
2									
3		Customer	Cust06						
4		Result	106266	=GETPIVOTDATA("Amount £",\$B\$12,B3,C3)					
5									
6									
7		Customer	Cust06						
8		Country	Italy						
9		Result	42519	=GETPIVOTDATA("Amount £",\$B\$12,B7,C7,B8,C8)					
10									
11									
12		Sum of Amount £	Column Labels						
13		Row Labels	France	Germany	Italy	Spain	UK	Grand Total	
14		Cust01	19383	25575		4486		49444	
15		Cust02	9386	40400	4449	15814	42662	112711	
16		Cust03	11117	20228	12759	68569	37917	150590	
17		Cust04	46494	12684	7768	20457		87403	
18		Cust05	10544	48403	60097	5457	23012	147513	
19		Cust06	13594	7992	42519	19950	22211	106266	
20		Cust07	15264		25587	16446	59372	116669	
21		Cust08	29188	5401	13933	36627	5777	90926	
22		Cust09	25022		15712	13232		53966	
23		Cust10			60294	27710	49796	137800	
24		Grand Total	179992	160683	243118	228748	240747	1053288	
25									
26									
27									
28									

Gambar 10.33 Penggunaan Fungsi GETPIVOTDATA



Gambar 10.34 Contoh Bagan Pivot dan Penyelarasan Filter dengan Tabel Pivotnya

Gambar 10.35 berisi contoh, yang ditunjukkan pada Gambar 26.33. Karena data harus sudah

terlihat, orang mungkin bertanya tentang manfaatnya. Satu kasus yang mungkin berguna adalah jika ada banyak PivotTable besar, tetapi orang hanya menginginkan data terpilih dari data tersebut. Dalam kasus seperti itu, mungkin fungsi seperti SUMIFS akan lebih sesuai (yang mungkin sering terjadi, kecuali fungsi filter dan slicer diperlukan secara bersamaan).

### Contoh: Membuat PivotChart

Setelah PivotTable dibuat, PivotChart juga dapat ditambahkan menggunakan ikon PivotChart pada tab Analisis di Alat PivotTable. Kedua objek tersebut kemudian terhubung secara inheren: penerapan filter pada satu objek akan menyesuaikan filter terkait pada objek lainnya, dan penyisipan pemotong akan memengaruhi keduanya.

	A	B	C	D	E	F	G	H	I	J	K
1		Table 1: Unique IDS, full list		Table2: Revenues						Table3: Costs	
2											
3		CUSTIDFULL		Custid	Revs	Date	Month		Custid	Costs	
4		Cust1		Cust1	100	01/01/2013	1		Cust1	80	
5		Cust10		Cust10	90	01/02/2013	2		Cust10	70	
6		Cust100		Cust100	80	07/03/2013	3		Cust100	60	
7		Cust101		Cust101	75	06/02/2013	2		Cust1	50	
8		Cust102		Cust102	70	10/02/2013	2		Cust102	40	
9		Cust103		Cust103	120	16/02/2013	2		Cust103	30	
10		Cust104		Cust104	150	29/03/2013	3		Cust104	75	
11		Cust105		Cust105	92	30/04/2013	4		Cust105	75	
12		Cust106		Cust106	73	03/04/2013	4		Cust106	75	
13		Cust107		Cust1	200	29/04/2013	4		Cust107	75	
14		Cust108		Cust101	150	24/03/2013	3				
15		Cust109		Cust102	50	12/02/2013	2				
16		Cust112									
17		Cust113									
18		Cust115									
19		Cust117									
20		Cust119									
21		Cust12									
22		Cust120									
23		Cust122									
24		Cust123									
25		Cust129									
26		Cust133									
27		Cust135									
28		Cust136									
29		Cust138									
30		Cust139									
31		Cust141									
32		Cust142									

**Gambar 10.35** Kumpulan Data yang Digunakan untuk Contoh PowerView

Gambar 10.34 berisi sebuah contoh. Gambar 10.34 memperlihatkan PivotChart yang menerapkan filter negara, dan format kolom PivotTable menyesuaikan secara otomatis. Pemotong Garis Waktu yang telah disisipkan untuk PivotTable juga memengaruhi tampilan yang ditampilkan pada bagan.

### Contoh: Menggunakan Model Data Excel untuk Menautkan Tabel

Jika seseorang ingin membuat laporan berdasarkan beberapa basis data atau tabel,

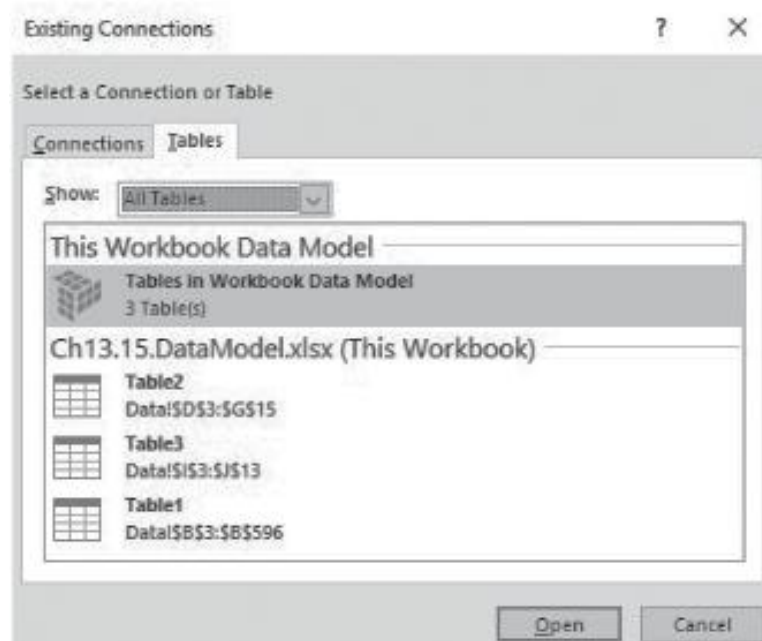
mungkin akan merepotkan jika harus menggabungkannya menjadi satu basis data. Model Data Excel memungkinkan analisis PivotTable dari beberapa set data dengan menautkannya melalui hubungan (tentu saja, seseorang tetap perlu memiliki, atau membuat, beberapa bentuk pengenal atau kunci yang memungkinkan proses pencocokan berlangsung melalui hubungan tersebut, tetapi fungsi pencarian yang merepotkan dan menghabiskan banyak memori sering kali dapat dihindari). Gambar 10.35 menunjukkan sebuah contoh. Gambar 10.35 menunjukkan tiga set data yang telah ditetapkan sebagai Tabel Excel, dengan yang pertama menyediakan daftar lengkap pengenal unik untuk setiap pelanggan, yang kedua menyediakan informasi tentang pendapatan dan tanggal, dan yang ketiga menyediakan informasi biaya.

Untuk membuat laporan PivotTable yang menggunakan informasi dari tabel-tabel ini (tanpa secara eksplisit menggabungkannya menjadi satu set data), langkah-langkah berikut diperlukan:

- Tetapkan setiap set data sebagai Tabel Excel (lihat sebelumnya).
- Pada tab Data, gunakan ikon Connections untuk menambahkan setiap Tabel secara bergantian ke DataModel. Ini dilakukan dengan menggunakan menu drop-down dari menu Add untuk menemukan AddtotheDataModel (lihat Gambar 10.36), dan memilih tab Tables. Setelah selesai, daftar koneksi akan ditampilkan seperti pada Gambar 10.37.



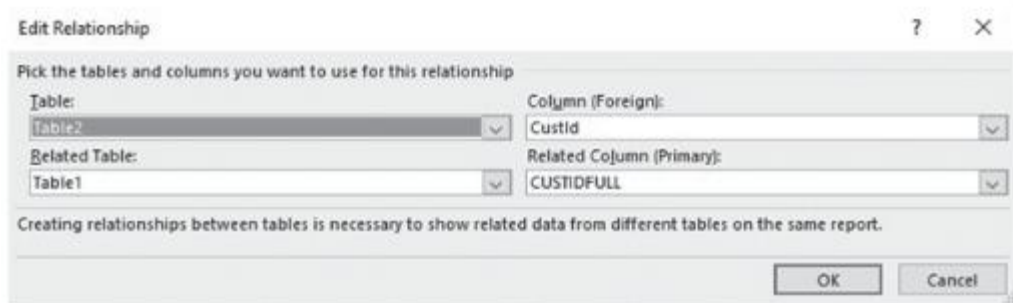
**Gambar 10.36 Menambahkan Koneksi Buku Kerja Baru I**



**Gambar 10.37 Daftar Koneksi yang Lengkap**

- Tentukan hubungan menggunakan Data/Relationships/New. Dalam contoh ini, kolom

- CUSTIDFULL dari tabel pertama bertindak sebagai Kunci Utama (lihat Gambar 10.38).
- Terakhir, buat PivotTable menggunakan Insert/PivotTable. Saat melakukannya, Anda perlu mencentang “Use an external data source” di dalam kotak dialog, dan memilih “Tables in Workbook Data Model”. Gambar 26.39 menunjukkan contoh laporan.



**Gambar 10.38 Menetapkan Hubungan Antar Tabel**

Row Labels	Sum of Revs	Sum of Costs
Cust1	300	130
Cust10	90	70
Cust100	80	60
Cust101	225	
Cust102	120	40
Cust103	120	30
Cust104	150	75
Cust105	92	75
Cust106	73	75
Cust107		75
<b>Grand Total</b>	<b>1250</b>	<b>630</b>

**Gambar 10.39 Laporan PivotTable Berdasarkan DataModel**

## BAB 11

### PINTASAN TERPILIH DAN FITUR LAINNYA

#### 11.1 PENDAHULUAN

Bab ini menyajikan daftar pintasan utama yang menurut penulis paling berguna. Sebagian besar pintasan ini sudah jelas setelah dipraktikkan beberapa kali, sehingga hanya beberapa klip layar yang ditampilkan untuk menunjukkan contoh-contoh spesifik. Penguasaan pintasan dapat meningkatkan efisiensi dan kecepatan kerja. Oleh karena itu, setiap pemodel harus mengembangkan keakraban yang baik dengan pintasan yang paling relevan bagi mereka, dan memiliki pemahaman umum tentang rangkaian pintasan yang lebih luas yang tersedia jika pintasan tersebut berguna dalam keadaan tertentu.

Perlu dicatat bahwa beberapa pintasan menjadi sangat relevan saat merekam kode VBA, karena pintasan tersebut memungkinkan untuk menangkap kode untuk operasi yang tidak akan terlihat jelas. Tentu saja ada pintasan yang tersedia di Excel yang tidak tercakup dalam bab ini, dan pembaca dapat melakukan penelitiannya sendiri dalam hal ini, mungkin menggunakan daftar dalam bab ini sebagai daftar periksa awal. Bab ini juga secara singkat menyebutkan beberapa Tips Tombol Excel, serta beberapa alat lain yang berpotensi berguna, seperti Sparklines dan Kamera.

#### 11.2 PINTASAN UTAMA DAN PENGGUNAANNYA

Pintasan utama yang kami bahas terstruktur ke dalam kategori utama penerapannya:

- Memasukkan dan mengubah data dan rumus,
- Pemformatan,
- Audit, navigasi, dan item lainnya.

Perlu dicatat bahwa, pada papan ketik Inggris, beberapa pintasan ini (seperti Ctrl+&) mengharuskan seseorang untuk menggunakan tombol Shift untuk mengakses simbol yang relevan (seperti &); dalam kasus seperti itu Shift tidak dianggap sebagai bagian dari pintasan.

##### **Memasukkan dan Memodifikasi Data dan Rumus**

Pintasan inti untuk menyalin nilai dan rumus, serta mengoreksi entri, meliputi:

- Ctrl+C untuk menyalin sel atau rentang, Ctrl+X untuk memotong, Ctrl+V untuk menempel.
- Ctrl+Z untuk membatalkan.
- Saat mengedit rumus di Bilah Rumus, F4 menerapkan referensi sel absolut (misalnya, menempatkan \$ sebelum referensi baris); penekanan berulang memungkinkan seseorang untuk menelusuri semua kemungkinan kombinasi (\$ sebelum baris dan kolom, dan penghapusan \$). Kemudahan penerapan pintasan dapat menyebabkan terciptanya rumus yang "dihargai terlalu mahal", yaitu rumus yang simbol \$-nya disisipkan sebelum referensi baris dan kolom, baik hanya satu atau yang lain yang

diperlukan; hal ini biasanya akan menghasilkan rumus yang tidak dapat disalin dengan benar ke tempat lain dalam model.

- Rumus (serta format selnya) dapat disalin dari satu sel ke dalam rentang dengan menggunakan Ctrl+C (untuk memulai prosedur penyalinan) lalu, dengan sel ini dipilih, salah satu dari berikut:
  - ✓ Menekan Shift dan secara bersamaan memilih sel terakhir dari rentang tempat rumus ingin disalin, diikuti oleh Ctrl+V.
  - ✓ Menekan F8 dan (setelah melepaskannya) memilih sel terakhir dari rentang tempat rumus ingin disalin, diikuti oleh Ctrl+V.
  - ✓ Menggunakan Ctrl+Shift+Panah untuk memilih rentang dari sel saat ini hingga akhir rentang searah panah, diikuti oleh Ctrl+V.
  - ✓ Rumus (tetapi bukan formatnya) dapat disalin dari satu sel ke dalam rentang dengan membuat rumus di sel pertama rentang, lalu menggunakan salah satu teknik yang disebutkan di atas untuk memilih rentang tempat rumus akan disalin, lalu – dengan menggunakan Bilah Rumus memasukkan Ctrl+Enter (atau Ctrl+Shift+Enter untuk rumus array). Hal ini khususnya berguna saat memodifikasi model yang sudah dibangun dan diformat sebagian, karena tidak menggantikan format yang telah ditetapkan dalam sel yang akan ditimpa. Gambar 27.1 memperlihatkan sebuah contoh, tepat sebelum menggunakan Ctrl+Enter, yaitu rumus telah dibuat dalam Bilah Rumus dan rentang telah dipilih.
  - ✓ Rumus dapat disalin ke semua baris "relevan" dalam kolom dengan mengklik dua kali di kanan bawah sel; jumlah baris yang disalin akan ditentukan oleh baris di Wilayah Saat Ini dari sel tersebut (yaitu dalam kasus yang paling sederhana, ke titik di mana kolom yang berdekatan juga memiliki konten).
  - ✓ Ctrl+' (apostrof) dapat digunakan untuk membuat ulang konten sel tepat di atasnya (daripada menyalin rumus yang kemudian secara umum akan mengubah referensi sel di dalamnya). Satu kasus di mana hal ini berguna adalah ketika seseorang ingin membagi rumus gabungan menjadi komponen-komponennya, memperlihatkan setiap komponen pada baris terpisah. Seseorang dapat mengulang rumus pada baris di bawah ini, dan menghapus komponen yang tidak diperlukan untuk mengisolasi elemen-elemen individual. Ini akan lebih cepat dan lebih kuat daripada mengetik ulang komponen-komponen individual. Sebagai contoh ilustrasi yang mudah dipahami, komponen-komponen individual dalam rumus seperti:

= *SUM('Pendapatan Keuangan'!P21,'Biaya lisensi'!P21'Biaya Berlangganan'!P21)*

dapat diisolasi dalam sel terpisah sebelum dijumlahkan. Saat membuat rumus, hal-hal berikut perlu diperhatikan:

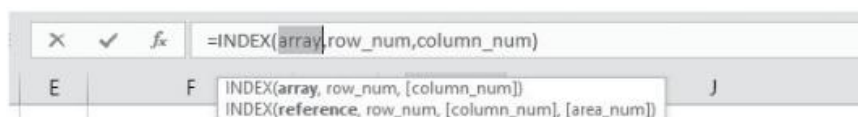
- Setelah mengetikkan nama fungsi yang valid terlebih dahulu (misalnya seperti

"MATCH"):

- Ctrl+Shift+A menampilkan nama argumen dan tanda kurung yang diperlukan untuk suatu fungsi, dan menempatkannya di dalam Bilah Rumus; contohnya ditampilkan pada Gambar 11.2.
- Ctrl+A memanggil dialog Sisipkan Fungsi, yang memungkinkan parameter fungsi dihubungkan ke sel dengan lebih mudah.
- Menetapkan dan menggunakan rentang bernama:
- Saat membuat rumus yang memerlukan referensi rentang yang telah diberi nama, menekan F3 saat berada di Bilah Rumus akan menyediakan menu nama yang dapat digunakan (misalnya nama buku kerja dan nama yang dicakup lembar kerja). Gambar 11.3 menunjukkan contoh rumus yang dibuat dengan menekan F3 untuk menampilkan daftar area bernama (dan memilihnya sebagai relevan).
- Ctrl+F3 dapat digunakan untuk memanggil Pengelola Nama guna menentukan rentang bernama.

Location	Customer no.	Date	Amount (local currency)	Currency	Xrate/€
Germany	C314	Jan	32,14	Euro	
London	C150	Jan	17,203	Pound	
France	C265	Feb	16,993	Euro	
US	C358	Feb	78,888	Dollar	
London	C879	Feb	43,957	Pound	
US	C323	Mar	25,898	Dollar	
London	C488	Mar	26,856	Pound	
London	C729	Mar	63,852	Dollar	
US	C288	April	28,188	Dollar	
Germany	C357	April	47,153	Euro	
France	C980	May	51,737	Euro	
London	C323	June	36,332	Pound	
Germany	C488	July	33,383	Euro	
Germany	C729	August	17,635	Euro	

Gambar 11.1 Menyalin Rumus Tanpa Mengubah Pemformatan Prasetel



Gambar 11.2 Memanggil Parameter Fungsi di Dalam Bilah Rumus Menggunakan Ctrl+Shift+A

Currency	Xrate/€
Dollar	1.60
Euro	1.20
Pound	1.00

Gambar 11.3 Membangun Rumus dengan Memanggil Daftar Rentang Bernama



- Ctrl+Shift+F3 dapat digunakan untuk membuat nama rentang dari label (meskipun penulis umumnya menyarankan untuk tidak melakukannya; meluangkan waktu untuk merenungkan struktur dan hierarki rentang bernama yang paling tepat, serta cakupannya, biasanya sangat penting, dan pemberian nama otomatis sering kali tidak akan menghasilkan serangkaian nama seperti itu).
- Ctrl+K akan menyisipkan hyperlink, menyertakan tautan ke rentang bernama, dan dapat menjadi alat navigasi dan dokumentasi model yang berguna.
- Shift+F3 memanggil dialog Sisipkan Fungsi (Fungsi Tempel) (bisa dibilang, melakukannya tidak lebih cepat daripada sekadar mengeklik tombol secara langsung).

### 11.3 PEMFORMATAN

Penggunaan pemformatan yang tepat dapat meningkatkan transparansi model secara drastis, dengan menyorot alurnya. Sayangnya, pemformatan yang buruk merupakan kelemahan utama banyak model. Sebagian, hal ini disebabkan oleh kurangnya disiplin dan waktu. Oleh karena itu, pengetahuan dan penggunaan beberapa pintasan terkait format utama dapat membantu meningkatkan model dengan cepat (bahkan saat kalkulasi yang mendasarinya tidak berubah). Elemen utama dalam hal ini adalah:

- Menggunakan Ctrl+\* (atau Ctrl+Shift+Spasi) untuk memilih Wilayah Saat Ini dari sebuah sel (yaitu untuk memilih rentang sel yang mungkin memerlukan pemformatan dengan cara yang sama atau memiliki batas yang ditempatkan di sekelilingnya) (lihat Gambar 11.4) untuk contoh di mana Wilayah Saat Ini dari sebuah sel (D13) dipilih dengan cara ini.
- Ctrl+1 untuk menampilkan menu Format Sel utama.
- Untuk bekerja dengan teks dalam sel atau rentang:
- Ctrl+2 (atau Ctrl+B) untuk menerapkan atau menghapus format tebal.
- Ctrl+3 (atau Ctrl+I) untuk menerapkan atau menghapus format miring.
- Ctrl+4 (atau Ctrl+U) untuk menerapkan atau menghapus garis bawah.

	A	B	C	D	E	F
1						
2		<b>Location</b>	<b>Customer no.</b>	<b>Date</b>	<b>Amount</b>	<b>Currency</b>
3		Germany	C314	Jan	32,143	Euro
4		London	C159	Jan	17,203	Pound
5		France	C265	Feb	16,993	Euro
6		US	C358	Feb	78,888	Dollar
7		London	C979	Feb	43,957	Pound
8		US	C323	Mar	25,898	Dollar
9		London	C486	Mar	26,856	Pound
10		London	C729	Mar	63,652	Dollar
11		US	C286	April	29,188	Dollar
12		Germany	C357	April	47,153	Euro
13		France	C989	May	51,737	Euro
14		London	C323	June	36,332	Pound
15		Germany	C486	July	33,363	Euro
16		Germany	C729	August	17,635	Euro
17						

**Gambar 11.4 Memilih Wilayah Sel Saat Ini (Sel D13)**

- Penempatan batas yang cepat di sekitar rentang dapat difasilitasi oleh:
- Ctrl+& untuk menempatkan batas.
- Ctrl+ \_ untuk menghapus batas.
- Format Painter (pada tab Beranda) dapat digunakan untuk menyalin format satu sel atau rentang ke yang lain (klik dua kali pada ikon akan membuatnya tetap aktif, sehingga dapat diterapkan ke beberapa rentang secara berurutan, hingga dinonaktifkan dengan satu klik).
- Alt+Enter untuk menyisipkan jeda baris dalam sel saat mengetik label.
- Ctrl+Enter untuk menyalin rumus ke dalam rentang tanpa mengganggu format yang ada (lihat di atas).
- Ctrl+T atau Ctrl+L dapat digunakan untuk membuat Tabel Excel dari rentang (ini lebih dari sekadar operasi pemformatan murni, seperti yang dibahas dalam Bab 10).

### **Audit, Navigasi, dan Item Lainnya**

Audit cepat suatu model dapat difasilitasi dengan penggunaan berbagai pintasan, termasuk:

- Ctrl+' (tanda kutip kiri) untuk memperlihatkan rumus (setara dengan Rumus/Tampilkan Rumus), yang dapat digunakan untuk:
- Mencari masukan model yang tersembunyi.
- Mencari rumus yang tidak konsisten di seluruh rentang.
- Penelusuran ketergantungan:
- Ctrl+[ memilih sel-sel tersebut (pada lembar kerja mana pun) yang merupakan preseden langsung dari suatu rumus (lihat Gambar 11.5).
- Ctrl+Shift+{ memilih sel-sel tersebut (pada lembar kerja yang sama dengan rumus (e)) yang merupakan preseden langsung atau tidak langsung (misalnya langkah-langkah

perhitungan mundur).

- Ctrl+] memilih sel-sel tersebut (pada lembar kerja mana pun) yang merupakan dependen langsung dari suatu rumus (e).
- Ctrl+Shift+} memilih sel-sel tersebut (pada lembar kerja yang sama dengan rumus(e)) yang merupakan dependen langsung atau tidak langsung (yaitu langkah-langkah perhitungan maju).
- Saat menggunakan Rumus/Telusuri Preseden atau Telusuri Dependen, klik ganda pada panah dependensi akan memindahkan satu ke tempat yang sesuai.
- Shift+F5 (atau Ctrl+F) untuk menemukan item yang ditentukan. Setelah ini dilakukan, Shift+F4 dapat digunakan untuk menemukan kemunculan berikutnya dari elemen tersebut (daripada harus memanggil ulang menu Temukan penuh dan menentukan item lagi).

	A	B	C	D	E	F	G	H	I
1									
2		Currency	Xrate/€						
3		Dollar	1.60						
4		Euro	1.20						
5		Pound	1.00						
6									
7		Location	Customer no.	Date	Amount (local currency)	Currency	Xrate/€		
8		Germany	C314	Jan	32.143	Euro	1.20		

**Gambar 11.5 Penggunaan Pintasan untuk Menelusuri Preseden (sel I8)**

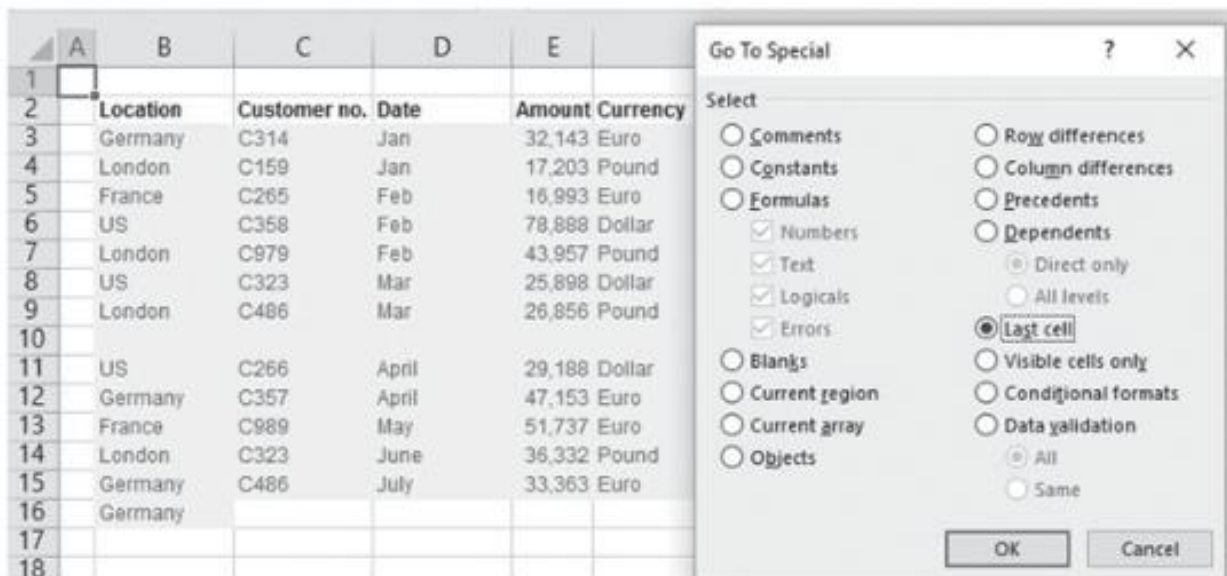
- F5 (atau Ctrl+G) untuk membuka sel, rentang, atau rentang bernama. Saat menggunakan ini, opsi Khusus dapat digunakan untuk menemukan rumus, konstanta, dan spasi. Dengan demikian, masukan model (saat berada di sel mandiri dalam lembar kerja yang sama) dapat ditemukan menggunakan F5/Khusus, memilih Konstanta (bukan Rumus), dan di bawah Rumus memilih Angka (serupa untuk bidang teks). Opsi "Sel terakhir" dari menu Khusus juga berguna, khususnya untuk merekam sintaks yang diperlukan saat mengotomatiskan manipulasi set data dengan VBA (lihat Gambar 11.6).
- F1 memanggil menu Bantuan.
- F2 memeriksa atau mengedit rumus di dalam sel (bukan di Bilah Rumus, yang biasanya memerlukan lebih banyak gerakan mata saat sering mengalihkan perhatian antara lembar kerja dan Bilah Rumus).

Bekerja dengan komentar:

- Shift+F2 menyisipkan komentar ke dalam sel aktif.
- Ctrl+Shift+O (huruf O) memilih semua sel dengan komentar (misalnya agar semuanya dapat ditemukan dan dibaca, atau sebagai alternatif semuanya dihapus secara

bersamaan).

- F3 akan menyediakan daftar rentang bernama yang valid dalam lembar kerja (yaitu rentang nama buku kerja dan rentang nama lembar kerja). Membuat daftar seperti itu tidak hanya berguna untuk audit dan dokumentasi, tetapi juga saat menulis kode VBA; penyalinan ke VBA (saat kode sedang ditulis) daftar nama dapat membantu memastikan ejaan yang benar dalam kode.
- F7 memeriksa ejaan.
- F9 menghitung ulang semua buku kerja yang terbuka.
- Alt+F8 memperlihatkan daftar makro dalam buku kerja.
- Alt+F11 membawa seseorang ke Editor VBA.
- F11 menyisipkan lembar bagan.



**Gambar 11.6 Menemukan Sel Terakhir dalam Rentang yang Digunakan**

Aspek lebih lanjut untuk berpindah di lembar kerja:

- Home untuk berpindah ke Kolom A dalam baris yang sama.
- Ctrl+Home untuk berpindah ke Sel A1.
- Ctrl+Shift+Home untuk memperluas pilihan ke Sel A1.
- Ctrl+Arrow berpindah ke sel pertama yang tidak kosong di baris atau kolom.
- Ctrl+Shift+Arrow memperluas pilihan ke sel terakhir yang tidak kosong di kolom atau baris.
- Shift+Arrow memperluas pilihan sel sebanyak satu sel.
- Ctrl+Shift+End memperluas pilihan ke sel terakhir yang digunakan pada lembar kerja.
- Ctrl+F1 beralih untuk menyembunyikan/memunculkan item menu Excel yang terperinci, dengan tetap mempertahankan nama tab menu dan Bilah Rumus. Ctrl+Shift+F1 berfungsi untuk mengubah item menu dan tab Excel menjadi

tersembunyi/tidak tersembunyi. Ini dapat berguna untuk meningkatkan tampilan file, misalnya saat melakukan presentasi ke grup.

#### 11.4 KEYTIPS EXCEL

KeyTips Excel diaktifkan dengan menggunakan tombol Alt, yang menghasilkan tampilan huruf yang dapat diketik untuk mengakses banyak bagian menu Excel. Beberapa contoh KeyTips penting meliputi:

- Alt-M-H untuk menampilkan rumus (setara dengan Ctrl+').
- Alt-M-P untuk melacak preseden.
- Alt-M-D untuk melacak dependen.
- Alt-A-V untuk menu validasi data.
- Alt-D-F-S untuk menghapus semua filter data.
- Alt-H-V-V untuk menempelkan nilai (Alt-E-S-V juga berfungsi). (Tanda “-” menunjukkan bahwa tombol digunakan secara terpisah dalam urutan, berbeda dengan pintasan tradisional, di mana tanda “+” menunjukkan bahwa tombol harus digunakan secara bersamaan.)

Pengaksesan KeyTips umumnya merupakan operasi visual yang sederhana dan jelas; ini berbeda dengan pintasan yang lebih tradisional, yang keberadaannya tidak jelas dengan sendirinya. Dengan demikian, pembaca dapat bereksperimen, dan menemukan pintasan yang mungkin paling berguna dalam bidang aplikasi mereka sendiri.

#### 11.5 ALAT DAN FITUR EXCEL LAIN YANG BERGUNA

##### Sparklines

Sparklines (Excel 2010 dan seterusnya) dapat digunakan untuk menampilkan serangkaian data dalam satu sel (seperti grafik mini). Ada tiga bentuk: grafik garis, kolom, dan menang/kalah. Grafik tersebut dapat dibuat dengan mengaktifkan lembar kerja tempat sparkline akan ditampilkan, menggunakan Sisipkan/Sparklines, dan memilih data yang relevan. (Rentang data mungkin berada pada lembar kerja yang berbeda dengan tempat grafik mini ditampilkan; jika grafik mini akan berada pada lembar yang sama dengan data, seseorang dapat memilih data terlebih dahulu dan menggunakan Sisipkan/Percikan Mini setelahnya.)

##### Alat Kamera

Alat Kamera dapat digunakan untuk membuat gambar langsung dari bagian lain buku kerja. Ini memungkinkan seseorang untuk membuat area laporan ringkasan tanpa harus membuat referensi rumus ke data, dan untuk dengan mudah menggabungkan beberapa area di satu tempat. (Gambar langsung diperbarui saat area sumber dalam buku kerja berubah.) Alat ini dapat ditambahkan ke Bilah Alat Akses Cepat dengan menggunakan Opsi Excel/Bilah Alat Akses Cepat dan di bawah "Pilih Perintah Dari", pilih "Perintah yang tidak ada di Pita" (pada menu tarik-turun) dan klik dua kali pada ikon Kamera.

Setelah ditambahkan dengan cara ini, alat ini dapat digunakan dengan memilih rentang Excel yang ingin diambil gambarnya, klik ikon Kamera di QAT, dan gambar persegi panjang di

area yang diinginkan untuk gambar. Bahkan, hasil serupa juga dapat dicapai dengan menggunakan Gambar Tertaut (tanpa mengakses alat Kamera secara eksplisit): seseorang cukup memilih rentang sumber yang diperlukan dan menyalinnya (menggunakan Ctrl+C), lalu klik pada rentang di Excel tempat gambar ingin ditempatkan, diikuti dengan menggunakan Beranda/Tempel/Opsi Tempel Lainnya/Gambar Tertaut.

## BAB 12

### MEMULAI DENGAN VBA

#### 12.1 PENDAHULUAN

Bab ini menjelaskan beberapa elemen inti Visual Basic for Applications (VBA) Excel. Pemahaman yang baik tentang hal ini sangat penting bagi siapa pun yang ingin maju melampaui aplikasi pemodelan keuangan tingkat dasar. Teks ini berfokus pada topik-topik yang kami anggap paling penting. Kami bertujuan untuk memungkinkan pembaca yang masih pemula dengan VBA untuk cepat terbiasa dengan operasi inti dan mampu menulis dan menggunakan beberapa kode yang sederhana, namun praktis dan tangguh.

Secara khusus, kami berfokus pada subrutin dan fungsi yang ditentukan pengguna: subrutin terdiri dari urutan tugas atau proses, sedangkan fungsi yang ditentukan pengguna melakukan kalkulasi khusus yang mungkin tidak tersedia sebagai fungsi Excel. Area VBA lainnya, seperti formulir pengguna dan modul kelas, tidak tercakup dalam teks ini. Dengan cara ini, kami berharap dapat membuat teks yang dapat diakses oleh pemula sekaligus tetap bernilai tinggi saat diterapkan pada banyak situasi pemodelan di dunia nyata.

Bab ini mencoba menyajikan hanya konsep-konsep yang penting untuk memulai dan menunjukkan konsep-konsep inti; topik-topik yang berkaitan dengan penulisan kode yang lebih umum dan kuat dibahas kemudian dalam teks. Bagian pertama dari bab ini memberikan gambaran umum tentang penggunaan utama VBA. Bagian kedua menyajikan beberapa blok penyusun inti dan operasi-operasi utama. Perhatikan bahwa dalam Bagian teks ini, kami menggunakan istilah "Excel" (atau "Excel murni") untuk merujuk ke bagian tradisional Excel yang tidak melibatkan VBA (yaitu terdiri dari buku kerja, kisi lembar kerja, fungsi, dan sebagainya).

Selain itu, sementara istilah "makro" sering digunakan untuk merujuk ke VBA, kami umumnya akan menggunakannya untuk merujuk ke subrutin atau prosedur lain (dan bukan untuk fungsi-fungsi yang ditentukan pengguna, misalnya). Penggunaan Utama VBA Penggunaan utama VBA dapat diringkas sebagai: Mengotomatiskan tugas. Membuat fungsi-fungsi yang ditentukan pengguna. Mendeteksi dan bereaksi terhadap peristiwa Excel atau model.

- Meningkatkan atau mengelola antarmuka pengguna.
- Mengembangkan aplikasi umum.

Bagian ini menyediakan deskripsi singkat tentang masing-masing, dengan sebagian besar teks lainnya dikhususkan untuk dua topik pertama.

#### 12.2 OTOMATISASI TUGAS

Tugas yang perlu diulang (dalam bentuk yang sama) berkali-kali dapat diotomatisasi untuk meningkatkan efisiensi. Area yang umum meliputi:

- Manipulasi set data. Ini dapat mencakup operasi untuk membersihkan data yang mendasarinya atau untuk menggabungkan data bersama-sama.
- Kueri dan ekstraksi basis data. Ini termasuk saat kueri basis data perlu dijalankan beberapa kali, terutama dalam kasus di mana penggunaan beberapa rentang kriteria basis data akan merepotkan.
- Menyelesaikan referensi melingkar melalui iterasi.
- Menjalankan skenario. Terutama dalam kasus di mana model berisi referensi melingkar yang diselesaikan menggunakan makro, menjalankan beberapa skenario umumnya juga memerlukan makro.
- Menjalankan simulasi. Penggunaan simulasi (Monte Carlo) merupakan generalisasi dari pemodelan skenario, di mana banyak skenario dihasilkan secara otomatis menggunakan pengambilan sampel acak dari distribusi probabilitas.
- Menjalankan GoalSeek atau Solver Excel beberapa kali.

### **Membuat Fungsi yang Ditentukan Pengguna**

Fungsi yang ditentukan pengguna (atau dibuat pengguna) dapat sangat ampuh dalam banyak situasi. Di sini, kami hanya mencatat bahwa keuntungan utama yang dapat dicapai meliputi melakukan perhitungan yang sulit dibuat dan dimodifikasi di Excel, mengurangi ukuran antarmuka model di Excel, meningkatkan fleksibilitas dalam struktur model, dan membuat logika yang mudah digunakan kembali serta transparansi yang lebih baik dalam perhitungan. Nanti di bab ini, kami memberikan contoh pengantar yang sederhana. Cakupan yang lebih rinci dan contoh praktis lebih lanjut disediakan di Bab 33.

### **Mendeteksi dan Bereaksi terhadap Peristiwa Model**

Dalam beberapa konteks, diperlukan untuk mendeteksi perubahan yang terjadi pada, atau di dalam, buku kerja, dan menanggapi dengan cara tertentu. Ini dapat mencakup:

- Menjalankan makro secara otomatis setiap kali buku kerja dibuka, jika data diubah, atau jika beberapa peristiwa buku kerja lainnya telah terjadi, termasuk:
- Memastikan bahwa pengguna selalu secara otomatis dibawa ke lembar kerja model utama (atau yang lain) saat buku kerja dibuka.
- Menampilkan pernyataan Disclaimer saat buku kerja dibuka.
- Menemukan sel yang nilainya terakhir diubah.
- (Otomatis) menjalankan makro saat sel mana pun dalam rentang input diubah.
- Menyorot baris dan kolom sel mana pun setiap kali seseorang mengklik dua kali di atasnya.

### **Meningkatkan atau Mengelola Antarmuka Pengguna**

VBA dapat digunakan untuk meningkatkan antarmuka pengguna. Jenis interaksi yang paling sederhana adalah kotak pesan (MsgBox) dan kotak input (InputBox). Contoh yang lebih canggih termasuk formulir pengguna (UserForm), yang dapat menerima input dari pengguna dalam proses terstruktur, memastikan integritas data input atau membuat basis data dengan



data yang diinput.

### **Pengembangan Aplikasi**

Lingkungan Excel-VBA secara mengejutkan sangat kuat dan fleksibel untuk pengembangan aplikasi "lengkap". Aplikasi ini dapat mengintegrasikan berbagai kemampuan, seperti meningkatkan antarmuka pengguna, mengakses data dalam aplikasi lain (seperti Access), memanipulasi data, melakukan perhitungan, dan membuat laporan dan bagan. Add-in juga cukup mudah dibuat, terutama untuk kode yang sering digunakan atau harus diakses dari sejumlah buku kerja. Add-in ini dapat dibagikan dengan mudah, dan juga dilindungi kata sandi. Meskipun pembuatan dan penggunaan add-in secara sistematis berada di luar cakupan teks ini, perlu dicatat bahwa add-in dapat dibuat hanya dengan menyimpan buku kerja asli dengan ekstensi file .xlam saat menggunakan fitur Simpan Sebagai Excel.

### **12.3 OPERASI UTAMA**

Di bagian ini, kami membahas elemen dasar yang berkaitan dengan area utama pembuatan dan pengoperasian kode VBA:

- Menampilkan tab Pengembang pada bilah alat Excel.
- Mengakses Editor Visual Basic.
- Merekam makro.
- Menulis kode dan mengadaptasi kode yang direkam.
- Menjalankan kode.
- Membuat kode dengan menulis daripada merekamnya.

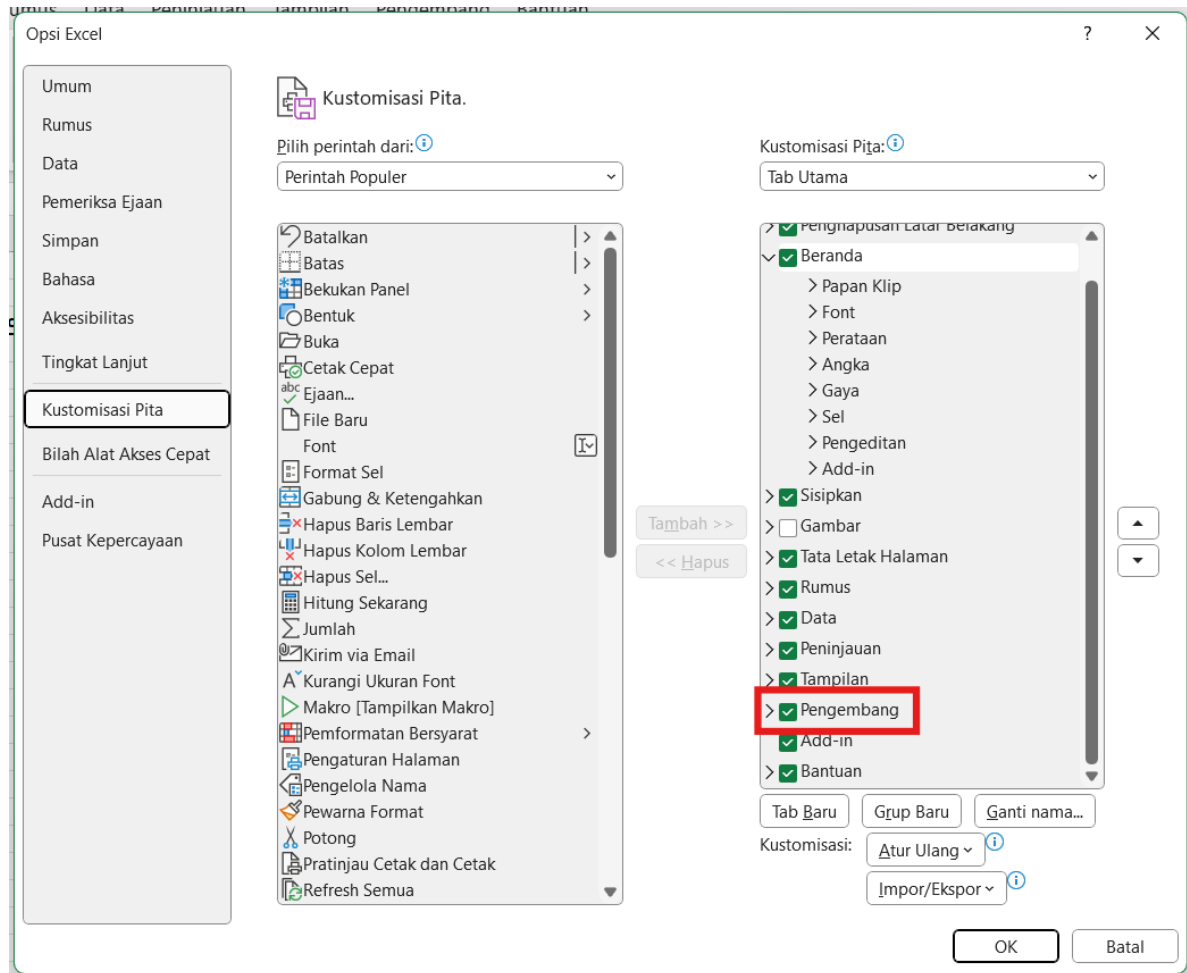
#### **Menambahkan Tab Pengembang ke Bilah Alat Excel**

##### **Menambahkan Tab Pengembang ke Bilah Alat Excel**

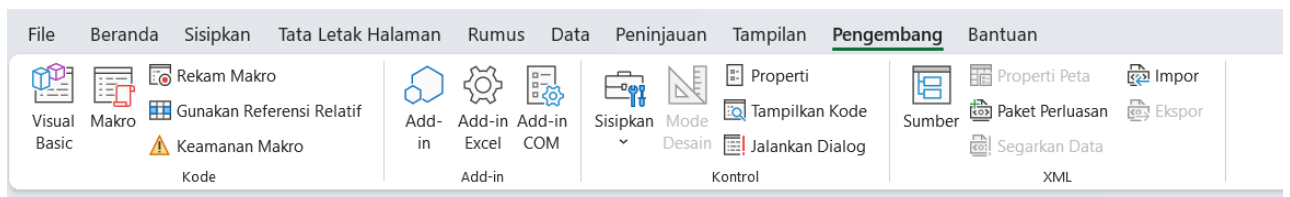
Tab Pengembang (Developer) di Excel memberikan akses ke berbagai alat canggih yang berguna untuk pengembangan dan pemrograman, seperti Visual Basic for Applications (VBA), pembuatan makro, dan kontrol formulir. Secara default, tab ini tidak muncul di bilah alat Excel. Namun, Anda dapat menambahkannya dengan mengikuti beberapa langkah sederhana.

Untuk menambahkan Tab Pengembang, buka menu "File", pilih "Options", lalu pilih "Customize Ribbon". Di bagian kanan, beri tanda centang pada opsi "Developer" dan klik "OK". Setelah itu, tab Pengembang akan muncul di bilah alat, memberikan akses langsung ke berbagai fitur pengembangan yang lebih lanjut.

Untuk bekerja dengan VBA, biasanya lebih mudah untuk menampilkan tab Pengembang Excel (yang biasanya tidak terjadi pada pengiriman default Excel). Ini dapat dilakukan di bawah menu Opsi Excel, meskipun cara yang tepat untuk melakukannya tergantung pada versinya. Gambar 12.1 memperlihatkan proses pemanggilan tab Developer di Excel 2016, dan Gambar 12.2 memperlihatkan toolbar Excel yang dihasilkan.



**Gambar 12.1 Memanggil Tab Developer**



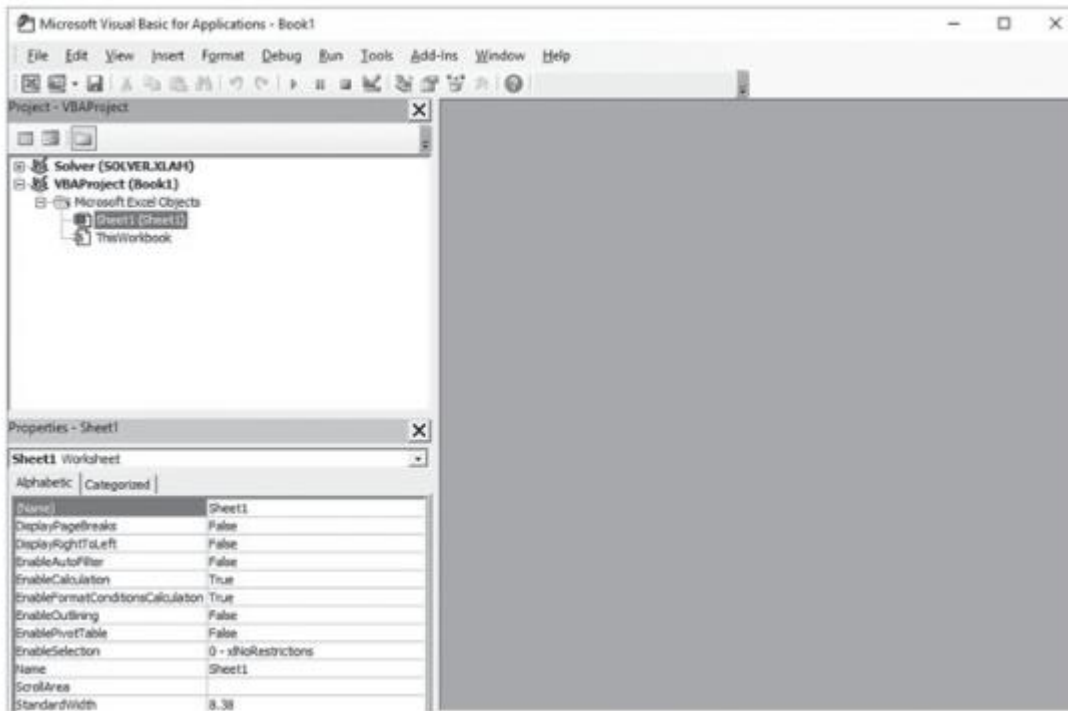
**Gambar 12.2 Tab Developer**

### Editor Visual Basic

Editor Visual Basic (VBE) diperlukan untuk menulis atau mengubah kode. Editor ini dapat diakses menggunakan tombol Visual Basic pada tab Developer (atau pintasan Alt+F11). Secara default, Project Explorer dan jendela Properties akan ditampilkan dalam VBE; jika tidak, keduanya dapat diakses di bawah menu VBE View (lihat Gambar 12.3).

Jendela Project memperlihatkan daftar buku kerja yang terbuka di lingkungan pengguna, sehingga akan berbeda untuk setiap pengguna dan bervariasi dari satu sesi kerja ke sesi kerja lainnya. Perlu dicatat bahwa sering kali lebih mudah menggunakan layar terbagi vertikal hanya dengan mengubah ukuran masing-masing jendela Excel dan VBE, dan

menatanya berdampingan.



**Gambar 12.3 Elemen Inti Editor Visual Basic**

### Merekam Makro

Perekaman makro adalah proses saat seseorang memerintahkan Excel untuk merekam semua perubahan yang dibuat pada lingkungan Excel saat seseorang bekerja. Proses perekaman dapat dilakukan dengan menggunakan tombol Rekam Makro pada tab Pengembang untuk memulai proses (lihat Gambar 12.2) lalu menggunakan tombol Hentikan Perekaman Makro (yang hanya muncul setelah proses perekaman dimulai) saat selesai.

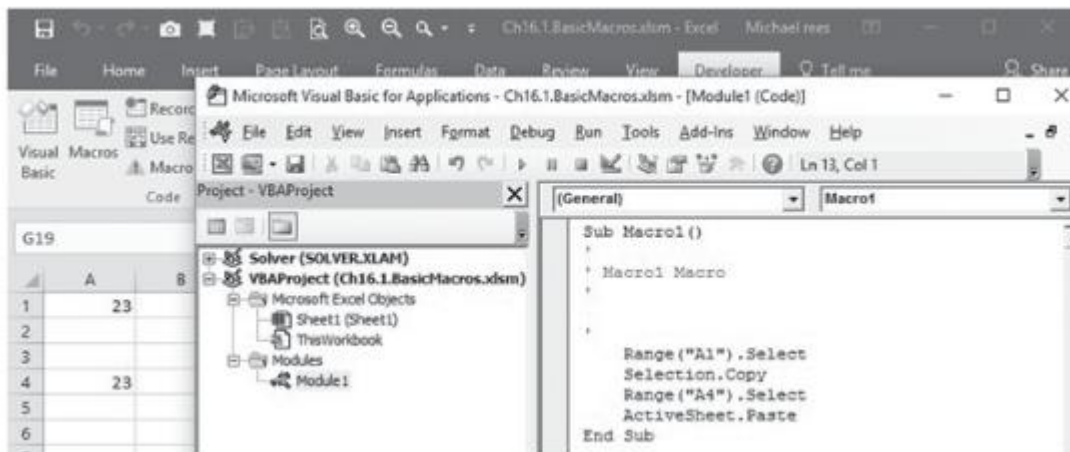
Beberapa hal yang perlu diingat adalah:

- Sebelum merekam, sebaiknya praktikkan urutan operasi yang akan digunakan. Ini membantu menghilangkan perekaman proses yang tidak perlu (seperti pemilihan tab lembar kerja atau item menu yang salah).
- Ada opsi (di awal proses perekaman) untuk memberi nama makro, menetapkan tombol pintasan, dan deskripsi. Namun, hal ini umumnya tidak diperlukan, karena proses perekaman biasanya hanya boleh digunakan untuk menyediakan informasi tentang sintaksis yang benar untuk digunakan pada operasi tertentu, dengan sintaksis ini digunakan (atau disalin dan diadaptasi) dalam kode baru yang ditulis secara langsung.
- Pilihan lain adalah menyimpan makro dalam Buku Kerja Makro Pribadi, yang akan membuat file dalam folder XLStart Excel yang selalu dimuat saat Excel dimulai. Pendekatan ini tidak digunakan dalam teks ini karena tidak sesuai jika makro (subrutin atau fungsi yang ditentukan pengguna) diperlukan sebagai bagian integral dari model

(yang mungkin perlu dibagikan dengan, atau dikirim ke, orang lain). Namun, untuk beberapa area aplikasi yang mungkin perlu dilakukan oleh pembuat model, tetapi tidak harus menjadi bagian inheren dari model tertentu (seperti beberapa aspek manipulasi data), penggunaannya mungkin sesuai.

- Umumnya masuk akal untuk menempatkan buku kerja pada Perhitungan ulang manual (di bawah File/Opsi/Rumus) sebelum makro direkam. Ini akan membantu seseorang untuk fokus pada tahap-tahap yang mengharuskan instruksi perhitungan ulang dalam kode (seperti saat menekan F9 akan diperlukan saat perhitungan ulang Manual). Melakukan hal itu akan membantu memastikan bahwa makro berfungsi di semua lingkungan (yaitu, terlepas dari apakah pengaturan perhitungan pada Manual atau Otomatis). Pendekatan alternatif adalah beralih ke perhitungan ulang Manual sebagai langkah pertama dalam proses perekaman.
- Jangan lupa untuk menghentikan perekaman saat langkah-langkah yang diinginkan telah selesai; ini adalah kesalahan umum.

Gambar 12.4 menunjukkan VBE dan menampilkan kode yang dihasilkan dari perekaman makro saat operasi Salin/Tempel dilakukan (di mana Sel A1 disalin ke Sel A4).



**Gambar 12.4 Merekam Operasi Salin/Tempel**

#### 12.4 Adaptasi Umum yang Diperlukan Saat Menggunakan Kode yang Direkam

Meskipun banyak pemula memulai dengan merekam kode (dan memang sering menggunakan kode tersebut dalam model akhir mereka), umumnya jarang kode yang dibuat dengan perekaman cukup umum, efisien, atau tangguh. Biasanya, kode yang direkam memerlukan adaptasi, seperti:

- Meningkatkan keumuman aplikasi:
  - ✓ Mengganti referensi sel tetap, baik dengan rentang bernama atau dengan kode yang menentukan rentang yang benar (seperti tempat data dimulai, atau merujuk ke rentang data dengan ukuran yang ditentukan oleh kumpulan data).
  - ✓ Menggunakan loop untuk mengulang operasi serupa (perhatikan bahwa loop tidak

dapat direkam).

- ✓ Menambahkan pernyataan bersyarat (ini juga tidak dapat direkam).
- ✓ Menambahkan perhitungan ulang paksa Excel; kode yang telah direkam saat mode perhitungan Excel diatur (default-nya) Otomatis tidak akan secara eksplisit menangkap bahwa Excel mungkin telah menghitung ulang selama proses tersebut. Untuk memastikan bahwa kode berfungsi dengan benar dengan pengaturan lain (seperti Manual), mungkin perlu untuk memaksa perhitungan ulang Excel secara eksplisit pada titik-titik tertentu.

b. Meningkatkan efisiensi:

- ✓ Mengganti operasi yang melibatkan pemilihan rentang (seperti Salin/Tempel) dengan pernyataan yang lebih langsung (seperti penugasan pada nilai dari satu rentang ke rentang lainnya); ini umumnya akan menghasilkan kode yang jauh lebih efisien dan lebih cepat untuk dijalankan. Demikian pula, pendekatan langsung untuk mengidentifikasi atau merujuk lembar kerja biasanya akan lebih efisien daripada melakukan operasi Excel untuk memilih objek tersebut.
- ✓ Menghapus atau mengubah langkah-langkah yang mungkin tidak diperlukan, tetapi muncul dalam perekaman (seperti operasi pemformatan).

Dengan demikian, alasan utama (paling valid) untuk merekam kode adalah sebagai sarana untuk memfasilitasi penemuan sintaksis yang benar untuk operasi khusus (seperti menangkap pintasan Excel yang setara dengan VBA, menyegarkan PivotTable, menjalankan GoalSeek atau Solver, dan sebagainya). Dengan kata lain, hasil rekaman digunakan untuk memberikan wawasan terhadap suatu fungsi, yang darinya instruksi relevan secara spesifik diturunkan.

Perhatikan juga bahwa banyak aspek VBA tidak dapat direkam, dan karenanya harus ditulis: ini termasuk loop, pernyataan kondisional (dan banyak aspek lain dari pengendalian eksekusi) dan penggunaan fungsi.

### **Menulis Kode**

Sebagai aturan, untuk memiliki kode VBA yang kuat dan efektif, seseorang perlu menulis sebagian besarnya dari awal (atau menyalin dan mengadaptasinya dari tempat lain), dilengkapi dengan sesekali merujuk dan mengadaptasi kode yang direkam. Seperti yang ditunjukkan pada Gambar 12.4, di mana kode dihasilkan dengan merekam operasi Excel, modul kode akan disisipkan secara otomatis (dan jika sudah ada, kode baru yang direkam akan ditempatkan di dalamnya). Namun, saat menulis kode dari awal, perlu dipastikan bahwa ada modul kode yang tersedia dalam buku kerja yang sesuai (Proyek); ini dapat dicapai dengan memilih nama buku kerja dalam Jendela Proyek dan menggunakan Sisipkan/Modul. Beberapa poin inti perlu diperhatikan dalam hal ini:

- ThisWorkbook dan Sheet adalah "modul objek" dan dicadangkan untuk prosedur yang berkaitan dengan peristiwa buku kerja dan lembar kerja, dan tidak boleh digunakan untuk menempatkan kode tujuan umum. Misalnya, modul ThisWorkbook dapat

digunakan untuk menulis kode yang akan dijalankan setiap kali buku kerja dibuka (lihat contoh selanjutnya). Kemungkinan modul Class Module dan UserForm juga berada di luar cakupan teks ini.

- Di dalam modul kode, seseorang perlu menentukan apakah ia menulis subrutin atau fungsi:
- Subrutin dapat dibuat dengan mengetik kata Sub diikuti dengan nama (seperti Sub FirstCode()); kata End Sub muncul secara otomatis. Tanda kurung kosong setelah nama menunjukkan bahwa subrutin tidak memiliki parameter yang terkait dengannya. Banyak subrutin adalah urutan tugas yang tidak memerlukan argumen, meskipun beberapa mungkin memerlukannya (kode pada Gambar 12.4 menunjukkan contoh yang serupa).
- Fungsi dapat dibuat dengan mengetik Function diikuti dengan nama dan daftar argumen dalam tanda kurung; kata End Function akan muncul secara otomatis.
- Jika beberapa prosedur harus ditulis, prosedur ini dapat disertakan dalam modul yang sama atau dalam modul baru. Jika beberapa prosedur telah dibuat dalam satu modul, daftar drop-down dalam kotak Prosedur dapat digunakan untuk berpindah dengan cepat di antara prosedur tersebut. Seseorang dapat memilih untuk membagi kode ke beberapa modul untuk meningkatkan transparansi dan memudahkan navigasi. Misalnya, subrutin dapat ditempatkan dalam satu (atau beberapa modul) dan fungsi di modul lain. Dalam kasus seperti itu, akan berguna untuk memberikan nama yang sesuai untuk setiap modul dengan menggunakan kotak Nama dalam jendela Properti.
- Nama prosedur dalam modul (subrutin atau fungsi) dapat dipilih menjadi apa saja (dengan beberapa batasan, seperti tidak mengandung spasi, tidak menggunakan kata-kata yang dicadangkan, dan sebagainya; lihat nanti untuk detail lebih lanjut).
- Kode akan disimpan setiap kali buku kerja disimpan sebagai file .xlsm (file .xlsx mungkin tidak berisi kode).

Jendela kode juga akan muncul setelah modul dimasukkan; jika tidak, Lihat/Kode dapat digunakan untuk menampilkannya. Kode dalam modul apa pun dapat dibuat terlihat dengan mengklik dua kali pada nama modul. Saat menulis kode, beberapa hal yang perlu diingat adalah:

- Baris komentar hanyalah catatan yang tidak berpengaruh saat kode dijalankan. Komentar dimulai dengan apostrof, dan dapat berada pada baris yang berdiri sendiri atau ditempatkan pada baris kode (setelah kode). Baris komentar ditampilkan secara otomatis dalam warna hijau; makro yang terekam pada Gambar 12.4 berisi beberapa baris komentar yang disisipkan secara otomatis.
- Baris kode atau komentar dapat dilanjutkan (pada baris lain), dengan menggunakan SPACE diikuti oleh UNDERSCORE di akhir baris yang akan dilanjutkan.
- Indentasi dapat digunakan untuk membuat kode lebih menarik secara visual. Tombol Tab dapat digunakan untuk membuat indentasi kode, dan Shift+Tab untuk

menghilangkan indentasi. Menu Tools/Options/Editor di VBE dapat digunakan untuk mengubah lebar tab (saat menggunakan beberapa tingkat indentasi, lebar tab default empat spasi sering kali terlalu lebar).

- Saat berada di jendela Kode, mengetikkan titik setelah objek akan menghasilkan menu tarik-turun yang muncul, yang menyajikan daftar item menu yang tersedia terkait objek tersebut. Misalnya, dengan mengetikkan Range("A1"). kita akan melihat item menu yang relevan dengan suatu rentang, seperti Range("A1").Activate untuk menjadikan A1 sel aktif, Range("A1").ClearContents untuk menghapus konten Sel A1, dan Range("A1").Value untuk merujuk ke nilainya (nilai juga merupakan properti default suatu rentang, jadi Range("A1") akan ditafsirkan sebagai merujuk ke nilai).

### Menjalankan Kode

Sebelum menjalankan makro VBA apa pun, penting untuk menyadari bahwa secara umum tidak ada fitur batal. Jadi, kode yang mengubah lembar kerja Excel dengan cara tertentu (seperti menimpa data atau rumus yang ada) dapat menyebabkan masalah jika belum diperiksa terlebih dahulu. Oleh karena itu, sebelum menjalankan kode apa pun, sering kali ada baiknya untuk menyimpan cadangan berkas, atau setidaknya memeriksa sebelum menyimpan berkas bahwa makro tidak menghasilkan kesalahan apa pun. Makro (atau subrutin) dapat dijalankan dengan beberapa cara:

- Saat berada di VBE, dengan kursor ditempatkan di dalam kode subrutin yang akan dijalankan, tekan F5 (atau Run/Run Sub). Makro juga dapat dipanggil dari makro lain, sehingga, dalam situasi yang lebih kompleks, struktur tertanam atau berurutan dapat dibuat untuk memodulasi kode menjadi komponen yang lebih kecil.
- Saat berada di tab Pengembang Excel, gunakan tombol Makro untuk memilih makro lalu pilih Jalankan. Pintasan Alt+F8 juga dapat digunakan (dari mana saja di Excel) untuk memanggil perintah Makro.
- Di dalam Excel, dengan mengklik objek mana pun yang telah ditetapkan makro padanya. Objek seperti itu dapat dibuat dengan cara standar apa pun di Excel (seperti menggunakan menu Sisipkan untuk membuat Kotak Teks, Bagan, atau bentuk atau grafik lainnya), dengan mengklik kanan pada bentuk yang digunakan untuk memanggil menu Tetapkan Makro. Atau, perintah Sisipkan/FormControls pada tab Pengembang dapat digunakan untuk membuat tombol atau kontrol lainnya, dalam hal ini proses penetapan makro dipanggil secara otomatis. (Kontrol ActiveX umumnya tidak boleh digunakan.) Objek seperti kotak teks dan tombol tentu saja dapat diubah namanya, diubah ukurannya, dan diposisikan ulang sesuai keinginan (misalnya dengan mengklik kanan untuk mengedit). Penting untuk memberi label dengan jelas untuk memastikan bahwa makro tidak berjalan secara tidak sengaja. Oleh karena itu, harus dijelaskan kepada pengguna dengan cara tertentu bahwa mengklik bentuk akan menjalankan makro (misalnya dengan memberi label pada bentuk); jika ini tidak jelas, maka makro dapat berjalan secara tidak sengaja, yang mungkin tidak diinginkan. Misalnya,

menetapkan makro yang diperlukan untuk memperbarui data bagan ke bagan akan tampak masuk akal, sedangkan jika makro melakukan beberapa operasi lain (tidak terkait dengan bagan), maka mungkin akan lebih baik untuk menggunakan tombol terpisah.

- Dari menu Kustomisasi pada Bilah Alat Akses Cepat, pilih Perintah Lainnya, lalu Makro (sebagai ganti Perintah Populer default), pilih nama makro, dan pilih Tambah. Kita kemudian dapat menggunakan menu Ubah untuk menggunakan simbol guna mewakili makro. Karena ini akan mengubah antarmuka Excel umum dan membuat ikon yang hanya akan berpengaruh saat buku kerja yang berisi makro aktif, mungkin ada baiknya melakukannya hanya untuk prosedur yang akan digunakan dalam berbagai buku kerja, seperti beberapa prosedur dalam Buku Kerja Makro Pribadi (yang tidak kita gunakan dalam teks ini; lihat sebelumnya).
- Dari Formulir Pengguna.

### **Teknik Debugging**

Sebelum menjalankan kode secara menyeluruh, ada baiknya menjalankan beberapa pemeriksaan sederhana:

- Menggunakan menu Debug/Compile akan sering menemukan kesalahan dasar sintaksis atau konsistensi.
- Menelusuri kode baris demi baris, dengan menggunakan tombol F8 (atau Debug/Step Into). Ini akan sering menyoroti lokasi kesalahan mendasar yang mencegah eksekusi kode secara menyeluruh. Saat menelusuri, beberapa hal perlu diingat:
- Baris kode yang akan dijalankan (tetapi belum dijalankan) akan muncul dengan warna kuning.
- Menempatkan kursor di atas variabel akan menunjukkan nilainya pada titik tersebut dalam eksekusi kode.
- Ctrl+F8 atau Debug/Run to Cursor dapat digunakan untuk menjalankan dari awal hingga posisi kursor; ini dapat berguna jika seseorang yakin bahwa kode hingga titik tersebut berfungsi dan seseorang ingin memeriksa ulang dengan menjalankannya.
- Kita dapat mengatur breakpoint dalam kode dengan mengklik bilah sisi, atau menempatkan kursor pada baris kode yang diinginkan dan menekan F9, atau menggunakan Debug/Toggle Breakpoint. Kode tersebut kemudian dapat dijalankan ke breakpoint dengan menggunakan F5 (dengan asumsi kode tersebut bebas kesalahan hingga titik tersebut) dan setelah itu menggunakan F8 untuk melangkah melalui baris-baris berikutnya secara individual.
- Kotak pesan sederhana (misalnya MsgBox x) dapat digunakan untuk mengembalikan nilai variabel pada baris tersebut dalam kode. Kotak tersebut dapat dikomentari (menggunakan apostrof) atau dihapus jika tidak lagi diperlukan.
- Saat menjalankan kode, tombol Run/Break dan Run/Reset dapat digunakan untuk menghentikan eksekusi kode dan keluar, seperti jika kita terjebak dalam loop yang



panjang atau tak berujung (Task Manager Excel dapat digunakan sebagai pilihan terakhir).

- Menu Bantuan VBA dapat diakses dari bilah alat atau menggunakan F1 dari dalam VBE. Meskipun penggunaan teknik step-through dan break-point ini biasanya cukup untuk kode sederhana, teknik debugging dan penulisan kode yang kuat lainnya mungkin diperlukan atau direkomendasikan dalam contoh praktis umum. Bab-bab selanjutnya dalam Bagian ini membahasnya secara terperinci.

Perhatikan bahwa fakta bahwa kode dapat melakukan operasi tanpa crash tidak sama dengan kode yang bebas dari kesalahan atau memenuhi persyaratan fungsionalitasnya! Ini sama seperti di Excel, di mana fakta bahwa rumus menghitung tanpa mengembalikan pesan kesalahan tidak berarti bahwa modelnya benar. Selain itu, sebagai aturan umum, jenis kemungkinan kesalahan yang dapat terjadi di VBA biasanya lebih bervariasi dan berpotensi lebih halus daripada di Excel.

## 12.5 CONTOH SEDERHANA

Bagian ini menyediakan beberapa contoh kode sederhana yang menggambarkan banyak poin yang dibahas sebelumnya dalam bab ini. Kami menunjukkan contoh:

- Mengambil nilai dari sel Excel dan menggunakannya dalam kode VBA.
- Menggunakan rentang Excel bernama untuk membuat komunikasi antara Excel dan VBA lebih kuat.
- Menempatkan nilai dari kode VBA ke dalam rentang Excel.
- Mengganti operasi Salin/Tempel dengan pernyataan penugasan.
- Membuat fungsi sederhana yang ditentukan pengguna.

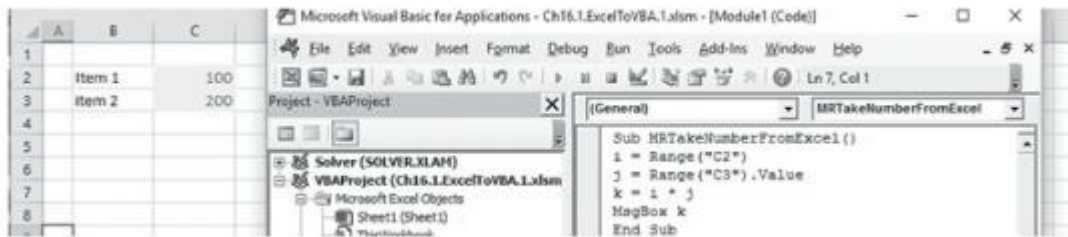
Saat menjalankan kode dalam contoh file, makro di dalamnya perlu diaktifkan. Ini dapat dilakukan secara individual (sebagai respons terhadap peringatan keamanan yang mungkin muncul), atau dengan mengaktifkan sementara semua makro menggunakan Pengaturan Makro dalam opsi Pusat Kepercayaan (pada menu Opsi Excel), atau dengan menetapkan folder yang berisi file sebagai lokasi tepercaya menggunakan opsi Tambahkan Lokasi Tepercaya.

### Contoh: Menggunakan Nilai Sel Excel dalam VBA

Gambar 12.5 berisi contoh kode yang membaca nilai dari dua sel Excel ke dalam VBA, mengalikannya, dan menampilkan hasilnya kepada pengguna dalam MessageBox (lihat Gambar 12.5).

Kode yang digunakan adalah:

```
Sub MRTakeNumberFromExcel ()
    i &equals; Range ("C2")
    j &equals; Range ("C3").Value
    k &equals; i * j
    MsgBox k
End Sub
```

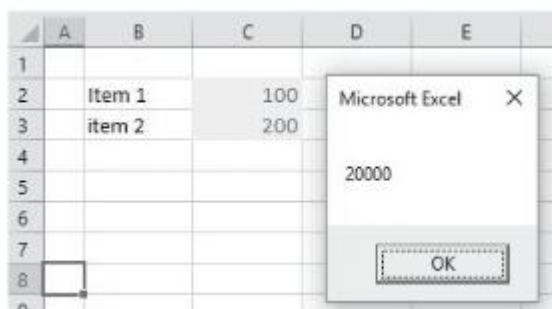


**Gambar 12.5 Contoh Sederhana Penggunaan Nilai dari Excel di VBA**

Perhatikan hal berikut:

- Rentang sel Excel dirujuk dengan cara yang menggunakan referensi sel sebagai bidang teks (mis. Rentang("C2"), bukan Rentang(C2)). Oleh karena itu, bidang teks tersebut tidak akan menyesuaikan secara otomatis jika baris diperkenalkan di bagian atas lembar kerja, dan kode tidak akan berfungsi seperti yang diantisipasi; ini adalah salah satu alasan mengapa penggunaan rentang bernama untuk sel Excel akan lebih kuat (lihat nanti).
- Tanda = bukanlah pernyataan kesetaraan matematis; melainkan operasi yang menetapkan item di sebelah kanan ke item di sebelah kiri. Ini mirip dengan bahasa komputasi lain, dan juga dengan Excel, di mana (misalnya) saat bekerja di Sel B5, pengetikan =B2 akan menetapkan konten B2 ke B5.
- Properti default dalam VBA dari satu sel Excel adalah nilainya. Oleh karena itu, pernyataan .Value yang digunakan dalam operasi penugasan kedua tidak diperlukan di sini. Namun, secara umum akan lebih kuat dan transparan untuk secara eksplisit menyatakan properti sel mana yang diperlukan (misalnya formatnya, warna teks atau ukuran font), karena kode VBA dapat digunakan untuk melakukan operasi berdasarkan banyak properti sel dan rentang.
- Jika rentang yang dimaksud adalah rentang multisel, properti .Value umumnya diperlukan agar kode dapat dijalankan dengan benar.

Gambar 12.6 menunjukkan hasil dari menjalankan kode ini, menampilkan kotak pesan (Msg-Box) dari hasilnya.

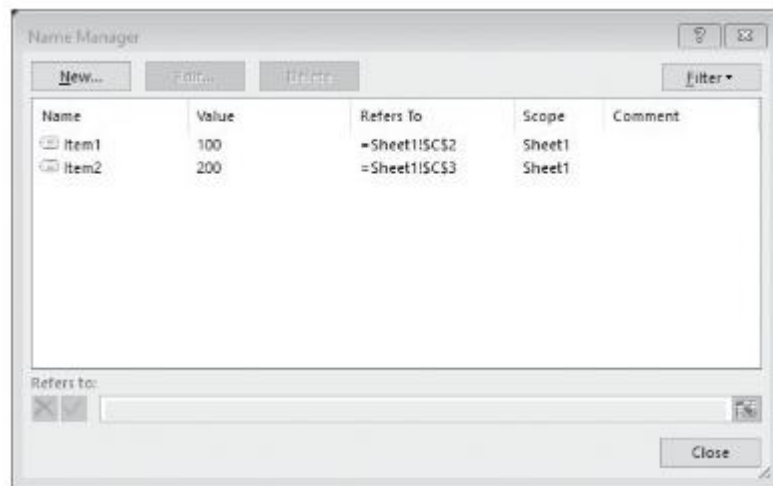


**Gambar 28.6 Hasil Menjalankan Beberapa Kode Sederhana, dengan Kotak Pesan Ditampilkan**

### Contoh: Menggunakan Rentang Excel Bernama untuk Ketahanan dan Fleksibilitas

Untuk membuat prosedur di atas lebih kuat, akan lebih baik jika rentang sel Excel diberi nama. Ini akan memungkinkan sel-sel individual dipindahkan ke tempat lain (atau baris baru disisipkan di atasnya, atau kolom di depannya) tanpa memengaruhi hasil. Gambar 12.7 menunjukkan nama-nama yang diberikan dalam contoh ini menggunakan menu Formula/Name Manager Excel. Kode VBA kemudian dapat disesuaikan sebagai berikut:

```
Sub MRTakeNumberFromExcel ()
    i &equals; Range ("Item1")
    j &equals; Range ("Item2").Value
    k &equals; i * j
    MsgBox k
End Sub
```



**Gambar 28.7** Membuat Rentang Bernama di Excel

Perhatikan bahwa:

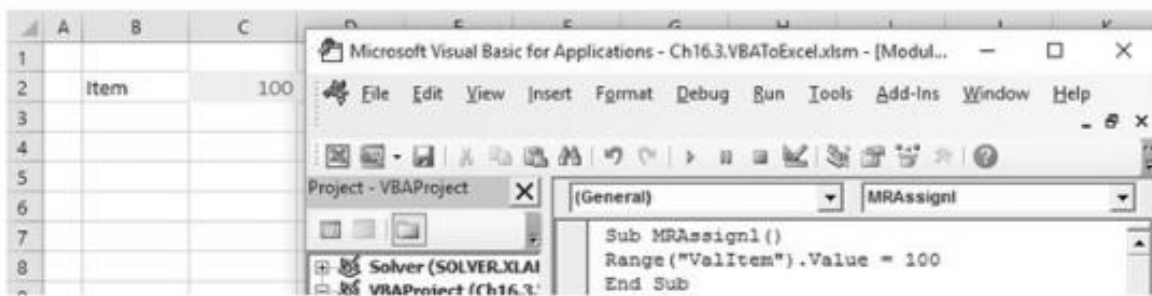
- Jika nama rentang diubah dalam buku kerja Excel, maka rentang tersebut juga perlu diubah dalam kode VBA. Tautan tersebut tidak otomatis, karena kode VBA menyertakan nama dalam tanda kurung, seperti pada bidang teks. (Sekali lagi, ini berbeda dengan rumus Excel, di mana nama yang digunakan diubah secara otomatis.)
- Dalam mengaudit model yang berisi kode VBA, sebaiknya periksa kode VBA apa pun sebelum mengubah bagian Excel dari model tersebut dengan cara apa pun (memasukkan baris, kolom, mengubah rentang bernama, atau operasi lain dapat membuat perubahan yang tidak diinginkan).
- Tombol pintas F3 dapat berguna untuk membuat daftar rentang bernama Excel, yang kemudian dapat disalin ke kode VBA untuk menghindari kesalahan ejaan.

Setelah perubahan ini dilakukan pada kode, baris dan kolom baru dapat diperkenalkan ke Excel, seperti yang diinginkan, dan kode tersebut akan berjalan dengan benar.

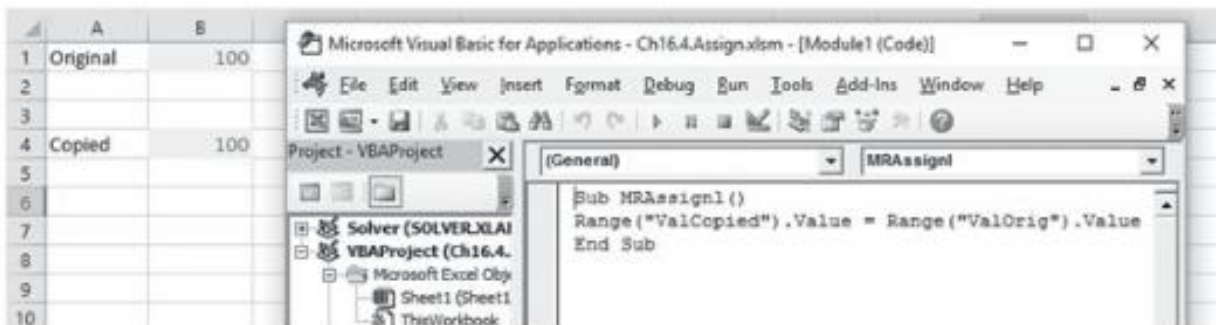
### Contoh: Menempatkan Nilai dari Kode VBA ke dalam Rentang Excel

Dalam contoh di atas, nilai ditempatkan dari kode ke dalam lembar kerja Excel. Proses tersebut dapat digunakan dalam pengertian yang berlawanan (yaitu di mana variabel yang digunakan dalam kode mengambil nilainya dari lembar kerja Excel). File Ch28.3.VBAtoExcel.xlsm berisi contoh sederhana; angka 100 ditetapkan ke nilai sel yang telah diberi rentang bernama Excel ValItem (Sel C2); Gambar 12.8 menunjukkan hasilnya.

```
Sub MRAssign1 ()
    Range("ValItem").Value &equals; 100
End Sub
```



**Gambar 12.8 Hasil Menjalankan Makro Sederhana untuk Menetapkan Nilai dari VBA ke Excel**



**Gambar 12.9 Hasil Operasi Penugasan Alih-alih Salin/Tempel**

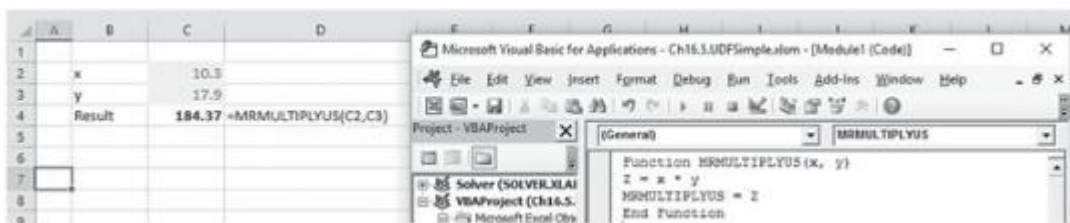
### Contoh: Mengganti Salin/Tempel dengan Penugasan

Sebagai aturan, akan jauh lebih efisien secara komputasi untuk menggunakan operasi penugasan alih-alih Salin/Tempel. Perbedaan kecepatan antara keduanya menjadi jauh lebih jelas ketika operasi dilakukan berkali-kali secara berurutan. Gambar 12.9 menunjukkan contoh bagaimana penugasan dapat digunakan untuk menempatkan nilai dari satu sel (rentang bernama) ke sel lain tanpa memilih sel secara eksplisit, atau menggunakan operasi Salin/Tempel (lihat Gambar 12.9).

```
Sub MRAssignl()
    Range("ValCopied").Value &equals; Range("ValOrig").Value
End Sub
```

### Contoh: Fungsi Sederhana yang Ditetapkan Pengguna

Fungsi dibedakan dari subrutin karena fungsi tersebut hanya dapat memberikan hasil dalam sel atau rentang Excel (untuk fungsi array). Sedangkan untuk fungsi Excel, hubungan antara parameter fungsi dan hasil perhitungan bersifat "langsung". Sebaliknya, subrutin dapat menjalankan prosedur (termasuk mengembalikan nilai ke dalam sel Excel), tetapi tidak ditautkan secara langsung ke asumsi input, sehingga perlu dijalankan ulang jika nilai asumsi tersebut berubah.



**Gambar 12.10** Contoh Sederhana Fungsi yang Ditentukan Pengguna

Gambar 12.10 menunjukkan contoh sederhana (lihat Gambar 12.10). Perhatikan bahwa fungsi yang dibuat pengguna dapat diakses di Excel dengan mengetik langsung (asalkan memiliki cakupan default Publik), atau dengan menggunakan menu Rumus/Sisipkan Fungsi (di mana fungsi tersebut akan dicantumkan di bawah kategori yang ditetapkan pengguna). Untuk memudahkan pendekatan pengetikan langsung, penulis umumnya memberi nama semua fungsinya sehingga dua huruf pertama adalah inisialnya (mis. MR. . .); tata nama seperti itu juga membantu mengidentifikasi (ketika mengaudit suatu model) apakah suatu fungsi merupakan fungsi Excel atau telah ditulis sebagai fungsi khusus.

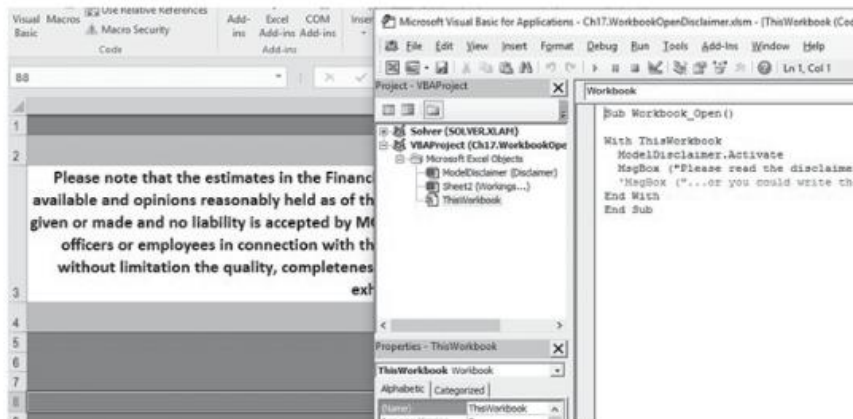
```
Function MRMULTIPLYUS(x, y)
    Z &equals; x * y
    MRMULTIPLYUS &equals; Z
End Function
```

Perhatikan bahwa kode tersebut harus berisi pernyataan return (sebelum End Function) yang secara eksplisit menetapkan nilai kalkulasi ke nilai fungsi (kesalahan yang sering terjadi adalah menghilangkan pernyataan return saat menulis kode awal). Tentu saja, seperti halnya saat menggunakan prosedur VBA lainnya, fungsi yang ditentukan pengguna memiliki kelemahan yaitu modelnya mungkin kurang transparan bagi pengguna yang tidak mengenalnya.

### Contoh: Menampilkan Pesan saat Buku Kerja Dibuka

Contoh sederhana kode berbasis peristiwa adalah menampilkan pesan secara otomatis setiap kali buku kerja dibuka. Poin utamanya adalah bahwa subrutin tertentu yang menggunakan kata khusus `Workbook_Open()` harus ditempatkan di dalam modul kode `ThisWorkbook`. Gambar 12.11 berisi contoh sederhana. Gambar 12.11 menunjukkan klip layar. Perhatikan bahwa dalam contoh ini, lembar kerja di Excel yang berisi pernyataan sanggahan diberi nama "Penafian", dan telah diberi nama kode `ModelPenafian` dalam VBA. Dengan demikian, kode tersebut mengaktifkan lembar kerja tersebut dan kemudian menampilkan kotak pesan kepada pengguna (kotak pesan tersebut juga dapat menampilkan teks pernyataan sanggahan, sebagai opsi tambahan atau sebagai alternatif).

```
Sub Workbook_Open()
    With ThisWorkbook
        ModelDisclaimer.Activate
        MsgBox ("Please read the disclaimer in the Disclaimer
worksheet")
        'MsgBox ("...or you could write the Disclaimer text here
...")
    End With
End Sub
```



**Gambar 12.11 Menampilkan Pesan saat Buku Kerja Dibuka**

Perhatikan juga bahwa prosedur dapat dijalankan setiap kali buku kerja ditutup menggunakan subrutin `Workbook_Close`, yang juga merupakan kata khusus yang harus ditempatkan dalam modul kode `ThisWorkbook`.

```
Sub Workbook_Close()
    'Write code to run here
End Sub
```

## BAB 13

### BEKERJA DENGAN OBJEK DAN RENTANG

Bab ini memperkenalkan struktur objek dalam lingkungan Excel. Kami fokus pada objek yang menjadi kunci dalam pemodelan keuangan umum, khususnya:

- Merujuk pada rentang sel.
- Menggunakan kumpulan objek dan hierarki objek.

#### 13.1 TINJAUAN UMUM MODEL OBJEK

Bagi banyak pemodel, subjek objek Excel bukanlah subjek yang tampaknya relevan secara langsung: pertama, banyak model Excel tradisional didasarkan pada operasi dan fungsi aritmatika, dengan objek yang tampaknya tidak memiliki peran nyata. Kedua, kode VBA sederhana dapat direkam atau ditulis tanpa perlu merujuk ke objek secara khusus. Namun, pengetahuan tentang prinsip objek Excel sangat penting untuk dapat menulis kode yang jelas, fleksibel, dan kuat, serta untuk mengakses rangkaian aplikasi yang luas yang dimungkinkan oleh lingkungan objek Excel/VBA, sehingga seseorang dapat mengembangkan (atau mengaudit) rangkaian aplikasi pemodelan yang lebih luas.

#### **Objek, Properti, Metode, dan Peristiwa**

Dalam banyak hal, logika yang digunakan dalam lingkungan Excel/VBA mirip dengan bahasa alami:

- Objek dianalogikan dengan kata benda. Objek pada dasarnya adalah bagian Excel yang dapat dilihat, termasuk sel, rentang, baris, kolom, buku kerja, lembar kerja, bagan, PivotTable, kotak teks, kotak komentar sel, bentuk, dan sebagainya.
- Properti dianalogikan dengan kata sifat, yang menjelaskan beberapa aspek objek, yang umumnya ada beberapa. Misalnya, properti buku meliputi penulisnya, judulnya, jumlah halamannya, beratnya, dan sebagainya, sedangkan properti sel Excel dapat meliputi nilainya, atau jenis pemformatannya.
- Metode dianalogikan dengan kata kerja (misalnya, buku dapat dibuka, ditutup, dibaca, ditaruh di rak buku, dibeli atau dijual).
- Peristiwa mirip dengan metode, tetapi tindakan dipicu oleh sistem (misalnya saat buku kerja dibuka, nilai dalam lembar kerja diubah, dan seterusnya).

Objek, properti, metode, dan peristiwa tentu saja khusus untuk hal-hal yang relevan dengan lingkungan Excel/VBA (dan tidak memiliki kekayaan dan kompleksitas penuh bahasa alami). Sebagai serangkaian instruksi yang harus dijalankan oleh komputer, sintaksis VBA lebih ringkas dan tepat daripada bahasa alami: misalnya, alih-alih "Maukah Anda mencuci piring?", kode akan dianalogikan dengan "cuci piring".

Selain itu, meskipun dalam bahasa alami umumnya jelas dari konteksnya piring mana yang dimaksud (piring kita yang kotor, alih-alih piring tetangga yang bersih), masalah

kontekstual seperti itu tidak selalu didefinisikan secara unik jika dilihat dari perspektif komputer. Untuk menulis kode yang kuat, biasanya perlu (atau lebih baik) untuk menentukan masalah tersebut dengan sangat tepat dan tidak ambigu.

Seperti dalam bahasa alami, properti suatu objek dapat menghasilkan objek lain: misalnya, "properti layar" suatu komputer juga merupakan suatu objek (yaitu layar). Demikian pula, dalam VBA, `Range("A2:B5").Rows` merujuk pada baris-baris rentang tersebut (yaitu Baris 2, 3, 4, dan 5), dan `Range("A2:B5").Rows.Count` akan menggunakan metode `Count` untuk menghitung jumlah baris (yaitu empat). Metode lain yang terkait dengan objek rentang meliputi `Activate`, `ClearContents`, `ClearFormats`, `Copy`, `PasteSpecial`, dan `Select`. Perhatikan bahwa satu objek tertentu adalah `Application`, yang mewakili seluruh aplikasi (yaitu Excel dalam kasus ini).

### Hierarki dan Koleksi Objek

Di dalam Excel/VBA, objek dapat berada dalam Koleksi dan juga dalam Hierarki.

Contoh Koleksi meliputi:

- Buku Kerja adalah kumpulan semua buku kerja yang terbuka.
- Lembar Kerja adalah kumpulan lembar kerja dalam buku kerja yang ditentukan atau aktif (objek Lembar mencakup lembar kerja dan lembar bagan).
- ObjekBagan adalah kumpulan bagan dalam lembar kerja yang ditentukan atau aktif. Bagan adalah kumpulan lembar bagan dalam buku kerja yang ditentukan atau aktif, bukan kumpulan bagan.
- Nama adalah kumpulan rentang bernama dalam buku kerja yang ditentukan atau aktif.
- TabelPivot adalah kumpulan semua TabelPivot dalam lembar kerja yang ditentukan atau aktif.
- FungsiLembar Kerja adalah objek tingkat atas, yang propertinya adalah fungsi Excel individual.
- Hal ini menimbulkan hierarki objek. Jadi, dengan menggunakan fakta bahwa Worksheets merupakan anggota objek Workbook:
  - `Workbooks("Model.xlsm").Worksheets("Data1")` merujuk ke worksheet yang ditunjukkan (yaitu Data1) dalam workbook Model.
  - `Workbooks("Model.xlsm").Worksheets.Add` merupakan bagian dari sintaksis yang diperlukan untuk menambahkan worksheet baru ke workbook Model.

Perlu diketahui bahwa saat bekerja di VBA, mengetik “.” setelah kumpulan objek akan memanggil VBA untuk menyediakan daftar metode dan properti yang dapat diterapkan pada titik tersebut. Penggunaan koleksi dapat memungkinkan operasi tertentu dilakukan dengan mudah, yang mungkin tampak rumit atau memakan waktu (lihat contoh selanjutnya).

### Menggunakan Set. . . = . . . .

Penggunaan pernyataan Set sangat penting saat bekerja dengan objek Excel. Misalnya, pada bab terakhir, kami menggunakan baris kode dasar seperti:



```
i = Range("C2")
j = Range("C3").Value
```

Setelah dipikir-pikir, menjadi jelas bahwa baris pertama kode tersebut tidak jelas atau ambigu: peran variabel `i` dalam kode tersebut adalah untuk mengambil nilai, sedangkan `C2` adalah sel (bukan hanya nilai) yang memiliki beberapa properti; nilainya, ukuran font, warna, aspek pemformatan lainnya, dan sebagainya. Oleh karena itu, baris pertama mencoba untuk menetapkan objek multifaset ke satu angka.

Hal ini pada dasarnya tidak konsisten, dan memang orang mungkin bertanya-tanya mengapa kode tersebut dapat dijalankan sama sekali (alasan adalah bahwa properti default dari objek rentang adalah nilainya). Baris kedua dari kode ini lebih jelas, karena secara eksplisit bekerja dengan properti Nilai dari rentang dan hanya menetapkan ini ke variabel `j`. Pernyataan `Set` dapat digunakan tidak hanya untuk mengatasi ambiguitas ini, dan pada kenyataannya umumnya diperlukan, terutama ketika sisi kiri dari pernyataan "=" adalah sebuah objek. Misalnya:

```
Set DataRange = Range("A1:A100")
Set Rng = ActiveSheet.UsedRange
```

Kegagalan untuk menyertakan `Set` merupakan kesalahan umum, yang mengakibatkan pesan kesalahan, seperti Variabel objek tidak disetel (Kesalahan 91) saat menjalankan kode.

### **Menggunakan Konstruksi With. . .End With**

Konstruksi `With...End With` memungkinkan seseorang untuk menentukan objek dan melakukan beberapa operasi dengan objek tersebut menggunakan sintaksis yang lebih ringkas. Jadi, seseorang dapat menulis:

```
With Range("A1").Font
    .Name = "Calibri"
    .Size = 10
    .Color = -16776961
End With
```

Konstruksi tersebut dapat ditanamkan ke dalam konstruksi lain, seperti:

```
With Application.ThisWorkbook
    With Range("A1")
        With .Font
            .Name = "Calibri"
            .Size = 10
            .Color = -16776961
        End With
    End With
End With
```

```
End With
```

```
End With
```

Penggunaan utama dalam praktik adalah:

- Untuk meningkatkan kecepatan penulisan kode yang merujuk ke objek yang sama beberapa kali.
- Untuk memfasilitasi penggunaan referensi penuh, dengan membuatnya lebih mudah diterapkan.
- Untuk membuat kode lebih transparan dengan menekankan objek mana yang sedang dikerjakan.

### Menemukan Alternatif untuk Pemilihan atau Pengaktifan Rentang dan Objek

Biasanya, kode yang dihasilkan dari proses perekaman berisi langkah-langkah di mana sel, rentang, atau objek dipilih (atau diaktifkan). Di sisi lain, sering kali operasi akhir yang dibutuhkan tidak memerlukan proses pemilihan atau aktivasi tersebut. Hampir selalu jauh lebih efisien secara komputasi dan jauh lebih cepat untuk menjalankan kode yang ditulis untuk menghilangkan pemilihan atau aktivasi objek secara eksplisit (yang sering kali, meskipun tidak selalu, memungkinkan).

Dalam beberapa kasus, seperti saat bekerja dengan bagan atau PivotTable, mungkin perlu untuk mengaktifkan objek sebelum beberapa operasi dapat dilakukan pada objek tersebut, tetapi umumnya ada baiknya mencoba menguji apakah aktivasi tersebut benar-benar diperlukan. Misalnya, perekaman kode untuk menghapus rentang A1:E8 akan memberikan:

```
Range("A1:E8").Select
    Selection.ClearContents
```

sedangkan padanan langsungnya adalah

```
Range("A1:E8").ClearContents
```

(Kata Selection bersifat khusus untuk lembar kerja dan merupakan kata yang disediakan oleh VBA untuk merujuk ke rentang terakhir yang terakhir dipilih pada lembar kerja tersebut.) Demikian pula, penggunaan (tetapi bukan pemilihan) sintaksis seperti `xlCellTypeLastCell`, `CurrentRegion` atau `UsedRange` (dibahas kemudian dalam bab ini) biasanya lebih efisien daripada hasil perekaman operasi Excel seperti `Ctrl+Shift+RightArrow`, yang mungkin memberikan kode seperti:

```
Range(Selection, Selection.End(xlDown)).Select
Range(Selection, Selection.End(xlToRight)).Select
```

### 13.2 BEKERJA DENGAN OBJEK RANGE: BEBERAPA ELEMEN UTAMA

Bagian ini berfokus pada elemen utama dalam bekerja dengan range (objek range).

#### Kemungkinan Sintaks Dasar dan Penggunaan Range Bernama

Ada banyak cara untuk merujuk ke objek range Excel:

- `Range("B2")` merujuk ke satu sel.
- `Range("B2:C10")` merujuk ke range yang terdiri dari sel B2 hingga C10.
- `Range("B2", "C10")` juga merujuk ke range yang terdiri dari sel B2 hingga C10.

#### Range Bernama dan Variabel Bernama

Seperti yang disebutkan sebelumnya, dalam kode VBA, hampir selalu lebih baik untuk menggunakan range bernama daripada referensi sel langsung. Selain lebih kuat jika struktur lembar kerja berubah (baris atau kolom baru ditambahkan), penggunaannya umumnya menciptakan lebih banyak transparansi. Ada beberapa situasi yang muncul, yang memengaruhi titik yang tepat di mana nama harus ditetapkan ke suatu rentang:

- Jika ukuran dan lokasi rentang ditetapkan, seseorang dapat menentukan nama dalam buku kerja Excel (menggunakan menu Formula/Pengelola Nama Excel) saat model sedang dibangun (atau saat kode VBA sedang ditulis). Rentang yang ditentukan dengan cara ini akan dirujuk dalam kode menggunakan tanda kutip (misalnya Rentang ("PriceInput") atau Rentang("NumberofTimestoRunLoop"), dan seterusnya). Saat mengaudit kode tersebut, untuk mengetahui rentang mana yang dirujuk oleh nama, seseorang dapat menggunakan Pengelola Nama Excel, atau jendela Langsung VBA (lihat nanti).
- Jika ukuran atau lokasi rentang ditentukan hanya saat kode dijalankan, nama akan ditentukan dalam kode VBA, seperti:

```
Set dRange = .....
```

Sisi kanan pernyataan Set akan merujuk ke rentang (berukuran variabel) yang ditentukan pada waktu proses dari kumpulan data (mungkin menggunakan beberapa teknik di bawah ini, seperti CurrentRegion, UsedRange, dan seterusnya). Sisi kiri membuat variabel objek yang sesuai dengan rentang tersebut. Saat merujuk ke rentang tersebut dalam kode, seseorang dapat langsung menggunakan nama variabel (bukan format tanda kutip terbalik), seperti:

```
NRows = dRange.Rows.Count
```

- Kita juga dapat menggunakan metode Add dalam VBA untuk membuat nama Excel saat kode dijalankan (untuk rentang berukuran tetap dan variabel). Misalnya, untuk membuat nama baru yang merujuk ke sel A1:B100 pada lembar kerja Sheet1, kita dapat menggunakan:

```
Names.Add Name:="DataRange", RefersTo:="=Sheet1!$a$1:$B$100"
```

Saat melakukannya, seseorang perlu memastikan bahwa nama tersebut belum ada (misalnya, dengan melampirkan nomor indeksasi yang dilacak dan ditingkatkan setiap kali nama ditambahkan). Oleh karena itu, sering kali pendekatan ini tidak berguna atau semudah pendekatan lainnya.

### Properti CurrentRegion

Properti CurrentRegion dari sel atau rentang adalah rentang dua dimensi terbesar yang dihasilkan dari penyertaan di dalamnya semua sel yang bersebelahan dengan beberapa sel lain di wilayah tersebut. Artinya, ini adalah rentang (dikelilingi oleh sel kosong atau batas fisik Excel) yang akan ditemukan jika seseorang memulai pada sel mana pun dalam rentang tersebut, dan menyertakan setiap sel yang tidak kosong yang bersebelahan dengannya, lalu semua sel yang tidak kosong yang bersebelahan dengannya, dan seterusnya, hingga tidak ada sel baru lagi yang akan disertakan. Jika seseorang tidak mengetahui sintaksnya, seseorang dapat membuatnya dengan merekam makro, menggunakan:

- Urutan menu F5 (GoTo)/Special/Current region.
- Pintasan Ctrl+\* (Ctrl+Shift+8).

Perlu diketahui bahwa proses perekaman umumnya akan menghasilkan rentang yang dipilih, sehingga menghasilkan kode seperti:

```
Selection.CurrentRegion.Select
```

sedangkan secara umum akan lebih efisien untuk tidak memilih objek (tetapi hanya menggunakan bagian CurrentRegion dari sintaksis dalam kode selanjutnya).

### Properti xlCellTypeLastCell

Seseorang dapat menemukan sel terakhir yang berisi data atau format dengan menggunakan:

- Urutan menu F5 (GoTo)/Special/Last cell.
- Pintasan Ctrl+End.

Sekali lagi, makro yang dihasilkan dari proses perekaman secara umum akan memilih sel terakhir, seperti:

```
Sub Macro4 ()
    Range ("A1") .Select
    Selection.SpecialCells(xlCellTypeLastCell) .Select
End Sub
```

Namun, untuk tujuan praktis, seseorang perlu mengekstrak elemen kode yang relevan tanpa

memilih sel apa pun secara eksplisit:

```
With Range("A1")
    Set lcell = .SpecialCells(xlCellTypeLastCell)
End With
```

setelah itu operasi selanjutnya dapat dilakukan, seperti:

Jadi, pada prinsipnya, jika diketahui (karena alasan struktural) bahwa semua data pada lembar kerja berada dalam satu rentang yang bersebelahan, maka baris kode seperti:

```
With Range("A1")
    Set dRange =
    .SpecialCells(xlCellTypeLastCell).CurrentRegion
End With
```

akan membuat variabel objek yang mewakili set data lengkap (termasuk header apa pun). Perhatikan bahwa sel terakhir mungkin kosong jika format atau konten sel tersebut telah diubah selama sesi kerja (misalnya konten dimasukkan lalu dihapus). Jadi, saat menggunakan ini untuk mengidentifikasi rentang data, seseorang mungkin harus berhati-hati apakah operasi tersebut dilakukan dalam kode, dan menyesuaikan kode sebagaimana mestinya.

#### **Nama Lembar Kerja dan Nama Kode**

Secara umum, seseorang mungkin perlu menyatakan lembar kerja mana (dan mungkin buku kerja mana) yang ingin dirujuk selama menjalankan kode. Misalnya, pernyataan seperti Range("A1") tidak menjelaskan dengan jelas lembar kerja mana yang menjadi acuan rentang tersebut. Lembar kerja dapat dirujuk ke:

- Menggunakan nama yang ditetapkan oleh pembuat model saat bekerja di Excel (atau menggunakan default Excel, seperti Sheet1, dsb.). Kode VBA akan menjadi, misalnya:

```
With Worksheets("Sheet1")
    ...
End With
```

- Menggunakan nama kode VBA untuk lembar kerja. Di dalam jendela VBE, lembar kerja dapat diberi nama dengan terlebih dahulu memilihnya di dalam jendela Proyek, dan mengubah nama terkaitnya di dalam jendela Properti. Nama kode ini dapat berbeda dengan nama Excel, sehingga penggunaan nama kode tersebut berarti bahwa kode tersebut akan tetap berfungsi meskipun nama lembar kerja di Excel diubah:

```
With DataSheet
    ...
End With
```

**Properti UsedRange**

UsedRange adalah properti lembar kerja yang menentukan rentang yang digunakannya (ini dapat mencakup baris dan kolom kosong yang telah digunakan dalam beberapa bentuk, termasuk penambahan dan penghapusan konten berikutnya). Sintaksnya mengharuskan lembar kerja telah ditentukan, seperti: Saat menggunakan nama lembar kerja Excel:

```
With Worksheets("Sheet1")
    Set dRange = .UsedRange
End With
```

atau saat menggunakan nama lembar kode:

```
With DataSheet
    Set dRange = .UsedRange
End With
```

Seseorang dapat dengan mudah melihat UsedRange dengan menjalankan kode seperti:

```
With DataSheet
    .UsedRange.Select
End With
```

Ini dapat digunakan dengan cara yang sama dengan properti xlCellTypeLastCell, dengan keuntungan menyatukan beberapa rentang yang tidak bersebelahan. Perhatikan juga bahwa beberapa definisi rentang (yang mungkin diharapkan) tidak ada di VBA (misalnya CurrentRange, ActiveRegion atau ActiveRange tidak ada).

**Properti Sel**

Properti Sel membuat rentang yang ditentukan relatif terhadap titik awal rentang lain. Properti ini dapat digunakan dalam banyak konteks: Untuk mengoperasikan semua sel dalam lembar kerja. Misalnya:

```
With Worksheets("Sheet1").Cells
    .ClearContents
End With
```

atau

```
With Worksheets("Sheet1").Cells
    With .Font
        .Name = "Courier New"
        .Size = 10
    End With
End With
```

```
End With
End With
```

- Untuk mengoperasikan sel yang ditentukan relatif terhadap titik awal lembar kerja. Misalnya, untuk menempatkan nilai 400 di Sel C5, yaitu baris ke-5 dan kolom ke-3 lembar kerja.

```
With Worksheets("Sheet1").Cells(5, 3)
    .Value = 400
End With
```

- Untuk mengoperasikan sel yang ditentukan relatif terhadap rentang lainnya. Misalnya, untuk menempatkan nilai 500 ke dalam sel yang berjarak 5 baris dan 3 kolom dari Sel B2 (yaitu ke dalam Sel D6), seseorang dapat menulis:

```
With Worksheets("Sheet1").Range("B2").Cells(5, 3)
    .Value = 500
End With
```

Perhatikan bahwa pernyataan berikut merujuk pada Sel C7, yaitu sel yang berada pada baris ke-5 dan kolom ke-3 dalam rentang yang dimulai pada Sel A3:

```
Worksheets("Sheet1").Range("A3:B10").Cells(5, 3)
```

- Untuk menentukan rentang sel pada lembar kerja, seperti rentang A1:C5:

```
With Worksheets("Sheet1")
    Range(.Cells(1, 1), .Cells(5, 3)).Value = 800
End With
```

- Untuk menemukan sel pertama di UsedRange pada lembar kerja:

```
With Worksheets("Sheet1").UsedRange
    Set dstart = .Cells(1, 1)
End With
```

Sel juga dapat digunakan dalam konteks nomor sel di lembar kerja atau relatif terhadap sel awal (dari kiri ke kanan lalu atas ke bawah), seperti Sel(1), Sel(2). Secara khusus, ini dapat digunakan untuk menemukan titik awal suatu rentang (mis. Sel(1)). Namun, ada risiko kurangnya kompatibilitas dan kesalahan saat menggunakan properti ini secara lebih luas, karena kode tersebut mungkin awalnya ditulis di Excel 2003 (yang memiliki 256 kolom): Sel(257) akan sesuai dengan Sel A2 di Excel 2003 tetapi dengan sel IW1 di Excel 2007.

### Properti Offset

Properti Offset suatu rentang secara konseptual mirip dengan fungsi OFFSET Excel. Namun, ada sejumlah perbedaan:

- Ini adalah properti suatu rentang, bukan fungsi.
- Rentang referensinya berada di luarnya (bukan bagian integral atau parameter fungsi).
- Tidak ada argumen tinggi dan lebar opsional. Contoh penggunaannya meliputi:
- `Range("A1").Offset(2, 0)` merujuk ke Sel A3, yang diimbangi oleh dua baris dan tidak ada kolom dari Sel A1.
- `Range("A1:B5").Offset(1,2)` adalah rentang C2:D6, yaitu yang diimbangi oleh satu baris dan dua kolom dari rentang referensi.

Secara umum, Offset merupakan alternatif untuk Sel, sehingga keduanya dapat dipertukarkan. Dalam banyak kasus, Offset mungkin lebih intuitif daripada Sel (seperti halnya, dengan fungsi Excel, OFFSET sering kali lebih intuitif daripada INDEX). Di Excel, fungsi OFFSET bersifat Volatile, dan karenanya harus dihindari kecuali tidak ada alternatif nyata yang tersedia (lihat Bab 5 dan Bab 26). Namun, tidak ada batasan seperti itu di VBA, sehingga penggunaan Offset atau Sel sering kali hanya merupakan kasus preferensi pribadi (penulis umumnya memiliki kebiasaan menggunakan Offset).

### Metode Union

Metode Union menyatukan beberapa rentang. Jadi, `Union(Range("B2"), Range("C10"))` hanya merujuk ke sel B2 dan C10, sedangkan `Range("B2", "C10")` akan merujuk ke rentang penuh B2:C10. Berikut ini menetapkan nilai dan format font sel yang tidak bersebelahan:

```
Set uRange = Union(Range("A1"), Range("B7"), Range("C5"))
With uRange
    .Value = 500
    With .Font
        .Name = "Arial"
        .Size = 18
    End With
End With
```

### 13.3 INPUTBOX DAN MSGBOX

InputBox dan MsgBox dapat digunakan untuk mengomunikasikan informasi dasar kepada dan dari pengguna, termasuk untuk mengambil informasi numerik sederhana yang dapat digunakan dalam kode (seperti nomor skenario yang akan dijalankan, atau berapa kali menjalankan loop kalkulasi ulang dalam model simulasi). Dalam kode di bawah ini, baris kode kotak input pertama akan memberikan pesan tetapi input pengguna tidak disimpan, sedangkan baris kode kotak input kedua akan menyimpan input pengguna dalam variabel x, dan kotak pesan akan menampilkan nilai tersimpan ini:



```

Sub MRTakeInput1 ()
    InputBox "Type a Number"
    x = InputBox("Sorry, retype it")
    MsgBox x
End Sub

```

### Application.InputBox

Dalam banyak aplikasi, penting bagi pengguna untuk memasukkan referensi sel (bukan sekadar angka). Misalnya, seseorang mungkin ingin menunjuk ke sel pertama dari kumpulan data, sehingga kode tersebut kemudian dapat mendeteksi kumpulan data lengkap yang bersebelahan dengan sel tersebut (dan yang dapat membentuk definisi data yang akan dimanipulasi). Dalam kasus ini, referensi sel input akan menjadi variabel objek, sehingga pernyataan Set diperlukan, dengan argumen Type digunakan untuk menunjukkan bahwa input yang diberikan adalah referensi sel (dengan menggunakan nilai 8):

```

Set dInputCell
= Application.InputBox("Pilih Sel Apa Pun dalam Rentang Data", Type:= 8)

```

Seseorang kemudian dapat menentukan rentang data sebagai (misalnya) wilayah saat ini yang dikaitkan dengan sel input tersebut:

```
Set dFullRange = dInputCell.CurrentRegion
```

dan seseorang dapat menemukan sel pertama dari rentang tersebut menggunakan:

```
Set dStartCell = dFullRange.Cells(1,1)
```

### Menentukan Rentang Multisel

Jika memungkinkan, biasanya lebih efisien (yaitu kode akan berjalan lebih cepat) untuk bekerja dengan rentang multisel, daripada sel-sel individual dalam rentang tersebut. Misalnya, untuk menempatkan nilai yang sama di semua sel rentang A1:C5, kode seperti berikut dapat digunakan:

```

With Worksheets("Sheet1")
    Set dRangel = Range(.Cells(1, 1), .Cells(5, 3))
    dRangel.Value = 100
End With

```

Demikian pula, untuk menempatkan nilai yang sama dalam rentang A12:C16, hal berikut dapat digunakan:

```

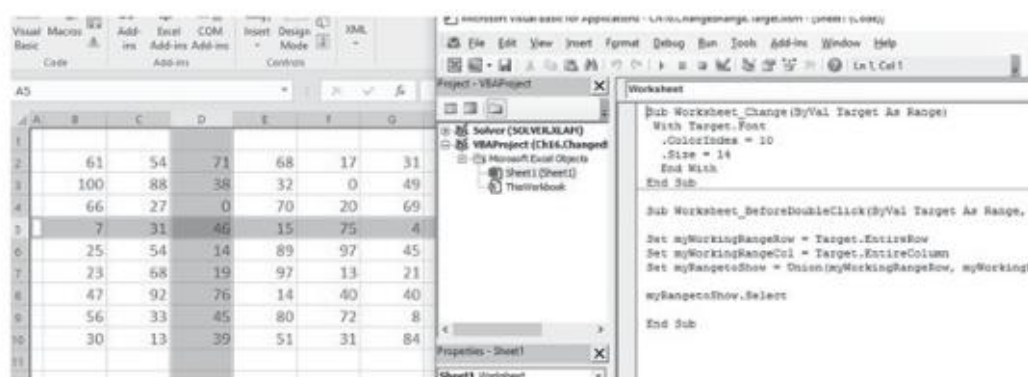
With Worksheets("Sheet1").Range("A1")
    Set dRangel = Range(.Offset(0, 0), .Offset(4, 2))

```

```
dRange1.Value = 200
End With
```

### Menggunakan Target untuk Bereaksi terhadap Peristiwa Lembar Kerja

Kata Target adalah kata khusus dalam VBA yang digunakan untuk merujuk ke rentang yang baru saja diubah oleh pengguna (bukan oleh perhitungan ulang Excel). Kata ini dapat digunakan untuk bereaksi terhadap peristiwa yang terjadi dalam lembar kerja (sehingga makro dijalankan saat peristiwa tersebut terjadi). Kode yang berisi kata ini harus ditempatkan dalam modul kode lembar kerja tempat seseorang ingin mendeteksi perubahan tersebut (bukan dalam modul kode umum), dan juga menggunakan subrutin Worksheet\_Change (yang merupakan sintaks khusus untuk kode yang akan dijalankan saat peristiwa lembar kerja tersebut terjadi). Sebagai contoh, kode berikut akan memastikan bahwa setiap perubahan yang dibuat pada lembar (seperti entri data, rumus, atau teks) akan ditampilkan dalam warna hijau dan dengan ukuran font 14.



**Gambar 13.1** Menyorot Baris dan Kolom Sel yang Dipilih

```
Sub Worksheet_Change(ByVal Target As Range)
    With Target.Font
        .ColorIndex = 10
        .Size = 14
    End With
End Sub
```

Demikian pula, kode berikut akan memastikan bahwa seluruh baris dan seluruh kolom sel disorot ketika seseorang mengklik dua kali di atasnya (lihat Gambar 29.1):

```
Sub Worksheet_BeforeDoubleClick(ByVal Target As Range, Cancel
As Boolean)

    Set myWorkingRangeRow = Target.EntireRow
    Set myWorkingRangeCol = Target.EntireColumn
```

```
Set myRangetoShow = Union(myWorkingRangeRow,  
myWorkingRangeCol)  
myRangetoShow.Select  
End Sub
```

### **Menggunakan Target untuk Bereaksi terhadap Peristiwa Buku Kerja**

Untuk mendeteksi dan bereaksi terhadap perubahan di mana pun dalam buku kerja, seseorang dapat menggunakan subrutin `Workbook_SheetChange`, yang harus ditempatkan dalam modul `ThisWorkbook`, misalnya:

```
Sub Workbook_SheetChange(ByVal Sh As Object, ByVal Target As  
Range)  
str = "You just changed cell " & Target.Address & " of " &  
Sh.Name & " to " & Target.Value  
MsgBox str  
End Sub
```

## BAB 14

### MENGENDALIKAN EKSEKUSI

Bab ini menjelaskan elemen-elemen kunci yang berkaitan dengan pendefinisian dan pengendalian langkah-langkah yang harus diikuti oleh kode saat dijalankan. Kami membahas berbagai topik termasuk sintaksis dan perintah inti, dan memberikan beberapa contoh pilihan dari topik-topik tersebut.

#### 14.1 KOTAK INPUT DAN KOTAK PESAN

Penggunaan `InputBox`, `MsgBox`, dan `Application.InputBox` telah disebutkan dalam bab terakhir, sehingga tidak dibahas lebih lanjut di sini secara terperinci. Namun, perlu dicatat bahwa penggunaan pendekatan ini berarti bahwa eksekusi kode dihentikan hingga beberapa umpan balik diberikan oleh pengguna, yang dapat merepotkan jika seseorang ingin menyelesaikan kode agar berjalan secara otomatis dan tanpa intervensi.

##### Perulangan For...Next

Penggunaan perulangan `For...Next` adalah pendekatan paling dasar dan penting untuk mengotomatiskan operasi berulang yang melibatkan kalkulasi. Sintaksis inti biasanya berbentuk seperti:

```
For i = 1 To 1000
    '...Code to be executed (that involves i)
Next i
```

Atau

```
For i = 1 To Range("NLoops")
    '...Code to be executed (that involves i)
Next i
```

Ini menciptakan proses perulangan di mana variabel indeksasi `i` dimulai dengan nilai 1 dan ditingkatkan secara default sebesar 1 setiap kali langkah perulangan dieksekusi (dan karenanya mengambil nilai 1, 2, secara berurutan). Beberapa poin dasar yang perlu diperhatikan adalah:

- Pernyataan `Next` diperlukan untuk menentukan pada titik mana operasi harus mulai diulang dalam loop, alih-alih beralih ke langkah kode yang dimaksudkan untuk dijalankan setelah loop.
- `i` setelah pernyataan `Next` bersifat opsional, tetapi penyertaannya membuat kode lebih mudah dibaca dan karenanya tidak mudah mengalami kesalahan. Terutama jika terdapat beberapa loop atau loop yang tertanam, "penutupan" loop dengan cara ini penting.

- Kode dalam loop akan (hampir) selalu bergantung pada nilai nomor indeksasi *i* dalam beberapa hal.
- Variabel indeksasi (di sini: *i*) pada dasarnya dapat diberi nama apa pun (yang valid, tidak dicadangkan). Tentu saja, disarankan dan merupakan praktik umum bagi nama variabel untuk memberikan beberapa indikasi mengenai sifat variabel, sehingga nama seperti *i* atau *iCount* dan seterusnya sering digunakan untuk bilangan bulat (seperti *j*, *k*, *l*, *m* dan *n*), dengan *x*, *y*, *z* sering digunakan untuk variabel yang dapat mengambil nilai apa pun.

Dalam beberapa kasus, penambahan selain 1 diperlukan, misalnya:

- For *i* = 1 To 1000 Langkah 2 akan mengulang melalui nilai untuk *i* dari 1, 3, 5, 7, . . . , 999.
- For *i* = 1000 To 1 Langkah -1 akan mengulang melalui nilai 1000, 999, 998, . . . 1.

### **For Each. . . In. . . Next (Untuk masing-masing. . . Dalam. . . Berikutnya)**

Perulangan For Each...In...Next dapat digunakan untuk mengotomatiskan pengulangan operasi dengan mengerjakan item dalam koleksi objek (daripada melalui serangkaian bilangan bulat atau serangkaian indeksasi terstruktur saat menggunakan perulangan For...Next). Sintaksis yang umum adalah:

```
For Each mySheet In Worksheets
    '...Code to be executed (that involves mySheet)
Next mySheet
```

Perhatikan bahwa nama *mySheet* di atas hanyalah variabel indeksasi, dan yang (umumnya) akan digunakan dalam perulangan untuk menentukan objek tempat operasi akan dilakukan (seperti *mySheet.Delete*); sekali lagi, nama tersebut pada dasarnya bersifat arbitrer (dalam batasan validitas nama, seperti tidak ada spasi, dan tidak ada kata yang dicadangkan). Tentu saja, masuk akal dari perspektif transparansi untuk menggunakan nama yang menunjukkan sifat objek dalam konteks tersebut. Penggunaan jenis perulangan ini meliputi:

- Untuk menghapus lembar kerja tertentu dalam buku kerja, seperti yang namanya dimulai dengan kata tertentu atau pengenal lain.
- Untuk memformat ulang semua bagan (*ChartObjects*) dalam buku kerja.
- Untuk membuat daftar nama semua lembar kerja dalam buku kerja.

Perlu dicatat juga bahwa item dalam koleksi objek dikerjakan dalam urutan numerik implisitnya (meskipun urutan tersebut tidak langsung terlihat oleh pemodel atau pengguna), sama seperti perulangan For...Next yang menggunakan langkah positif satu secara default. Misalnya, kode seperti:

```
For Each nm In Names
```

```
MyStrName = Range(nm).Name.Name
Next nm
```

akan bekerja melalui nama rentang bernama dalam buku kerja. Perhatikan bahwa kata Nama diperlukan dua kali dalam kode di atas: karena Nama mengacu pada kumpulan rentang bernama, Nama>Nama mengacu pada objek nama tertentu (yang memiliki beberapa properti, termasuk namanya) dan dengan demikian Nama>Nama>Nama mengacu pada nama.

### **If . . Then (Jika . . Lalu)**

Pernyataan Jika...Lalu pada dasarnya cukup jelas, menciptakan situasi di mana kode melakukan operasi yang berbeda tergantung pada beberapa kondisi. Bentuk yang paling sederhana pada dasarnya adalah:

```
If ConditionTest Then
    '... first possible set of instructions
Else
    '... second set of instructions
End If
```

Perhatikan bahwa ada beberapa kasus di mana operasi hanya perlu dilakukan jika suatu kondisi terpenuhi, dan tidak diperlukan operasi apa pun jika kondisinya tidak terpenuhi. Dalam kasus seperti itu, seseorang dapat mempertimbangkan untuk tidak menggunakan pernyataan Else. Namun, secara umum lebih transparan untuk menyatakan secara eksplisit bahwa tidak diperlukan operasi apa pun; ini dapat dicapai dengan menggunakan pernyataan Else tetapi dengan baris komentar sederhana setelahnya. Jika ada beberapa kondisi yang harus diuji, pernyataan Elself dapat digunakan, sebelum satu Else:

```
If ConditionTest-1 Then
    '... first possible set of instructions
ElseIf ConditionTest-2 Then
    '... second possible set of instructions
ElseIf ConditionTest-n Then
    '... nth possible set of instructions
Else
    '... final possible set of instructions
End If
```

### **Select Case. . End Select (Pilih Kasus. . Akhiri Pilih)**

Ini dapat digunakan untuk menjalankan salah satu dari serangkaian pernyataan tergantung pada hasil pengujian. Dalam arti tertentu, ini merupakan alternatif untuk pernyataan If..Then, yang lebih transparan dalam beberapa kasus (seperti saat menjalankan skenario), hanya karena sering kali lebih mudah untuk mengidentifikasi dalam kode kasus mana yang akan diterapkan dalam keadaan tertentu. Sintaksis yang umum adalah:

```

ScNo = InputBox("Enter Scenario Number")
Select Case ScNo
Case 1 To 3
    '... Apply the relevant operation (e.g. lookup function
based on ScNo)
Case 4, 5
    '... Apply the relevant operation (e.g. lookup function
based on ScNo)
Case 6 To 10
    '... Apply the relevant operation (e.g. lookup function
based on ScNo)
Case Else
    MsgBox "INVALID CASE NUMBER: RERUN AND CHECK YOUR ENTRY"
End Select

```

## 14.2 GoTo

Pernyataan GoTo juga dapat digunakan untuk menentukan percabangan yang akan diterapkan sebagai hasil evaluasi suatu kondisi (seperti kondisi If..Then atau Case) atau dengan pernyataan On Error untuk bercabang ke prosedur penanganan kesalahan (lihat nanti). Sebagai contoh, berikut ini, jika pengguna telah memberikan entri yang tidak valid ke dalam InputBox, pernyataan Case Else akan diterapkan, yang mengakibatkan kode menggunakan perintah GoTo untuk kembali ke baris kode input skenario di awal pernyataan Select Case.

```

TrySelectCaseAgain:
ScNo = InputBox("Enter Scenario Number")
Select Case ScNo
Case 1 To 3
    '...Apply the relevant operation (e.g. lookup
function based on ScNo)
Case 4, 5
    '...Apply the relevant operation (e.g. lookup
function based on ScNo)
Case 6 To 10
    '...Apply the relevant operation (e.g. lookup
function based on ScNo)
Case Else
    GoTo TrySelectCaseAgain
End Select

```

Perhatikan bahwa sintaksis kunci adalah penggunaan nama (yang sah, tetapi pada dasarnya sewenang-wenang) setelah pernyataan GoTo, dan penggunaan nama tersebut (diikuti oleh titik dua) untuk menunjukkan tempat dalam kode di mana seseorang ingin kembali (tanpa titik

dua, pesan kesalahan akan muncul karena seseorang akan secara efektif membuat variabel tanpa menentukan operasi apa pun yang akan dilakukan padanya, sehingga baris kode akan menjadi tidak lengkap).

### **Do...While/Until...Loop**

Struktur Do...Loop berguna ketika kita ingin mengulang bagian-bagian eksekusi kode sejumlah kali yang tidak terbatas, hingga suatu kondisi terpenuhi.

Perhatikan hal-hal berikut:

- ✓ Struktur ini umumnya digunakan dengan salah satu dari pernyataan While atau Until. While akan terus mengulang selama kondisi masih benar, sedangkan Until akan mengulang hingga kondisi terpenuhi untuk pertama kalinya.
- ✓ Pernyataan While (atau Until) dapat digunakan di awal atau akhir loop, tergantung apakah kita ingin operasi loop dijalankan setidaknya sekali atau tidak. Dengan demikian, dua struktur generik untuk bentuk While adalah:

```
Do While ConditionToCheck
... operation to run ...
Loop
```

dan

```
Do
... operation to run ...
Loop While ConditionToCheck
```

(Penggunaan Until juga memiliki dua struktur generik yang serupa).

Struktur tambahan yang serupa adalah konstruk While...Wend (meskipun sering dianggap kurang fleksibel dan jelas dibanding yang lain):

```
While ConditionToCheck
... Operasi yang dijalankan ...
Wend
```

Contoh penting dari struktur ini adalah dalam penyelesaian referensi sirkular, seperti yang dibahas di Bab 10 jilid I.

Perhatikan bahwa ketika menggunakan struktur loop seperti ini (termasuk dengan penggunaan GoTo untuk memperbaiki data input dalam contoh sebelumnya), ada kemungkinan kondisi tidak pernah terpenuhi, terlepas dari berapa kali loop dijalankan. Oleh karena itu, loop semacam ini mungkin perlu menyertakan angka indeks yang meningkat di setiap iterasi loop, yang akan menghentikan prosedur jika jumlah iterasi yang telah ditentukan terlampaui (mengggunakan pernyataan Exit Sub pada titik yang sesuai dalam kode).

### **Perhitungan dan Kalkulasi**

Topik penting terkait dengan perhitungan ulang di Excel saat macro dijalankan. Ada dua tujuan utama:

1. **Memastikan nilai numerik yang benar dihasilkan selama eksekusi kode.**

Sebuah macro yang ditulis atau direkam di lingkungan dengan pengaturan perhitungan



Excel otomatis (di bawah File/Opsi/Formula) mungkin bergantung pada perhitungan otomatis yang dilakukan oleh Excel pada waktu yang tepat. Macro tersebut dapat menghasilkan hasil yang salah jika dijalankan ketika pengaturan perhitungan diubah ke Manual, karena perhitungan ulang tidak akan terjadi. Beberapa teknik yang dapat digunakan untuk mengurangi kemungkinan hal ini terjadi (dengan memfokuskan kapan perhitungan ulang diperlukan) adalah:

- ❖ Mengubah pengaturan perhitungan Excel ke Manual sebelum proses perekaman.
- ❖ Mengubah pengaturan perhitungan Excel ke Manual sebagai langkah pertama dalam proses perekaman.
- ❖ Menulis dan/atau menguji macro saat perhitungan diatur ke Manual.

## 2. Mengoptimalkan waktu komputasi dengan hanya melakukan perhitungan yang diperlukan

Ini adalah topik yang sangat kompleks: tentu saja, perhitungan ulang sebuah workbook selama eksekusi kode akan meningkatkan waktu komputasi, sehingga sebaiknya perhitungan ulang hanya dilakukan jika diperlukan. Namun, menentukan kapan perhitungan ulang diperlukan bisa cukup rumit, sebagian karena—ketika Excel disetting untuk perhitungan otomatis—beberapa tindakan yang memicu perhitungan ulang mungkin tidak langsung diperkirakan untuk melakukannya (seperti memasukkan data baru, menghapus atau menyisipkan baris, mengganti nama worksheet, menyembunyikan atau menampilkan baris, beberapa tindakan penyaringan, membuat atau memodifikasi rentang yang dinamai, dan sebagainya). Selain itu, hanya bagian-bagian tertentu dari model yang mungkin perlu diperbarui saat macro dijalankan. Misalnya, beberapa bagian mungkin hanya perlu dihitung ulang di awal atau akhir macro (atau serangkaian macro), sementara item lainnya mungkin perlu dihitung ulang di setiap langkah dalam loop.

Pengaturan perhitungan workbook dapat diterapkan dari dalam kode (misalnya, di awal dan akhir) menggunakan sintaks berikut (yang dapat ditetapkan dengan merekam macro saat memilih pengaturan perhitungan):

- ✓ `Application.Calculation: xlManual` untuk perhitungan manual.
- ✓ `Application.Calculation: xlSemiautomatic` untuk perhitungan otomatis, kecuali untuk `DataTables`.
- ✓ `Application.Calculation: xlAutomatic` untuk perhitungan otomatis, termasuk `DataTables`.
- ✓ `Application.Iteration: True` untuk menggunakan perhitungan iteratif guna menyelesaikan referensi sirkular.

Seperti yang disebutkan sebelumnya, dalam banyak kasus, membuat macro yang secara default mengatur workbook pada perhitungan Manual di awal dan kemudian hanya menghitung ketika diperlukan dapat menjadi solusi yang masuk akal. Namun, di sisi lain, ada risiko bahwa beberapa langkah perhitungan yang diperlukan terlewatkan saat melakukan hal ini.

Ketika kita ingin mengontrol perhitungan selama eksekusi kode, beberapa kemungkinan yang tersedia adalah:

- 1) `Calculate` atau `Application.Calculate` (dengan shortcut F9). Ini memanggil mesin perhitungan "cerdas" Excel, yang hanya menghitung item yang telah berubah sejak perhitungan terakhir, fungsi `Volatile`, format kondisional, serta item yang bergantung pada item lainnya atau yang ditandai sebagai memerlukan perhitungan ulang (kadang

disebut item "kotor"). Ini akan menghitung item-item tersebut untuk semua workbook yang terbuka dan akan memperbarui DataTables sesuai kebutuhan jika pengaturan Automatic Except Tables (xlSemiautomatic) digunakan untuk workbook tersebut.

- 2) `Application.CalculateFull` (dengan shortcut `Ctrl+Alt+F9`). Ini memaksa perhitungan ulang semua rumus di semua workbook yang terbuka, termasuk semua DataTables.
- 3) `Application.CalculateFullRebuild` (dengan shortcut `Shift+Ctrl+Alt+F9`). Ini memaksa pembangunan ulang semua dependensi dan perhitungan penuh workbook.

Dalam banyak kasus praktis, biasanya cukup hanya menggunakan perintah `Application.Calculate` di tempat yang tepat dalam kode (seperti ketika data baru dimasukkan atau dihasilkan, dll.). Ini adalah pendekatan perhitungan ulang yang cepat, karena dalam sebagian besar kasus hanya sebagian data input atau beberapa rumus yang berubah di antara perhitungan. (Perhatikan bahwa kata "Calculate" akan muncul di Status Bar Excel jika dibutuhkan pembangunan tingkat yang lebih tinggi, seperti pembangunan ulang penuh, atau jika model mengandung referensi sirkular.)

Beberapa hal tambahan yang perlu diperhatikan terkait perhitungan ulang dan optimasi waktu eksekusi adalah:

- Jika tidak yakin apakah perhitungan ulang diperlukan, sebaiknya lebih baik menggunakan terlalu banyak perhitungan ulang daripada terlalu sedikit.
- Sebagian besar manfaat dalam meningkatkan kecepatan komputasi umumnya diperoleh dari melakukan beberapa perubahan kunci pada kode. Mencapai optimasi yang sempurna mungkin bukan tujuan yang masuk akal (jika kecepatan komputasi sangat penting, mungkin perlu bekerja di luar lingkungan Excel/VBA, dan menggunakan kode yang telah dikompilasi sebelumnya, bukan kode yang dikompilasi saat runtime, seperti yang terjadi pada VBA).
- Waktu eksekusi keseluruhan seringkali sangat dipengaruhi oleh desain dan struktur model yang mendasari, serta kode VBA. Misalnya:
  - ✓ Penggunaan fungsi pencarian yang tidak tepat (seperti `VLOOKUP`) atau penggunaan fungsi Volatile (seperti `OFFSET` dan `INDIRECT`) dapat memperlambat model secara signifikan.
  - ✓ Dalam kode VBA, penggunaan operasi `Copy/Paste` (dan pemilihan objek secara umum) akan jauh kurang efisien dibandingkan dengan penggunaan pernyataan penugasan (atau manipulasi objek secara langsung tanpa memilihnya secara eksplisit).
  - ✓ Penggunaan array VBA daripada sel atau rentang Excel untuk menyimpan data (lihat bagian selanjutnya) atau perhitungan antara dapat menjadi cara yang lebih efisien untuk menghasilkan waktu eksekusi yang optimal. Ini memungkinkan kita untuk lebih jelas melihat kapan Excel melakukan perhitungan ulang atau perlu melakukannya. Excel tidak akan menghitung ulang selama melakukan operasi yang hanya melibatkan nilai-nilai dalam array (sementara operasi yang setara yang dilakukan pada tabel data dalam rentang Excel mungkin memaksa perhitungan ulang setiap kali sebuah data di rentang tersebut diubah).
- Secara prinsip, kita dapat menyusun model berdasarkan bagian-bagian yang perlu dihitung ulang pada saat yang sama, sehingga bagian-bagian model dapat dihitung ulang hanya ketika benar-benar diperlukan (seperti worksheet atau rentang tertentu). Ini akan menjadi persyaratan jika tujuan kita adalah mencapai efisiensi komputasi maksimal, meskipun dalam beberapa kasus, perubahan kecil pada model pun dapat

membuat perbedaan.

Saat kode sedang dijalankan, kita bisa menggunakan kode seperti berikut:

```
vb
With AnalysisSheet
    .EnableCalculation = False
End With
```

Dalam kasus seperti ini, menjelang akhir eksekusi kode, perhitungan untuk worksheet tersebut perlu direset dan dilakukan ulang:

```
vb
With AnalysisSheet
    .EnableCalculation = True
    .Calculate
End With
```

(Misalnya, kode pertama digunakan di awal keseluruhan kode, dan kode kedua digunakan di akhir.)

Demikian juga, rentang tertentu dapat dihitung menggunakan metode `Range.Calculate`. Dengan kata lain:

- a. `Application.Calculate` menghitung semua workbook yang terbuka.
- b. `Worksheets("Sheet1").Calculate` hanya menghitung worksheet yang ditentukan.
- c. `Worksheets("Sheet1").Rows(2).Calculate` hanya menghitung rentang yang ditentukan (yaitu baris yang ditentukan).

### **Pembaruan Layar (Screen Updating)**

Salah satu cara sederhana untuk mempercepat eksekusi kode (dalam banyak kasus sekitar 30%) adalah dengan mematikan pembaruan otomatis tampilan Excel saat kode dijalankan. Hal ini dapat dilakukan dengan menggunakan perintah berikut di bagian awal kode:

```
Application.ScreenUpdating = False dan di bagian akhir kode:
Application.ScreenUpdating = True
```

Kelemahan utama dari teknik ini adalah bahwa terkadang tampilan yang terlihat dapat membantu dalam proses debugging atau membantu melihat perkembangan eksekusi kode. Terkadang juga, melihat aksi yang sedang dilakukan memberikan rasa nyaman. Oleh karena itu, teknik ini mungkin paling cocok digunakan pada tahap akhir pengembangan kode, setelah pengujian hampir selesai.

### **Mengukur Waktu Eksekusi**

Secara prinsip, waktu yang dibutuhkan untuk menjalankan kode dapat diukur menggunakan fungsi `Timer` dalam VBA, yang diukur dalam detik (fungsi ini tidak memiliki argumen, jadi tanda kurungnya bisa dihilangkan atau dibiarkan kosong). Kode untuk mengukur

waktu eksekusi secara umum akan terlihat seperti ini:

```
Starttime = Timer()
... Lakukan perhitungan atau operasi
Endtime = Timer()
RunTime = Endtime - Starttime
```

Beberapa hal yang perlu diperhatikan adalah:

- Fungsi Timer mengukur waktu yang telah berlalu sejak tengah malam, sehingga dapat memberikan hasil yang menyesatkan jika kode dijalankan melewati waktu tersebut. Sebagai alternatif, fungsi Time yang lebih umum dapat digunakan, namun sedikit kurang praktis karena memberikan waktu sistem saat ini dalam satuan hari, sehingga biasanya perlu dikonversi ke detik dengan perkalian yang sesuai.
- Jika aplikasi lain berjalan pada saat yang sama atau komputer terlibat dalam kegiatan pemrosesan lain di latar belakang (misalnya, pemeriksaan virus), hal ini dapat mempengaruhi waktu eksekusi. Sulit untuk menentukan sejauh mana hal ini berperan dalam setiap eksekusi. Meskipun demikian, waktu eksekusi secara keseluruhan (terutama jika diukur beberapa kali) dapat memberikan indikasi yang baik mengenai efektivitas langkah-langkah yang diambil untuk meningkatkan kecepatan dan efisiensi.
- Jika menggunakan pendekatan step-through untuk memeriksa apakah kode berfungsi dengan benar, waktu yang tercatat tidak akan mencerminkan waktu yang sebenarnya jika kode dijalankan secara penuh, karena waktu yang tercatat juga akan mencakup waktu yang dibutuhkan untuk menjalankan kode secara langkah demi langkah.

### Menampilkan Waktu Eksekusi pada StatusBar Excel

Total waktu eksekusi juga dapat ditampilkan pada StatusBar Excel dengan menggunakan kode berikut:

```
Application.StatusBar = "RunTime " & Round(RunTime, 2) & " Seconds"
```

### Menonaktifkan Peringatan

Dalam beberapa kasus, pesan peringatan yang muncul di Excel perlu dinonaktifkan agar eksekusi kode dapat berlangsung sesuai keinginan. Misalnya, ketika mencoba menginstruksikan Excel untuk menghapus sebuah worksheet, biasanya akan muncul peringatan bahwa operasi tersebut tidak dapat dibatalkan, dan kita harus (secara manual) mengonfirmasi bahwa kita benar-benar ingin melanjutkan instruksi tersebut. Dalam VBA, hal ini juga akan menyebabkan penangguhan eksekusi kode pada titik tersebut.

Peringatan default dapat dimatikan dengan menggunakan kode:

```
Application.DisplayAlerts = False
```

Tentunya, secara umum, sebaiknya peringatan ini tetap aktif. Setelah dinonaktifkan dalam bagian kode tertentu untuk alasan spesifik, biasanya peringatan tersebut harus diaktifkan kembali di akhir bagian kode tersebut menggunakan:

```
Application.DisplayAlerts = True
```

### 14.3 MENGAkses FUNGSI WORKSHEET EXCEL

VBA memiliki beberapa fungsi bawaan yang dapat digunakan dalam kode. Sebagai contoh, fungsi Sqr mengembalikan akar kuadrat dari argumennya (setara dengan fungsi Excel SQRT), Log memberikan logaritma natural (setara dengan fungsi Excel LN), dan Rnd mengembalikan angka acak terdistribusi merata antara 0 dan 1 (setara dengan fungsi Excel RAND).

Beberapa hal penting terkait fungsi-fungsi ini:

- Daftar fungsi VBA yang tersedia dapat dilihat dengan mengetikkan VBA. pada jendela kode. Dengan demikian, dapat dilihat bahwa jumlah operasi aritmatika yang tersedia sangat terbatas jika dibandingkan dengan Excel. Sebagai contoh, meskipun terdapat banyak fungsi untuk memanipulasi teks (seperti Left, Mid, IsEmpty), operasi aritmatika dasar yang setara dengan fungsi Excel SUM, MIN, dan MAX tidak ada. Daftar lengkap fungsi matematika dan keuangan yang tersedia dapat diakses dengan mengetikkan VBA.Math. dan VBA.Financial..
- Daftar semua fungsi worksheet yang tersedia dalam VBA dapat dilihat dengan mengetikkan WorksheetFunction. pada jendela kode. Fungsi ini dapat diterapkan dalam VBA dengan mengetikkan WorksheetFunction.Sum(), dan seterusnya. Seringkali lebih praktis untuk membuat variabel objek fungsi worksheet, sehingga fungsi-fungsi individu dapat diakses dengan lebih efisien:

```
Set wsf = Application.WorksheetFunction
Set wsfSum = wsf.Sum
Set wsfCount = wsf.Count
...
```

- Untuk fungsi worksheet yang menggunakan notasi “titik” di Excel, seperti banyak fungsi statistik pada Excel 2010 dan seterusnya, diperlukan tanda garis bawah (\_) ketika diakses dalam VBA (misalnya, WorksheetFunction.StDev\_S).
- Jika baik fungsi VBA maupun fungsi Excel ada untuk perhitungan yang sama (termasuk Sqr, Rnd, Log), maka fungsi VBA harus digunakan.

#### Menjalankan Prosedur dalam Prosedur

Ketika kode menjadi lebih besar, sangat berguna untuk menyusunnya menjadi prosedur terpisah, dengan masing-masing memiliki fungsi yang terbatas dan jelas. Ini akan membutuhkan penggunaan subrutin dan fungsi dalam modul yang sama, workbook yang sama, atau workbook lain.

Saat menggunakan beberapa prosedur, perlu diperhatikan agar cakupan (scope) setiap prosedur sesuai. Cakupan prosedur mendefinisikan ketersediaannya untuk digunakan oleh prosedur lain. Subrutin dan fungsi dapat memiliki salah satu dari dua cakupan berikut:

- **Public:** Prosedur dengan cakupan Public dapat digunakan dalam modul kode umum, tersedia dalam proyek (workbook) tersebut, serta proyek lain yang merujuknya. Public adalah cakupan default untuk prosedur (kecuali untuk prosedur event, yang secara default adalah Private), jadi sebagian besar prosedur tidak memerlukan definisi eksplisit sebagai Public.

- **Private:** Deklarasi prosedur sebagai Private (misalnya dengan Private Sub NameofSub()) berarti prosedur ini hanya tersedia untuk prosedur lain dalam modul yang sama di workbook tersebut, dan tidak untuk seluruh workbook atau modul lain dalam workbook itu. Sebuah subrutin yang dideklarasikan sebagai Private tidak akan muncul di kotak dialog Makro Excel (prosedur lain yang tidak muncul di dialog ini adalah prosedur yang memerlukan argumen, dan prosedur yang ada di dalam add-ins). Begitu juga, fungsi yang dideklarasikan sebagai Private (misalnya dengan Private MyFunction(arg1, arg2)) tidak akan dapat diakses dari Excel, dan hanya tersedia dalam modul kode VBA-nya sendiri. Pernyataan Option Private Module dapat digunakan di awal modul untuk menjadikan semua prosedur dalam modul tersebut Private.

### Eksekusi Prosedur dari Prosedur Lain dalam Workbook yang Sama

Subrutin dan fungsi yang bersifat Public dapat dijalankan dari workbook Excel di mana mereka berada menggunakan metode yang telah dijelaskan sebelumnya (misalnya, menggunakan dialog Makro atau menu Formula/Insert Function).

Ketika menjalankan prosedur dari prosedur lain dalam workbook yang sama, beberapa kemungkinan berikut dapat terjadi:

- ❖ **Subrutin yang Public** dapat dijalankan (begitu juga dengan subrutin Private dalam modul yang sama) dengan:
  - ✓ Menggunakan metode Run, mengetikkan nama prosedur dalam tanda kutip diikuti dengan daftar parameter (tanpa tanda kurung), misalnya: `Run "SubName", arg1, arg2`.
  - ✓ Menggunakan pernyataan Call, mengetikkan argumen yang diperlukan dalam tanda kurung, misalnya: `Call SubName(arg1, arg2)`. Pernyataan Call bisa dihilangkan jika daftar argumen tidak diletakkan dalam tanda kurung, misalnya: `SubName arg1, arg2`. Penggunaan pernyataan Call bisa dianggap lebih baik karena lebih eksplisit menyatakan bahwa kontrol diberikan ke prosedur lain.
- ❖ **Fungsi yang Public** (atau Private dalam modul yang sama) juga dapat diakses dengan cara di atas, meskipun secara umum kita lebih tertarik untuk mengetahui nilai yang dikembalikan oleh fungsi tersebut. Oleh karena itu, metode alternatif atau varian berikut sering digunakan:
  - ✓ Menggunakan variabel untuk mewakili nilai yang dikembalikan oleh fungsi (misalnya: `ValueToUse = MyFunction(arg1, arg2)`).
  - ✓ Menggunakan metode Run, dengan menempatkan nama prosedur dan argumen dalam tanda kurung, misalnya: `ValueToUse = Run("MyFunction", arg1, arg2)`.

### Menjalankan Prosedur Public dari Workbook Lain

Ketika menjalankan prosedur Public dari workbook lain (baik dari Excel atau kode VBA), perlu dibuat referensi ke prosedur tersebut dengan salah satu cara berikut:

- **Menyebutkan nama workbook tempat prosedur berada** sebelum nama prosedur, dan menggunakan metode Run atau metode pengembalian nilai (misalnya:

`Run "Book2.xlsm!SubName" atau`

```
x = Book2.xlsm!MyFunction(arg1, arg2) .
```

- **Membuat referensi ke workbook kedua** menggunakan menu Tools/References di VBE. Dalam hal ini, prosedur dapat diakses tanpa menyebutkan nama workbook, dan workbook yang dirujuk tidak perlu dibuka.

Saat melewati argumen antar prosedur, perlu berhati-hati mengenai apakah ini dilakukan dengan cara ByRef atau ByVal.

### Mengakses Add-ins

Saat menggunakan add-ins (seperti Solver), pertama-tama perlu dibuat referensi dalam VBE ke add-in tersebut. Hal ini dapat dilakukan dengan menggunakan menu Tools/References, lalu mencentang (tick-selecting) kotak yang relevan. Jika kotak add-in tidak terlihat, pastikan bahwa add-in tersebut telah dimuat dengan memeriksa menu Excel Options, kemudian di bawah Options/Add-ins/Manage: Excel Add-ins, pilih Go.

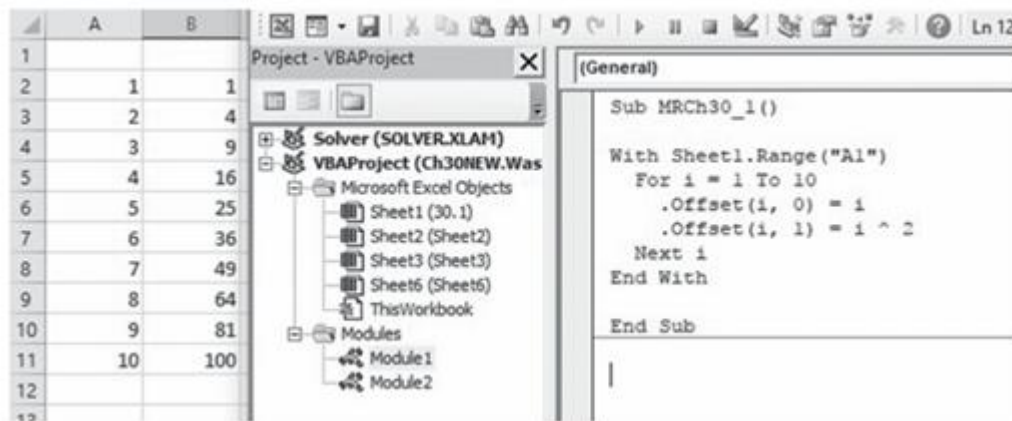
## 14.4 APLIKASI PRAKTIS

Bagian ini menunjukkan beberapa contoh sederhana dari blok bangunan utama yang telah dijelaskan sebelumnya dalam bab ini. Aplikasi yang lebih lengkap dan kuat akan dibahas dalam bab-bab berikutnya.

### Contoh: Perulangan Numerik

Salah satu teknik yang sering digunakan adalah bekerja melalui elemen-elemen individual dalam suatu rentang dan menulis nilai yang berbeda di setiap sel. Gambar 14.1 menunjukkan kode dan hasil eksekusi, di mana tujuannya adalah untuk menulis angka dari 1 hingga 10, serta kuadratnya, dalam dua kolom berdekatan di Excel. Beberapa hal yang perlu diperhatikan:

- ✓ Dari jendela Project, kita bisa melihat bahwa nama worksheet dalam workbook adalah "30.1", sementara nama kode (code name)-nya adalah Sheet1. Nama kode ini digunakan untuk merujuk ke worksheet, daripada menggunakan Worksheets("30.1"), sehingga kode tetap berfungsi meskipun nama worksheet diubah dalam workbook.
- ✓ Pernyataan With...End With digunakan untuk memastikan bahwa semua operasi dilakukan pada Cell A1 dari worksheet yang ditentukan.
- ✓ Properti Offset dari rentang digunakan untuk bergerak ke bawah baris dan melintasi kolom; indeks perulangan i digunakan untuk mewakili jumlah baris yang akan dipindahkan, dan offset kolom (0 dan 1) ditulis langsung dalam kode. Properti Cells bisa digunakan sebagai pengganti dalam hal ini.



Gambar 14.1 Contoh Kode Untuk Perulangan For...Next

**Contoh: Membuat Daftar Nama Semua Lembar Kerja di Buku Kerja**  
Daftar semua nama lembar kerja dalam buku kerja dapat dibuat dengan menggunakan kode berikut, dengan memperhatikan hal-hal berikut:

- ❖ Konstruksi For Each... Next digunakan untuk mengulang koleksi lembar kerja dalam buku kerja (dikenal sebagai *Worksheets*).
- ❖ Variabel iCount digunakan sebagai indeksasi untuk memastikan bahwa nama-nama lembar kerja ditulis secara berurutan di Kolom D worksheet dengan nama kode *Sheet1*. Variabel ini bertambah 1 setiap kali perulangan dilakukan.

Berikut adalah contoh kodenya:

```
iCount = 0
For Each ws In Worksheets
    With Sheet1.Range("D1")
        .Offset(iCount, 0) = ws.Name
    End With
    iCount = iCount + 1
Next ws
```

Beberapa hal yang perlu diperhatikan:

- **Koleksi lembar kerja** dapat diakses menggunakan kode seperti `Workbooks.Item(3)` atau cukup `Workbooks(3)`. Teknik ini juga bisa digunakan untuk mengakses elemen individual dari suatu rentang (misalnya `Range("DataSet")(i)` yang setara dengan `Range("DataSet").Cells(i, 1).Value`).
- Namun, dalam konteks koleksi objek (seperti *Worksheets*), biasanya kita tidak mengetahui berapa banyak item yang ada di dalam koleksi tersebut, sehingga merujuk langsung ke item tertentu bukanlah cara yang paling tepat, kecuali jika jumlah totalnya sudah ditentukan terlebih dahulu.

```
With ThisWorkbook
    N = Worksheets.Count
End With
```

Dengan menggunakan teknik ini, Anda dapat membuat daftar otomatis semua nama lembar kerja yang ada dalam workbook tertentu.



**Contoh: Menambahkan Lembar Kerja Baru ke Workbook**

Koleksi Worksheets juga dapat digunakan untuk menambahkan lembar kerja ke workbook aktif dan memberi nama lembar kerja baru tersebut. Misalnya, jika lembar kerja tersebut akan berisi hasil perhitungan, lembar kerja tersebut dapat ditambahkan segera setelah eksekusi makro dimulai, menggunakan kode berikut:

```
vba
Worksheets.Add.Name = "Results"
```

Proses ini juga bisa digeneralisasi, sehingga banyak lembar kerja Results dapat ditambahkan dan dinamai sesuai dengan jumlah lembar kerja yang sudah ada di dalam workbook:

```
vba
With ThisWorkbook
    N = Worksheets.Count
    Worksheets.Add.Name = "Results" & N
End With
```

Kode ini bisa juga dibagi menjadi langkah-langkah terpisah untuk menambahkan lembar kerja baru, lalu mengganti namanya:

```
vba
With ThisWorkbook
    N = Worksheets.Count
    Set Sheetnew = Worksheets.Add
    Sheetnew.Name = "Results" & N
End With
```

Penting untuk dicatat bahwa baris kode ini menambahkan angka pada nama lembar kerja Results yang bergantung pada jumlah lembar kerja dalam workbook, bukan berdasarkan jumlah lembar kerja Results yang sudah ada (sehingga lembar kerja Results pertama bisa dinamai misalnya Results3). Untuk memberi nama lembar kerja berdasarkan jumlah lembar kerja Results yang sudah ada, perlu dilakukan penghitungan lembar kerja tersebut. Sebagai contoh, variabel (seperti iCount pada contoh di atas) dapat digunakan, yang diinisialisasi dengan nilai 0 dan kemudian ditambah 1 setiap kali lembar kerja Results baru ditambahkan.

Alternatifnya, loop For Each dapat digunakan untuk memeriksa semua lembar kerja dalam workbook dan menghitung berapa banyak yang dimulai dengan kata "Results", misalnya dengan memeriksa apakah tujuh huruf pertama dari nama tersebut adalah "Results" (dan menambah variabel iCount jika demikian). Walaupun pendekatan pertama lebih efisien dalam hal perhitungan, kedua pendekatan ini umumnya diperlukan. Pendekatan terakhir dapat digunakan untuk menetapkan nilai awal variabel iCount, karena kode mungkin sudah berisi lembar kerja Results, tetapi setelah nilai awal ini ditetapkan, iCount hanya perlu ditambah setiap kali lembar kerja Results baru ditambahkan (atau dihapus).

**Contoh: Menghapus Lembar Kerja Tertentu dari Workbook**

Lembar kerja dapat dihapus dari workbook menggunakan kode VBA, namun secara umum, Anda akan ingin mematikan tampilan pemberitahuan Excel (seperti yang telah

disebutkan sebelumnya). Selain itu, Anda hanya ingin menghapus lembar kerja tertentu, bukan semuanya. Kode berikut menghapus semua lembar kerja yang namanya dimulai dengan kata "Results". Kode ini menggunakan fungsi Left (VBA) dan properti Name untuk mengambil tujuh karakter pertama dari nama lembar kerja, serta fungsi UCase (VBA) yang setara dengan fungsi UPPER di Excel untuk mengubah karakter tersebut menjadi huruf kapital, guna keperluan perbandingan (agar memastikan semua lembar kerja tersebut teridentifikasi terlepas dari penggunaan huruf kapital, huruf kecil, atau campuran).

```
Application.DisplayAlerts = False
With ThisWorkbook
    For Each ws In Worksheets
        If UCase(Left(ws.Name, 7)) = "RESULTS" Then ws.Delete
    Next ws
End With
Application.DisplayAlerts = True
```

#### Penjelasan Kode:

- ❖ Application.DisplayAlerts = False: Menonaktifkan pemberitahuan atau peringatan Excel, sehingga proses penghapusan lembar kerja dapat dilakukan tanpa konfirmasi.
- ❖ For Each ws In Worksheets: Looping melalui setiap lembar kerja dalam workbook.
- ❖ UCase(Left(ws.Name, 7)) = "RESULTS": Mengambil tujuh karakter pertama dari nama lembar kerja dan mengubahnya menjadi huruf kapital, kemudian memeriksa apakah karakter tersebut cocok dengan kata "RESULTS".
- ❖ ws.Delete: Menghapus lembar kerja yang memenuhi kondisi tersebut.
- ❖ Application.DisplayAlerts = True: Mengaktifkan kembali pemberitahuan Excel setelah proses selesai, sehingga pengguna dapat menerima peringatan atau konfirmasi di masa mendatang.

Kode ini akan menghapus semua lembar kerja yang namanya diawali dengan kata "Results", tanpa memperhatikan kapitalisasi huruf.

#### Contoh: Menyegarkan PivotTable, Memodifikasi Grafik, dan Bekerja dengan Koleksi Objek Lain

Penggunaan koleksi objek juga dapat sangat efektif untuk memodifikasi atau memanipulasi tipe objek lain di Excel. Sebagai contoh, daftar PivotTables (yaitu objek PivotTable) dalam sebuah worksheet dapat diidentifikasi menggunakan kode berikut:

```
Worksheets("NamaWorksheet").PivotTables
```

Kode ini dapat menjadi dasar untuk operasi lainnya, seperti:

```
NPvtTbls = Worksheets("SummarySheet").PivotTables.Count
```

Untuk menyegarkan (satu atau semua) PivotTable, cara paling mudah adalah dengan merekam proses menyegarkan satu PivotTable (untuk membentuk sintaks dasar) dan menyesuaikan kode sesuai kebutuhan. Kode yang dihasilkan bisa seperti ini:

```
For Each pt In ActiveSheet.PivotTables
```

```

    pt.RefreshTable
Next pt

```

Jika hanya ingin menyegarkan PivotTable tertentu saja, kode dapat ditulis seperti ini:

### Menyegarkan PivotTable Tertentu

```

For Each pt In ActiveSheet.PivotTables

```

```

    Select Case pt.Name
        Case "PivotTable1", "PivotTable3", "PivotTable6"
            pt.RefreshTable
        Case Else
            ' Tidak ada tindakan
    End Select
Next pt

```

Untuk PivotTables yang tersebar di beberapa worksheet, kita juga dapat menggunakan loop untuk melalui setiap worksheet dalam workbook dan menyegarkan PivotTables pada setiap worksheet saat melakukannya:

```

For Each ws In ActiveWorkbook.Worksheets
    For Each pt In ws.PivotTables
        pt.RefreshTable
    Next pt
Next ws

```

### Penjelasan Kode:

- `ActiveSheet.PivotTables`: Mengakses semua PivotTable yang ada di worksheet aktif.
- `pt.RefreshTable`: Menyegarkan PivotTable tertentu.
- `Select Case pt.Name`: Menyaring PivotTable yang ingin disegarkan berdasarkan namanya.
- `For Each ws In ActiveWorkbook.Worksheets`: Looping melalui setiap worksheet dalam workbook aktif untuk mengakses dan menyegarkan PivotTables yang ada di masing-masing worksheet.

Dengan teknik ini, Anda dapat dengan mudah memanipulasi dan menyegarkan PivotTable baik secara selektif maupun di seluruh workbook.

## DAFTAR PUSTAKA

- Aiken, M., & Brooks, R. (2020). *Advanced financial risk management: Tools and techniques for integrated credit risk and interest rate risk management* (2nd ed.). Wiley.
- Baetge, M., & Dufresne, R. (2017). *Financial modeling: A practical approach*. Wiley.
- Bekaert, G., & Hodrick, R. J. (2017). *International financial management*. Pearson.
- Benninga, S. (2023). *Financial modeling* (5th ed.). MIT Press.
- Bodie, Z., Kane, A., & Marcus, A. J. (2021). *Investments* (12th ed.). McGraw-Hill Education.
- Bragg, S. M. (2022). *Business ratios and formulas: A comprehensive guide* (4th ed.). Wiley.
- Brealey, R. A., Myers, S. C., & Allen, F. (2019). *Principles of corporate finance* (13th ed.). McGraw-Hill Education.
- Brigham, E. F., & Ehrhardt, M. C. (2023). *Financial management: Theory & practice* (16th ed.). Cengage Learning.
- CFA Institute. (2023). *Financial modeling and analysis: A practitioner's guide*. CFA Institute Research Foundation.
- Cheng, W., & Lee, S. (2020). Financial modeling for risk analysis: A practical approach. *Journal of Financial Modeling*, 18(2), 233-250. <https://doi.org/10.1016/j.jfm.2020.05.003>
- Damodaran, A. (2012). *Investment valuation: Tools and techniques for determining the value of any asset* (3rd ed.). Wiley.
- Damodaran, A. (2022). *Investment valuation: Tools and techniques for determining the value of any asset* (4th ed.). Wiley.
- Day, A. L. (2021). *Mastering financial modeling: A professional's guide to building financial models in Excel*. Pearson.
- DeMello, A. T. (2021). *Excel financial modeling for beginners*. Prentice Hall.
- Fairhurst, D. S. (2019). *Financial modeling in Excel For Dummies*. Wiley.
- Fernandez, P. (2016). *Valuation and financial modeling*. *European Financial Management*, 22(4), 567-586. <https://doi.org/10.1111/j.1468-036X.2016.12103.x>

- Fischer, L., & Heinkel, R. (2019). *Corporate financial modeling and decision-making*. Oxford University Press.
- Frykman, D., & Tolleryd, J. (2019). *Corporate valuation: An easy guide to measuring value*. Pearson.
- Gallo, M. A., & Conlon, M. E. (2018). *Financial modeling in practice: A guide to current practice*. Wiley.
- Godek, P., & Kelliher, C. (2021). *Financial modeling for mergers and acquisitions*. Routledge.
- Graham, B., & Dodd, D. L. (2008). *Security analysis: Sixth edition (2nd ed.)*. McGraw-Hill.
- Green, S., & Dodd, P. (2017). *Financial modeling for business owners and entrepreneurs*. Wiley.
- Higgins, R. C. (2022). *Analysis for financial management (13th ed.)*. McGraw-Hill Education.
- Ho, J., & Lee, S. B. (2020). *Financial modeling and simulation: A case study approach*. Palgrave Macmillan.
- Hull, J. C. (2016). *Risk management and financial institutions (4th ed.)*. Wiley.
- Hull, J. C. (2017). *Options, futures, and other derivatives (9th ed.)*. Pearson.
- Ibbotson, R. G., & Kaplan, P. D. (2016). *Financial models in practice*. Wiley.
- Jackson, C. (2014). *Applied financial modeling and risk analysis*. Springer.
- Jorion, P. (2010). *Financial risk management: A practical approach*. Wiley.
- Koller, G., & Peterson, L. (2016). *Financial modeling for decision making*. McGraw-Hill Education.
- Koller, T., Goedhart, M., & Wessels, D. (2020). *Valuation: Measuring and managing the value of companies (7th ed.)*. Wiley.
- Lister, M. A. (2015). *The complete guide to financial modeling*. Financial Times Prentice Hall.
- Mayes, T. R., & Shank, T. M. (2021). *Financial analysis with Microsoft Excel (9th ed.)*. Cengage Learning.
- McKinney, J. (2015). *Financial modeling using Excel and VBA*. Wiley.
- Mehta, A. (2018). *Financial risk modeling and management*. Springer.
- Morris, G. (2021). *Practical financial optimization: The development of a financial planning model*. Wiley.

- Neveu, R. (2023). *Fundamentals of financial management* (11th ed.). Cengage Learning.
- Penman, S. H. (2021). *Financial statement analysis and security valuation* (6th ed.). McGraw-Hill Education.
- Perold, A. F. (2017). *Modeling financial markets: A practitioner's guide*. Pearson.
- Petty, J. W., & Keown, A. J. (2013). *Financial management: Principles and applications* (12th ed.). Pearson.
- Pignataro, P. (2019). *Financial modeling and valuation: A practical guide to investment banking and private equity* (2nd ed.). Wiley.
- Pinto, J. E., & Henry, E. (2015). *Financial modeling and valuation: A practical guide to investment banking and private equity* (3rd ed.). Wiley.
- Pratt, S. P., & Grabowski, R. J. (2020). *Cost of capital: Applications and examples* (5th ed.). Wiley.
- Prober, L. M. (2021). *Financial modeling for business owners and entrepreneurs: Developing Excel models to raise capital, increase cash flow, and improve operations*. Apress. <https://doi.org/10.1007/978-1-4842-6375-6>
- Rees, M. (2019). *Financial modeling in practice: A concise guide for intermediate and advanced levels*. John Wiley & Sons.
- Reilly, F. K., & Brown, K. C. (2012). *Investment analysis and portfolio management* (10th ed.). Cengage Learning.
- Roberts, G. (2019). *Financial modeling and analysis with Excel*. Wiley.
- Ross, S. A., Westerfield, R. W., & Jaffe, J. F. (2022). *Corporate finance* (12th ed.). McGraw-Hill Education.
- Ross, S. A., Westerfield, R. W., & Jordan, B. D. (2016). *Essentials of corporate finance* (9th ed.). McGraw-Hill Education.
- Schwartz, E. S., & Trigeorgis, L. (2004). *Real options and financial practice*. Wiley.
- Sengupta, C. (2020). *Financial analysis and modeling using Excel and VBA* (2nd ed.). Wiley.
- Sharpe, W. F., Alexander, G. J., & Bailey, J. V. (2014). *Investments* (10th ed.). Pearson.
- Smith, J. R., & Taylor, R. A. (2018). *Financial models for risk management and portfolio optimization*. Springer.

- Suozzo, P., Cooper, S., Sutherland, G., & Deng, Z. (2020). *The banker's guide to financial modeling*. Risk Books.
- Tirole, J. (2017). *Financial regulation: An institutional approach*. MIT Press.
- Titman, S., & Martin, J. D. (2022). *Valuation: The art and science of corporate investment decisions* (4th ed.). Pearson.
- Tjia, J. S. (2018). *Building financial models: The complete guide to designing, building, and applying projection models* (3rd ed.). McGraw-Hill Education.
- Triantis, A. J. (2015). *Financial modeling: A comprehensive guide to modeling financial risk*. Springer.
- Van Horne, J. C., & Wachowicz, J. M. (2012). *Fundamentals of financial management* (14th ed.). Pearson.
- Verma, P., & Chordia, A. (2018). *Financial modeling and data analysis*. Wiley.
- Walkenbach, J. (2013). *Excel 2013 power programming with VBA*. Wiley.
- West, R. (2015). *The essentials of financial modeling in Excel*. Wiley.
- Wiggins, R., & Smith, S. (2017). *Applied financial modeling and optimization techniques*. Wiley.
- Williams, J. (2015). *Financial modeling for analysts and corporate finance*. Wiley.
- Willmott, P., & Taleb, N. (2018). *Risk and financial modeling in practice*. Wiley.
- Winston, W. L. (2023). *Microsoft Excel data analysis and business modeling* (7th ed.). Microsoft Press.
- Young, S. D., & Taylor, C. M. (2014). *Advanced financial modeling techniques*. McGraw-Hill Education.
- Zakamouline, V., & Koekebakker, S. (2019). *Financial modeling of energy markets: A stochastic approach*. Springer. <https://doi.org/10.1007/978-3-030-18087-1>
- Zeisler, A. (2021). *Financial modeling and analysis in Excel: A case study approach*. Business Expert Press.
- Zutter, C. J., & Harrington, R. J. (2014). *Corporate financial management* (3rd ed.). Pearson.

# PRAKTIK MODEL KEUANGAN DENGAN EXCEL & VBA

Dr. Agus Wibowo, M.Kom, M.Si, MM.

## BIO DATA PENULIS



Penulis memiliki berbagai disiplin ilmu yang diperoleh dari Universitas Diponegoro (UNDIP) Semarang. dan dari Universitas Kristen Satya Wacana (UKSW) Salatiga. Disiplin ilmu itu antara lain teknik elektro, komputer, manajemen dan ilmu sosiologi. Penulis memiliki pengalaman kerja pada industri elektronik dan sertifikasi keahlian dalam bidang Jaringan Internet, Telekomunikasi, Artificial Intelligence, Internet Of Things (IoT), Augmented Reality (AR), Technopreneurship, Internet Marketing dan bidang pengolahan dan analisa data (komputer statistik).

Penulis adalah pendiri dari Universitas Sains dan Teknologi Komputer (Universitas STEKOM ) dan juga seorang dosen yang memiliki Jabatan Fungsional Akademik Lektor Kepala (Associate Professor) yang telah menghasilkan puluhan Buku Ajar ber ISBN, HAKI dari beberapa karya cipta dan Hak Paten pada produk IPTEK. Sejak tahun 2023 penulis tercatat sebagai Dosen luar biasa di Fakultas Ekonomi & Bisnis (FEB) Universitas Diponegoro Semarang. Penulis juga terlibat dalam berbagai organisasi profesi dan industri yang terkait dengan dunia usaha dan industri, khususnya dalam pengembangan sumber daya manusia yang unggul untuk memenuhi kebutuhan dunia kerja secara nyata.



YAYASAN PRIMA AGUS TEKNIK

### PENERBIT :

YAYASAN PRIMA AGUS TEKNIK  
Jl. Majapahit No. 605 Semarang  
Telp. (024) 6723456. Fax. 024-6710144  
Email : penerbit\_ypat@stekom.ac.id

ISBN 978-623-8642-91-5 (PDF)



9

786238

642915