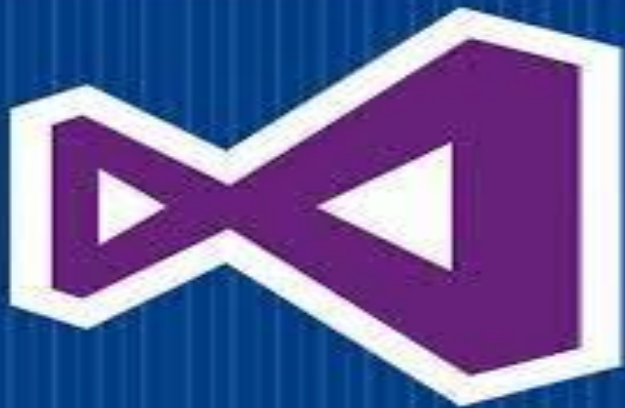




# PEMROGRAMAN VISUAL VB.NET 2008

DARI DASAR SAMPAI DENGAN DATABASE



Visual Basic Dot Net

Microsoft  
.net



YAYASAN PRIMA AGUS TEKNIK

Rusito, S.Kom, M.Kom

# PEMROGRAMAN VISUAL VB.NET 2008

DARI DASAR SAMPAI DENGAN DATABASE

**Rusito, S.Kom, M.Kom**



YAYASAN PRIMA AGUS TEKNIK

Pemrograman Visual VB.NET 2008 Dari Dasar Sampai Dengan Database

**Penulis:**

Rusito, S.Kom, M.Kom

**ISBN :**

**978-623-8120-23-9 (PDF)**

**Editor:**

Indra Ava Dianta, S.Kom., M.T

**Penyunting :**

Danang, S.Kom., M.T

**Desain Sampul dan Tata Letak :**

Yuli Fitrianto, M.Kom

**Penerbit :**

Yayasan Prima Agus Teknik

Redaksi: Jln Majapahit No 605 Semarang

Tlpn. (024) 6723456

Fax . 024-6710144

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

**Distributor Tunggal:**

UNIVERSITAS STEKOM

Jln Majapahit No 605 Semarang

Tlpn. (024) 6723456

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 tertulis dan penerbit.

## KATA PENGANTAR

Dengan menyebut nama Allah SWT yang Maha Pengasih dan Penyayang, kami mengucapkan Bismillahirrahmanirrahim. Kami memanjatkan puji dan syukur kepada-Nya yang telah memberikan rahmat, hidayah, dan berbagai karunia-Nya, sehingga kami dapat menyelesaikan buku ajar ini untuk perkuliahan Pemrograman Visual VB.NET 2008 Dari Dasar Sampai Dengan Database. Semoga buku ajar ini bermanfaat bagi para pembaca dan menjadi sarana untuk meningkatkan ilmu pengetahuan. Aamiin.

Buku ini terdiri dari 12 bab yang masing-masing bab yang isinya mengenai : Bab I Mengenal Visual Basic.Net 2008 Bab ini lebih mengenal dasar tentang VB.NET dan sejarahnya yang merupakan dasar dari perkembangan VB.NET. Bab II Form Dan Objek Kontrol yang mempelajari apa saja yang ada di Visual Basic Net 2008. Bab III Form Objek Kontrol Lanjutan, masih kelanjutan dari bab 2 karena terlalu luas di buat bab 3. Bab IV Tipe Data, Variable, Konstanta, Dan Operator. Bab V Percabangan. Bab VI Perulangan. Bab VII Array. Bab VIII Procedure, Function Dan Module. Bab IX Database. Bab X Exception Handling. Bab XI Keamanan Dan Administrasi Database. Bab XII Konsep Data Warehouse Dan Schema.

Kami telah menyiapkan buku ajar ini dengan sebaik-baiknya dan telah memperoleh bantuan dari berbagai pihak untuk memudahkan pembuatannya. Oleh karena itu, kami ingin mengucapkan terima kasih yang sebesar-besarnya kepada semua pihak yang telah berkontribusi dalam pembuatan buku ajar ini.

Kepada istri tercinta anak-anaku Naya dan Arka yang saya sayangi kalian semua telah memberi inspirasi dan semangat untuk terus berkarya. Keluargaku yang saya sayangi sangat membantu dalam proses pembuatan buku ini dengan terus memberi keyakinan kan selesainya buku ini.

Kami menyadari sepenuhnya bahwa ada kesenjangan dalam struktur kalimat dan tata bahasa dalam buku ajar ini. Oleh karena itu, kami mengharapkan masukan, saran, dan kritik dari pembaca untuk membantu memperbaiki buku ajar ini.

Akhir kata kami berharap semoga buku ajar Pemrograman Visual VB.NET 2008 Dari Dasar Sampai Dengan Database ini dapat memberi manfaat maupun inspirasi yang luas terhadap pembaca.

Semarang, Februari 2023

Penulis

Rusito, S.Kom, M.Kom

## DAFTAR ISI

KATA PENGANTAR / UCAPAN TERIMAKASIH .....	iv
DAFTAR ISI.....	v
BAB I MENGENAL VISUAL BASIC.NET 2008.....	1
BAB II FORM DAN OBJEK KONTROL .....	5
BAB III FORM OBJEK KONTROL LANJUTAN.....	10
BAB IV TIPE DATA, VARIABLE, KONSTANTA, DAN OPERATOR.....	17
BAB V PENCABANGAN.....	27
BAB VI PERULANGAN.....	34
BAB VII ARRAY .....	40
BAB VIII PROCEDURE, FUNCTION DAN MODULE .....	45
BAB IX DATABASE.....	54
BAB X EXCEPTION HANDLING .....	61
BAB XI KEAMANAN DAN ADMINISTRASI DATABASE.....	67
BAB XII KONSEP DATA WAREHOUSE DAN SCHEMA.....	83
DAFTAR PUSTAKA.....	98

# BAB I

## MENGENAL VISUAL BASIC .NET 2008

### 1.1 Tujuan Pembelajaran

- Memahami apa itu platform Microsoft .NET.
- Kenal dekat dengan Visual Basic.NET, pada lingkungan pengembangan terintegrasi (IDE).
- Harus memahami struktur proyek Visual Basic.NET Anda.
- Memahami pada jenis-jenis aplikasi .NET.
- Anda dapat membuat antarmuka pengguna grafis (GUI) menggunakan Visual Basic .NET

### 1.2 Teori Dasar VB.NET

#### Platform .NET Framework

Awalnya dikenal sebagai Next Generation Windows Services (NGWS), Microsoft .NET adalah platform untuk membangun dan menjalankan aplikasi terdistribusi generasi mendatang. Microsoft.NET adalah kerangka kerja pengembangan yang menyediakan antarmuka pemrograman baru untuk layanan Windows dan antarmuka pemrograman aplikasi (API). Microsoft.NET adalah strategi Microsoft untuk menghubungkan sistem, data, dan perangkat sehingga orang dapat berkomunikasi dan berkolaborasi dengan lebih efektif. Teknologi .NET sepenuhnya terintegrasi ke dalam produk Microsoft dan menyediakan kemampuan untuk mengembangkan solusi menggunakan layanan web. Platform Microsoft .NET terdiri dari lima komponen utama, yang terletak di tiga level. Lapisan paling bawah adalah sistem operasi dan lapisan kedua terdiri dari tiga bagian. Lapisan atas adalah Visual Studio .NET.

#### Visual Studio .NET

Microsoft Visual Studio .NET merupakan kumpulan lengkap tools pengembangan untuk membangun aplikasi Web ASP.NET, XML Web Services, aplikasi dekstop, dan aplikasi mobile. Di dalam Visual Studio inilah bahasa-bahasa pemrograman .NET seperti Visual Basic, Visual C++, Visual C# (CSharp), dan Visual J# (JSharp). Semuanya menggunakan lingkungan pengembangan terintegrasi atau IDE yang sama sehingga memungkinkan untuk saling berbagi tools dan fasilitas.

#### Visual Basic .NET

Visual Basic .NET (atau VB.NET) adalah bahasa pemrograman yang dapat digunakan untuk membuat aplikasi .NET di platform Microsoft .NET. Berbeda dengan generasi sebelumnya, Visual Basic versi 6.0 dan sebelumnya yang lebih fokus pada pengembangan aplikasi desktop, Visual Basic .NET memungkinkan pengembang untuk membuat berbagai macam aplikasi, baik aplikasi desktop maupun web. Dengan perkembangan program yang semakin kompleks muncullah Visual Basic .NET (Visual Basic 2008).

#### IDE Visual Basic

IDE (Integrated Development Environment), juga dikenal sebagai desain terintegrasi atau lingkungan debugging, adalah program perangkat lunak komputer yang membantu pemrogram mengembangkan perangkat lunak. Singkatnya, IDE adalah lingkungan pengembangan aplikasi terintegrasi. Itu dilengkapi dengan berbagai alat pendukung atau alat bantu.

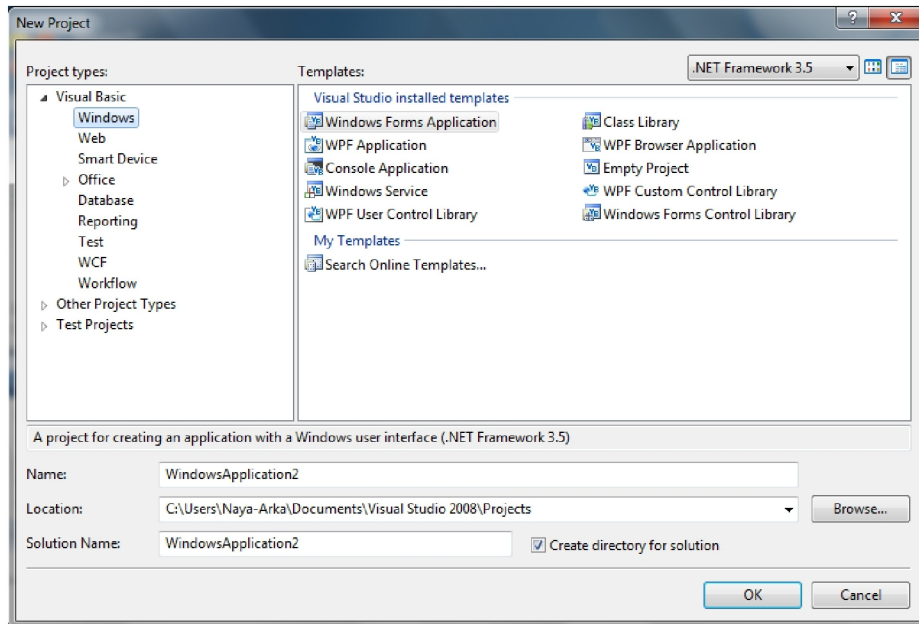
### 1.3 Praktikum

#### Membuat Project Baru

Proyek adalah kerangka kerja aplikasi dasar yang menentukan jenis aplikasi yang akan dibangun.

Selesaikan langkah-langkah berikut untuk mulai membuat proyek:

1. Jalankan **Visual Basic 2008 Express Edition**.
2. Klik menu **File > New Project**, atau melalui ikon **New Project**, atau melalui kombinasi tombol **Ctrl+N**.
3. Pada kotak dialog **New Project**, pilih template **Windows Application**, kemudian klik **OK**.



**Windows** ( Windows Form, Concole, Class Library, WPF dll)

**Web** (ASP Web Application, ASP.Net Web Servis dll)

**Smart Dvice** (Smart Device Project)

**Database** (Access dan SQL Server)

**Report** (Report Application dan Crystal Report)

**WCF** (Windows Communication Foundation) dll.

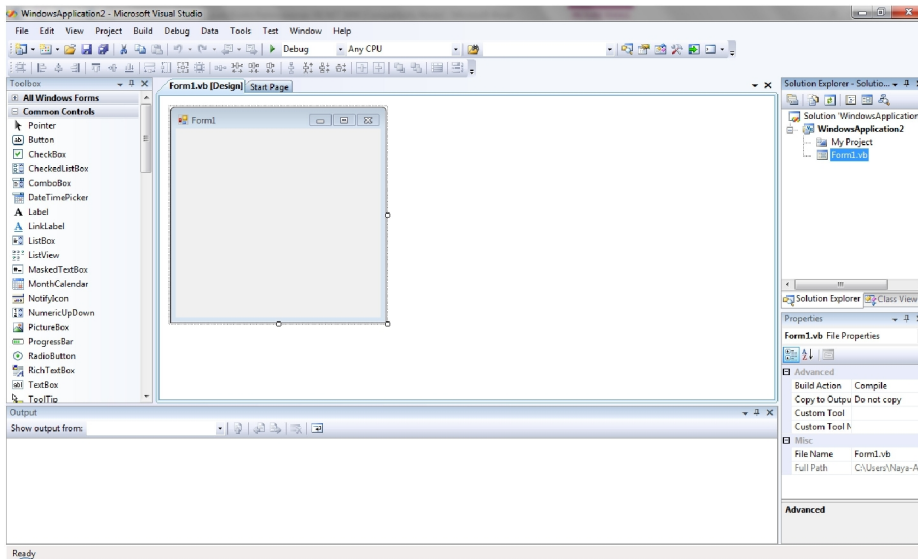
**Windows Application.** Untuk membuat aplikasi-aplikasi Windows yang running secara lokal di komputer user.

**Class Library.** Untuk membuat kelas-kelas atau bagian-bagian reusable yang dapat di-bagi dengan project lainnya.

**Console Application.** Untuk menciptakan aplikasi-aplikasi commandline.

#### Mengenal IDE Visual Basic 2008

Untuk upaya memudahkan penggunaan IDE, maka diharuskan pemahaman yang sempurna mengenai IDE Visual Basic 2008.



Interface	Keterangan
Menu Bar	Menu standar pada VB
Toolbar	Daftar tool yang dipakai untuk running perintah yang sering dipakai
Toolbox	Daftar control yang tambahkan ke dalam program sebagai interface
Form Design	Digunakan untuk mengedit tampilan form serta mengatur posisi control pada form
Solution Explorer	Digunakan untuk mengolah file dan project
Properties	Digunakan untuk mengedit property dari form dan control yang sedang diedit
Error List	Digunakan untuk menampilkan pesan error jika terjadi kesalahan

### Membuat Aplikasi Windows

1. Buat **project baru**, atau lanjutkan project pertama Anda sebelumnya.
2. Tambahkan sebuah kontrol Label ke form, dengan cara mengklik ganda Label di Toolbox, atau men-drag Label ke form. Atur propertinya seperti tabel berikut:

Object	Properties	Nilai
Form1	Name Text	Frmlatihan Latihan Visual Basic
Label	Name	lblText

3. Tambahkan **Event Load** pada form dengan salah satu cara berikut:
  - a. Klik ganda area form kosong.
  - b. Klik area form kosong, kemudian tekan **F7**.
4. Lengkapi kode event handler-nya seperti berikut:

```
Public Class Form1
Private Sub Form1_Load(ByVal sender As Object, ByVal e As
System.EventArgs) Handles Me.Load
' Men-set teks kontrol label
Me.lblText.Text = "Belajar Pemrograman Visual Basic"
End Sub
End Class
```

5. Jalankan aplikasi dengan menekan tombol **F5** (di keyboard), atau melalui ikon Start Debugging di toolbar, atau melalui menu **Debug > Start Debugging**.
6. Simpan aplikasi **Anda**.

### Membuat Aplikasi Console

1. Buat **project baru**, pilih template **Console Application**, kemudian klik **OK**.
2. Lengkapi kode programnya seperti berikut:

```
Module Module1
Sub Main()
' Mencetak teks ke console
Console.WriteLine("Apa Kabar Teman-Teman ")
Console.WriteLine("Selamat Belajar Pemrograman Visual")
Console.Read()
End Sub
End Module
```

3. Tekan **F5** untuk menjalankan aplikasi.
4. Simpan aplikasi Anda.

## BAB II FORM DAN OBJEK KONTROL

### 3.1. Tujuan Pembelajaran

- Memahami dan mengenali bentuk dan objek kontrol.
- Tahu cara membuat aplikasi sederhana dengan objek bentuk dan kontrol;

### 3.2. Teori Dasar Form dan Objek Kontrol

Form merupakan alat interaktif antara pengguna dengan aplikasi yang dibuat. Bentuk dapat digunakan sebagai wadah atau wadah untuk objek kontrol. Bentuk juga dapat dipandang sebagai objek karena dapat bereaksi terhadap suatu peristiwa. Bentuk dapat diklasifikasikan menjadi dua jenis, yaitu:

#### 1. Form Dinamis

Ini adalah bentuk yang dapat dimanipulasi atau dibentuk kembali dan menyisipkan objek kontrol yang berisi perintah yang diperlukan oleh aplikasi untuk membuat, misalnya:

- Window (Windows Form, Console, Class Library, WPF dsb.)
- Web (ASP.Net Web Application, ASP.NET Web Servis dsb)
- Smart Device (Smart Device Project)
- Database(Access dan SQL Server)
- Report (Report Application dan Crystal Report)
- WCF (Windows Communication Foundation)
- dll

#### 2. Form Statis

Ini adalah bentuk form yang tidak dapat dimanipulasi atau diubah, dan objek kontrol dimasukkan. Formulir ini hanya dapat dipanggil dengan skrip, Misalnya:

##### a. Form Pesan (MessageBox)

Formulir bertanggung jawab untuk menampilkan deskripsi pesan dari suatu peristiwa yang diterima oleh aplikasi

##### b. InputBox (Kotak Input Pesan)

Formulir ini digunakan untuk interaksi antara pengguna dan aplikasi yang dihasilkan, di mana pengguna memasukkan nilai, kemudian mengklik tombol dan menunggu efek yang ditimbulkan oleh aplikasi yang dihasilkan.

## OBJEK KONTROL

### Label, TextBox dan Button

**Label** merupakan objek control yang dapat menampilkan output tetapi tidak dapat memberikan input pada saat dijalankan.

**TextBox** Merupakan objek control yang dapat diberikan input pada saat program dijalankan.

**Button** merupakan objek control yang dapat mengeksekusi perintah-perintah yang telah diberikan.

### CheckBox dan RadioButton

**CheckBox** merupakan objek control yang berfungsi untuk memilih beberapa item data. Dengan objek ini anda dapat lebih dari satu pilihan dan bahkan memilih semua pilihan tersedia.

**RadioButton** fungsi hampir sama dengan checkbox, tetapi hanya dapat memilih satu pilihan yang tersedia.

**ComboBox dan ListBox**

**ComboBox** merupakan objek control yang dapat digunakan untuk menampilkan daftar item dengan pilihan dropdown.

**ListBox** merupakan objek control yang jauh berbeda dengan ComboBox hanya saja pada listbox menu daftar pilihan ditampilkan secara keseluruhan.

**Timer dan DateTimePicker**

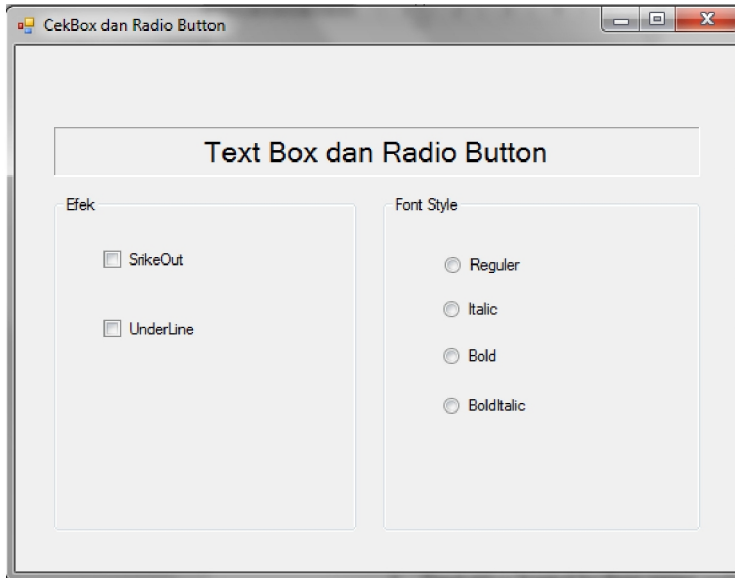
**Timer** merupakan objek control yang berfungsi untuk mengatur dan menampilkan waktu yang diperlukan.

**DateTimePicker** objek control yang berfungsi untuk menampilkan tanggal dengan menu DropDown.

**3.3. Praktikum Form dan Objek Kontrol**

**CheckBox dan RadioButton**

1. Buat **project baru**, dengan nama **Prak2**.
2. Desain Tampilan form seperti gambar berikut ini :



3. Tambahkan kontrol ke form seperti pada gambar diatas dan atur propertinya seperti tabel berikut:

Object	Properties	Nilai
Form1	Name Text	Form1 CheckBox dan RadioButton
Label1	Name Autosize BorderStyle Font TextAlign Text	Lbl_Font False Fixed3D Microsoft Sans Serif, 16 Center CheckBox dan RadioButton

GroupBox1	Text	Efek
GroupBox2	Text	Font Style
CheckBox1	Text	StrikeOut
CheckBox2	Text	Underline
RadioButton1	Text	Regular
RadioButton2	Text	Italic
RadioButton3	Text	Bold
RadioButton4	Text	BoldItalic

4. Klik ganda pada CheckBox1, kemudian ketikkan kode program berikut ini :

```

Private Sub CheckBox1_CheckedChanged(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
CheckBox1.CheckedChanged

    Lbl_Font.Font = New Font("Microsoft sans Serif",
Lbl_Font.Font.Size, _
    Lbl_Font.Font.Style Xor FontStyle.Strikeout)

End Sub

```

5. Klik ganda pada CheckBox2, kemudian ketikkan kode program berikut ini :

```

Private Sub CheckBox2_CheckedChanged(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
CheckBox2.CheckedChanged

    Lbl_Font.Font = New Font("Microsoft sans Serif",
Lbl_Font.Font.Size, _
    Lbl_Font.Font.Style Xor FontStyle.Underline)

End Sub

```

6. Klik ganda pada RadioButton1, kemudian ketikkan kode program berikut ini :

```
Private Sub RadioButton1_CheckedChanged(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
RadioButton1.CheckedChanged

    Lbl_Font.Font = New Font("Microsoft sans Serif",
Lbl_Font.Font.Size, _
    Lbl_Font.Font.Style Xor FontStyle.Regular)

End Sub
```

7. Klik ganda pada RadioButton2, kemudian ketikkan kode program berikut ini :

```
Private Sub RadioButton2_CheckedChanged(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
RadioButton2.CheckedChanged

    Lbl_Font.Font = New Font("Microsoft sans Serif",
Lbl_Font.Font.Size, _
    Lbl_Font.Font.Style Xor FontStyle.Italic)

End Sub
```

8. Klik ganda pada RadioButton3, kemudian ketikkan kode program berikut ini :

```
Private Sub RadioButton3_CheckedChanged(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
RadioButton3.CheckedChanged

    Lbl_Font.Font = New Font("Microsoft sans Serif", _
Lbl_Font.Font.Size, Lbl_Font.Font.Style Xor FontStyle.Bold)

End Sub
```

9. Klik ganda pada RadioButton4, kemudian ketikkan kode program berikut ini :

```
Private Sub RadioButton4_CheckedChanged(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
RadioButton4.CheckedChanged

    Lbl_Font.Font = New Font("Microsoft sans Serif", _
    Lbl_Font.Font.Size, Lbl_Font.Font.Style Xor FontStyle.Bold Xor
FontStyle.Italic)

End Sub
```

10. Jalankan aplikasi dengan menekan tombol **F5** (di keyboard), atau melalui ikon Start Debugging di toolbar, atau melalui menu **Debug > Start Debugging**.
11. Simpan aplikasi **Anda**.

## BAB III FORM DAN OBJEK KONTROL LANJUTAN

### 3.1. Tujuan Pembelajaran

- Memahami dan mengenal Form dan Objek Kontrol;
- Dapat membuat aplikasi sederhana dengan menggunakan form dan objek control.

### 3.2. Dasar Teori

#### OBJEK KONTROL

#### MenuStrip, ContextMenuStrip dan RichTextBox

**MenuStrip** adalah merupakan objek control yang berfungsi untuk membuat menu pilihan dropdown.

**ContextMenuStrip** adalah merupakan objek control yang berfungsi untuk membuat menu pilihan yang akan tampil pada saat tombol kanan mouse di klik.

**RichTextBox** adalah merupakan objek control yang fungsinya hampir sama seperti textbox hanya saja di objek control ini lebih banyak menampung karakter dan bisa manipulasi karakter seperti menggaris bawah sbd.

#### OpenFileDialog dan SaveFileDialog

**OpenFileDialog** adalah merupakan objek control yang berfungsi untuk membuka kotak dialog Open.

**SaveFileDialog** berfungsi untuk membuka kotak dialog save yang dimiliki windows.

#### ToolStrip dan StatusStrip

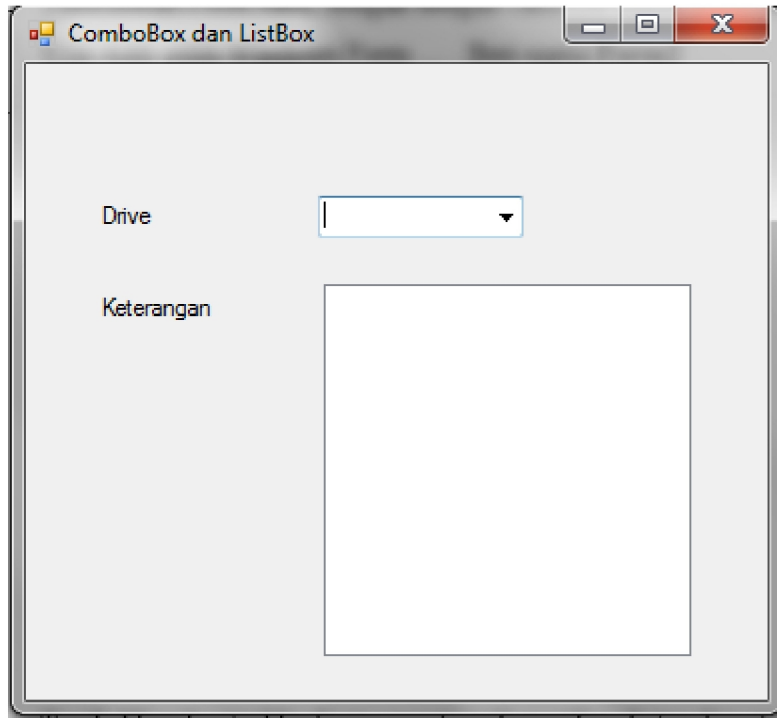
**ToolStrip** adalah merupakan objek control yang berfungsi hampir sama dengan tombol pada umumnya hanya saja pada objek control ini setiap tombol memiliki gambar dan digunakan untuk mengakses gambar dengan cepat.

**StatusStrip** digunakan untuk menampilkan informasi seperti nama aplikasi, jam, tanggal dll. Pada umumnya objek control ini terletak di bagian bawah aplikasi.

### 3.3. Praktikum

#### ComboBox dan ListBox

1. Tambahkan Form baru, dengan dengan cara klik kanan pada **Prak2**. Klik Add, pilih Windows Form ... . Beri nama **Form2**
2. Buatlah desain Tampilan form seperti gambar berikut ini :



3. Tambahkan kontrol ke form seperti pada gambar diatas dan atur propertinya seperti tabel berikut:

Object	Properties	Nilai
Form2	Name Text Start Position	Form2 ComboBox dan ListBox CenterScreen
Label1	Text	Drive
Label2	Text	Keterangan
ComboBox1	Name	Cmb_Drive
ListBox1	Name	LstInfo

4. Klik icon view code pada solution Explorer, di atas Public Class Form1, ketikkan kode program berikut ini :

```
Imports System.IO
Public Class Form2
```

5. Klik ganda pada form, kemudian ketikkan kode program berikut ini :

```
Private Sub Form2_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
    Cmb_Drive.Items.AddRange(IO.DriveInfo.GetDrives)
End Sub
```

6. Klik ganda pada ComboBox, Kemudian ketik kode program berikut ini :

```

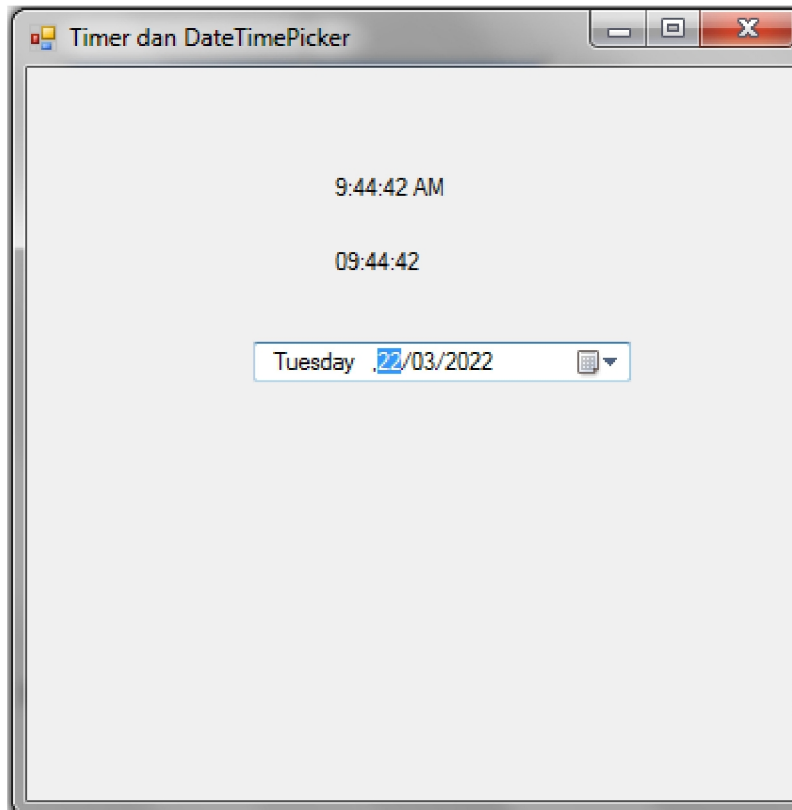
Private Sub Cmb_Drive_SelectedIndexChanged(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
Cmb_Drive.SelectedIndexChanged
    LstInfo.Items.Clear()
    LstInfo.Items.Add("Nama Drive :" & Dir(" " & _
Cmb_Drive.Text & " ", FileAttribute.Volume) & " ")
    LstInfo.Items.Add("System :" & Dir(" " & _
Cmb_Drive.Text & " ", FileAttribute.System) & " ")
    Dim dvr As New DriveInfo(Cmb_Drive.Text)
    Dim kapasitas = CStr(dvr.TotalSize / 1000000) & " MB"
    Dim sisa = CStr(dvr.TotalFreeSpace / 1000000) & " MB"
    Dim Terpakai = CStr(Val(kapasitas) - Val(sisa)) & " MB"
    LstInfo.Items.Add("Kapasitas : " & kapasitas)
    LstInfo.Items.Add("Terpakai : " & Terpakai)
    LstInfo.Items.Add("Sisa : " & sisa)
End Sub

```

7. Jalankan aplikasi dengan menekan tombol **F5** (di keyboard), atau melalui ikon Start Debugging di toolbar, atau melalui menu **Debug > Start Debugging**.
8. Simpan aplikasi **Anda**

#### Timer dan DateTimePicker

1. Tambahkan Form baru, dengan dengan cara klik kanan pada **Prak2**. Klik Add, pilih Windows Form ... . Beri nama **Form3**
2. Buatlah desain Tampilan form seperti gambar berikut ini :



3. Tambahkan kontrol ke form seperti pada gambar diatas dan atur propertinya seperti tabel berikut:

Object	Properties	Nilai
Form3	Name Text Start Position	Form3 Timer dan DateTimePicker CenterScreen
Label1	name	Lbl12jam
Label2	Name	Lbl24jam
DateTimePicker1	Name	Dtanggal
Timer1	Enable Interval	True 1000

4. Klik ganda pada form, ketikkan kode program berikut ini :

```

Private Sub Form3_Load(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles MyBase.Load
    DTtanggal.Format = DateTimePickerFormat.Custom
    DTtanggal.CustomFormat = "dddd,dd/MM/yyyy"
    DTtanggal.Value = Format(Now)
End Sub

```

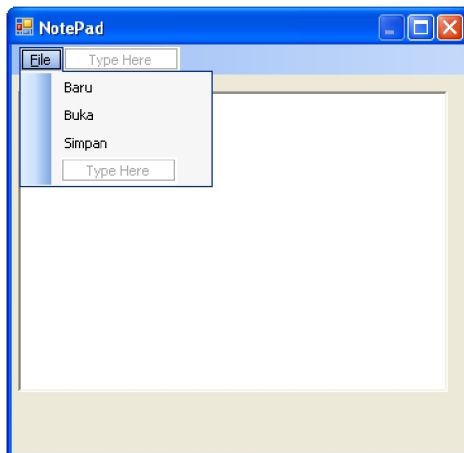
- Klik ganda pada Timer1, ketikkan kode program berikut ini :

```
Private Sub Timer1_Tick(ByVal sender As System.Object, ByVal e As System.EventArgs)
    Handles Timer1.Tick
        Lbl12Jam.Text = Format(Now, "h:mm:ss tt")
        Lbl24Jam.Text = Format(Now, "hh:mm:ss")
End Sub
```

- Jalankan aplikasi dengan menekan tombol **F5** (di keyboard), atau melalui ikon Start Debugging di toolbar, atau melalui menu **Debug > Start Debugging**.
- Simpan aplikasi **Anda**

### MenuStrip, ContextMenuStrip dan RichTextBox

- Tambahkan Form baru, dengan dengan cara klik kanan pada **Prak2**. Klik Add, pilih Windows Form ... . Beri nama **Form4**
- Buatlah desain Tampilan form seperti gambar berikut ini :



- Tambahkan menuStrip seperti pada form di atas
- Tambahkan kontrol ke form seperti pada gambar diatas dan atur propertinya seperti tabel berikut:

Object	Properties	Nilai
Form1	Name Text Start Position	Form1 NotePad CenterScreen
MenuStrip1	name	MenuStrip1
RichTextBox1	Name ContextMenuStrip	RchTeks ContextMenuStrip1
ContextMenuStrip1	Name	ContextMenuStrip1

5. Klik ganda pada menu buka, ketikkan kode program berikut ini :

```
Private Sub BukaToolStripMenuItem_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
BukaToolStripMenuItem.Click
    Dim kotak_dialog As OpenFileDialog = New OpenFileDialog
    kotak_dialog.Filter = "File Text (*.txt) | *.txt"
    kotak_dialog.ShowDialog()
    Dim fileText As String
    =My.Computer.FileSystem.ReadAllText(kotak_dialog.FileName)
    RchTeks.Text = fileText
End Sub
```

6. Klik ganda pada menu simpan, ketikkan kode program berikut ini :

```
Private Sub SimpanToolStripMenuItem_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
SimpanToolStripMenuItem.Click
    Dim kotak_dialog As SaveFileDialog = New SaveFileDialog
    kotak_dialog.Filter = "File Text (*.txt) | *.txt"
    kotak_dialog.ShowDialog()
    My.Computer.FileSystem.WriteAllText("" &
kotak_dialog.FileName & "",RchTeks.Text, False)
End Sub
```

7. Klik ganda pada menu Baru, ketikkan kode program berikut ini :

```
Private Sub BaruToolStripMenuItem_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
BaruToolStripMenuItem.Click
    RchTeks.Text = ""
End Sub
```

8. Klik ganda pada menu ContextMenuStrip dengan menu regular, kemudian ketik kode program berikut ini :

```
Private Sub RegularToolStripMenuItem_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
RegularToolStripMenuItem.Click
    RchTeks.SelectionFont = New Font(RchTeks.SelectionFont,
FontStyle.Regular)
End Sub
```

9. Klik ganda pada menu ContextMenuStrip dengan menu Bold, kemudian ketik kode program berikut ini :

```
Private Sub BoldToolStripMenuItem_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
BoldToolStripMenuItem.Click
    RchTeks.SelectionFont = New Font(RchTeks.SelectionFont,
FontStyle.Bold)
End Sub
```

10. Jalankan aplikasi dengan menekan tombol **F5** (di keyboard), atau melalui ikon Start Debugging di toolbar, atau melalui menu **Debug > Start Debugging**.
11. Simpan aplikasi **Anda**

**BAB IV**  
**VARIABEL, KONSTANTA, TIPE DATA DAN OPERATOR**

**4.1 Tujuan Pembelajaran**

- Memahami dan mengenal Tipe Data, Variabel, Konstanta dan Operator
- Dapat membuat aplikasi sederhana dengan menggunakan Tipe Data, Variabel, Konstanta dan Operator.

**4.2 Teori Tipe Data, Variabel, Konstanta Dan Operator**

**Tipe Data**

Tipe data merupakan suatu bentuk penggolongan jenis data berdasarkan kategori data, ukuran dan kegunaan data yang dapat ditampung oleh sebuah variabel. Tipe data pada visual basic 2008 dapat dilihat pada table berikut ini :

<b>Tipe Data</b>	<b>.Net Tipe</b>	<b>Ukuran</b>	<b>Jangkauan</b>
Decimal	System.Decimal	12 Bytes	0 – 28 bilangan decimal
Byte	System.Byte	1 Bytes	0 – 255
Double	System.Double	8 Bytes	Bilangan negative antara : -1.79769313486232e+308 sampai -4.94065645841247E-324 Bilangan positif antara : 4.94065645841247E-324 sampai 1.79769313486232e+308
Long	System.Int64	4 Bytes	-9.223.372.036.854.775.808 sampai 9.223.372.036.854.775.808
Single	System.Single	4 Bytes	Bilangan negative antara : -3.402823E sampai -1.401298E Bilangan positif antara : 1.401289E-45 sampai 3.402823e
Integer	System.Int32	4 Bytes	-2.147.483.648 sampai - 2.147.483.647
Boolean	System.Boolean	2 Bytes	True or False
Date	System.DateTime	2 Bytes	1 Januari 0001 sampai 31 Desember 9999
Short	System.Int16	2 Bytes	-32.768 – 32.767
Char	System.Char	2 Bytes	0 to 65535
String	System.String	10 Bytes + (2 x jumlah karakter	0 sampai kira-kira 2 milyar karakter Unicode
Object	System.Object	4 Bytes	Semua Tipe, jenis

**Konversi Tipe Data**

Konversi Tipe data adalah sebuah fungsi yang digunakan untuk mengkonversi ( mengubah tipe dari satu tipe ke tipe data yang lain). Konversi tipe data dapat dilihat pada table berikut ini :

<b>Konversi</b>	<b>Tipe Data</b>	<b>Keterangan</b>
-----------------	------------------	-------------------

CInt	Integer	Integer
cBool	Boolean	Nilai 0 false, selain 0 true
cByte	Byte	Pembulatan nilai decimal
cLng	Long	Pembulatan nilai decimal
cCur	Currency	Money
cShort	Short	Pembulatan nilai decimal
cSng	Single	Pembulatan nilai decimal
cDate	Date	DateTime
CStr	String	Karakter Unicode
cDbl	Double	
cDec	Decimal	Bilangan decimal
cObj	Object	Object

### Variabel

Variabel adalah merupakan tempat penyimpanan data dalam memori computer yang mengandung data atau nilai sementara dari sebuah proses pada suatu pemrograman.

Deklarasi Variabel

Deklarasi Variabel adalah merupakan fungsi untuk mengenal suatu variabel berikut dengan tipe data yang pakai pada variabel tersebut. Pendeklarasian variabel di bagi menjadi dua kategori antara lain :

#### 1. Deklarasi Ekplisit

Suatu pendeklarasian umum yang sangat sering dilakukan dengan memberikan kata kunci as.

Bentuk umum :

{Private | Public | Dim | ...} Nama\_Var [{As Tipe data}]

Contoh :

Dim Nim as Integer

Dim Nama as String

#### 2. Deklarasi Emplisit

Deklarasi Emplisit adalah suatu pendeklarasian yang tidak memberikan kata kunci As pada semua variabel tersebut. Melainkan memberikan satu karakter atau symbol di akhir variabel. Tabel symbol deklarasi emplisit.

Karakter / Simbol	Tipe Data
%	Integer
\$	String
&	Long
@	Currency
!	Single
	Double

Contoh :

Dim NIM% 'mewakili tipe data integer

Dim Nama\$ 'mewakili tipe data string

### **Ruang Lingkup Variabel**

Ruang lingkup variabel sangat mempengaruhi sampai dimana nama variabel tersebut dapat dikenali oleh aplikasi. Ruang Lingkup Variabel dapat digolongkan menjadi :

1. Local Variabel

Local variabel merupakan suatu pendeklarasian variabel di dalam ruang lingkup objek saja. Tidak dapat digunakan di dalam objek lain.

2. Global Variabel

Global variabel adalah merupakan suatu pendeklarasian variabel tidak hanya di dalam ruang lingkup objek saja. Tetapi mencakup semua presedure dan objek.

### **Konstanta**

Konstanta merupakan suatu variabel yang mempunyai nilai tetap dan tidak dapat diubah nilainya.

#### **Deklarasi Konstanta**

Deklarasi konstanta :

```
{Const} Nama_Var [{AS tipedata}]=<nilai>
```

Contoh :

```
Const nama_bidang as string="persegi_panjang"  
Public const Panjang As Integer=8  
Private const Lebar As Integer=5  
Private const luas As Integer=Panjang*Lebar  
Private const Keliling As Integer=2*(Panjang+Lebar)
```

#### **Deklarasi Konstanta Enumerasi**

Deklarasi Konstanta Enumerasi merupakan pendeklarasian daftar konstanta yang bersipat global . Dengan deklarasi konstanta enumerasi dapat membuat daftar konstanta sendiri dapam bentuk seperti modul kelas dan dapat diakses oleh projek lain.

Format penulisannya :

```
{Private | Public | ... }[enum] Nama_Variabel_konstanta
```

Contoh :

```
Const nama_bidang as string="persegi_panjang"  
Private const Panjang As Integer=8  
Private const Lebar As Integer=5  
Private const luas As Integer=Panjang*Lebar  
Private const Keliling As Integer=2*(Panjang+Lebar)  
Public enum Hitung  
    Luas = Luas  
    Keliling=Keliling  
end Enum  
A=Hitung.Luas 'menggunakan variabel enum luas  
B=Hitung.Keliling ' menggunakan variabel enum keliling
```

### **Operator**

Operator Aritmatika

Operator Aritmatika adalah merupakan suatu operator perhitungan matematika yang digunakan untuk proses perhitungan matematika. Operator aritmatika dapat dilihat pada table berikut ini :

Operator	Fungsi
+	Penjumlahan
-	Pengurangan
*	Perkalian
/	Pembagian
Mod	Sisa hasil pembagian
^	Perpangkatan

### Operator Perbandingan

Operator perbandingan adalah merupakan suatu operator yang menyatakan suatu kondisi itu benar atau salah. Operator perbandingan dapat dilihat pada table berikut ini :

Operator	Fungsi
<>	Tidak sama dengan
<	Lebih kecil
>	Lebih besar
>=	Lebih besar sama dengan
<=	Lebih kecil sama dengan
Like	Ciri yang sama
Is	keterangan

### Operator Logika

Operator Logika adalah merupakan suatu operator yang dapat membandingkan dua ekspresi atau lebih data logika (Boolean) dan menghasilkan data logika (Boolean) baru. Operator logika dapat dilihat pada table berikut ini:

Operator	Fungsi
NOT	Tidak
AND	Dan
OR	Atau
XOR	Ekklusif atau

### Operator Penugasan

Operator penugasan adalah merupakan operator yang bertugas untuk menentukan nilai yang dikeluarkan oleh suatu variabel terhadap kondisi tertentu.

Operator	Fungsi
=	Sama dengan

### Operator Penggabungan (Concatenation)

Operator Penggabungan adalah merupakan suatu operator yang dapat digunakan untuk menggabungkan dua buah string atau lebih. Operator Penggabungan dapat dilihat pada table berikut ini :

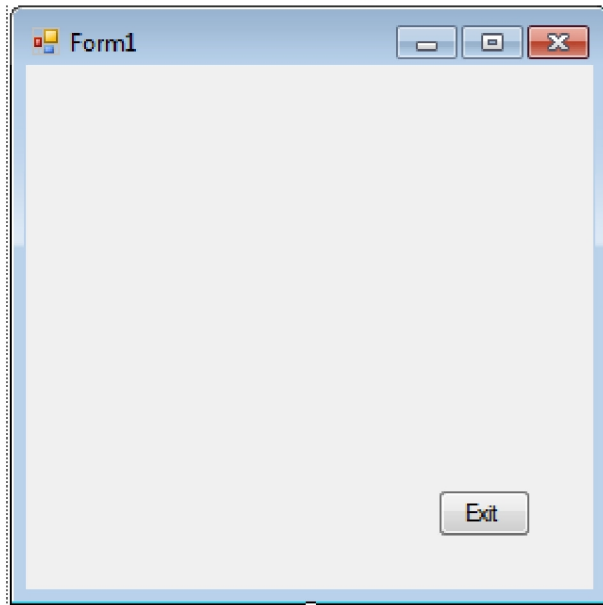
Operator	Fungsi
----------	--------

+	Plus
&	Dan

### 4.3 Praktikum

#### Variabel Lokal

1. Buat **project baru**, dengan nama **Prak3**.
2. Desain Tampilan form seperti gambar berikut ini :



3. Tambahkan kontrol ke form seperti pada gambar diatas dan atur propertinya seperti tabel berikut:

Object	Properties	Nilai
Form1	Name Text	Form1 Variabel Lokal
Button1	Name Text	BtnExit E&xit

4. Klik ganda pada BtnExit1, kemudian ketikkan kode program berikut ini :

```

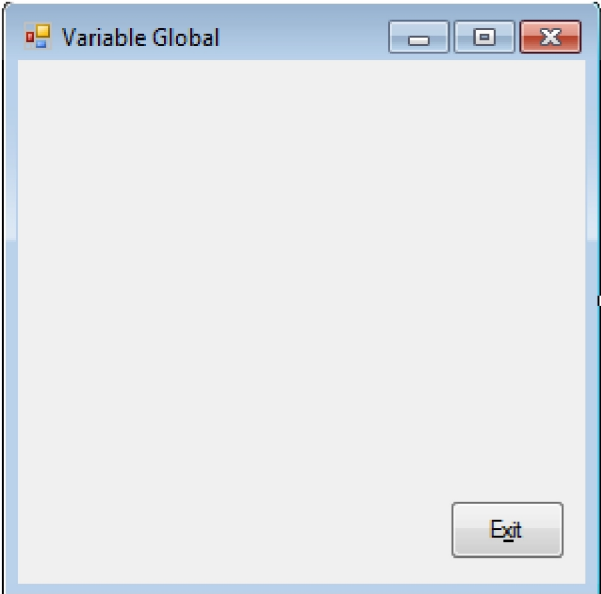
Dim tutup As String
tutup = MessageBox.Show("Anda yakin tutup form ini ?", "Lokal
Variabel", MessageBoxButtons.YesNo, MessageBoxIcon.Question)
    If tutup = MsgBoxResult.Yes Then
        End
    Else
    End If

```

5. Jalankan aplikasi dengan menekan tombol **F5** (di keyboard), atau melalui ikon Start Debugging di toolbar, atau melalui menu **Debug > Start Debugging**.
6. Simpan aplikasi **Anda**.

**Variabel Global**

1. Tambahkan form baru , dengan nama **Form2**.
2. Desain Tampilan form seperti gambar berikut ini :



3. Tambahkan kontrol ke form seperti pada gambar diatas dan atur propertinya seperti tabel berikut:

Object	Properties	Nilai
Form2	Name Text	Form2 Variabel Global
Button1	Name Text	BtnExit E&xit

4. Klik ganda pada Form2, kemudian ketikkan kode program berikut ini :

```
Public Class Form2
    Dim tutup As String
```

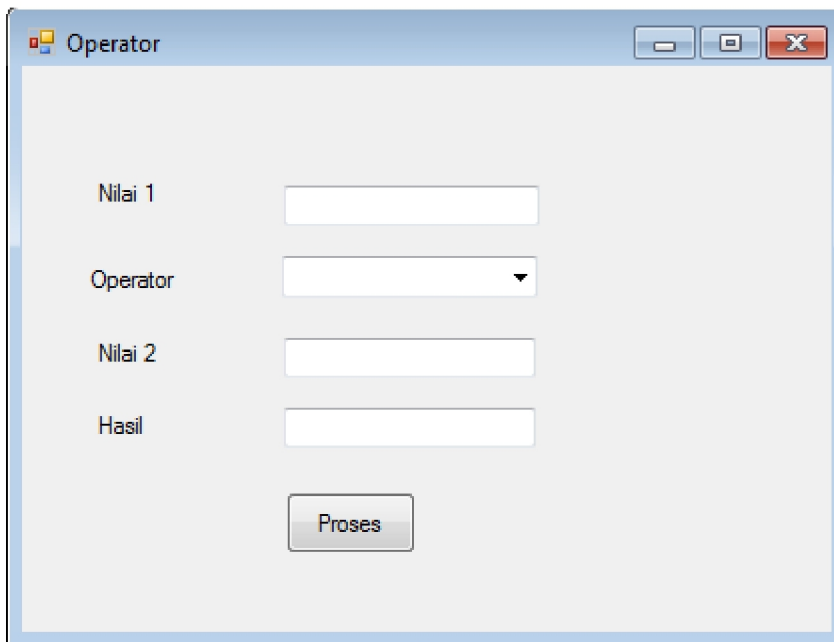
5. Klik ganda pada **BtnExit**, kemudian ketikkan kode program berikut ini :

```
tutup = MessageBox.Show("Anda yakin tutup form ini ?", "Global
Variabel", MessageBoxButtons.YesNo, MessageBoxIcon.Question)
    If tutup = MsgBoxResult.Yes Then
        End
    Else
    End If
```

6. Jalankan aplikasi dengan menekan tombol **F5** (di keyboard), atau melalui ikon Start Debugging di toolbar, atau melalui menu **Debug > Start Debugging**.
7. Simpan aplikasi **Anda**.

### Operator

1. Tambahkan **Form baru**, dengan nama **Form3**.
2. Desain Tampilan form seperti gambar berikut ini :



3. Tambahkan kontrol ke form seperti pada gambar diatas dan atur propertinya seperti tabel berikut:

Object	Properties	Nilai
--------	------------	-------

Form3	Name Text	Form3 Operator
Label1	Text	Nilai1
Label2	Text	Operator
Label3	Text	Nilai2
Label4	Text	Hasil
TextBox1	Name	TxtNilai1
ComboBox1	Name	CmbOperator
TextBox2	Name	TxtNilai2
TextBox3	Name	TxtHasil
Button1	Name Text	BtnProses Proses

4. Klik ganda pada Form3, kemudian ketikkan kode program berikut ini :

```

CmbOperator.Items.Add("+")
CmbOperator.Items.Add("-")
CmbOperator.Items.Add("x")
CmbOperator.Items.Add("/")
CmbOperator.Items.Add("Sisa")
CmbOperator.Items.Add("Pangkat")
CmbOperator.Items.Add("<")
CmbOperator.Items.Add("<")
CmbOperator.Items.Add(">")
CmbOperator.Items.Add("<=")
CmbOperator.Items.Add(">=")
CmbOperator.Items.Add("Like")
CmbOperator.Items.Add("Not")
CmbOperator.Items.Add("And")
CmbOperator.Items.Add("Or")
CmbOperator.Items.Add("Xor")
CmbOperator.Items.Add("+p")
CmbOperator.Items.Add("&")

```

5. Klik ganda pada tombol Proses, lalu ketikkan kode program berikut ini :

```

Select Case CmbOperator.Text
Case "+"
    TxtHasil.Text = Val(TxtNilai1.Text) + Val(TxtNilai2.Text)
Case "-"
    TxtHasil.Text = Val(TxtNilai1.Text) - Val(TxtNilai2.Text)
Case "x"
    TxtHasil.Text = Val(TxtNilai1.Text) * Val(TxtNilai2.Text)
Case "/"
    TxtHasil.Text = Val(TxtNilai1.Text) / Val(TxtNilai2.Text)
Case "Sisa"
    TxtHasil.Text = Val(TxtNilai1.Text) mod
Val(TxtNilai2.Text)
Case "Pangkat"
    TxtHasil.Text = Val(TxtNilai1.Text) ^ Val(TxtNilai2.Text)

Case "<"
    TxtHasil.Text = TxtNilai1.Text < TxtNilai2.Text
Case "<"
    TxtHasil.Text = TxtNilai1.Text < TxtNilai2.Text
Case ">"
    TxtHasil.Text = TxtNilai1.Text > TxtNilai2.Text
Case "<="
    TxtHasil.Text = TxtNilai1.Text <= TxtNilai2.Text
Case ">="
    TxtHasil.Text = TxtNilai1.Text >= TxtNilai2.Text

```

```

TxtHasil.Text = TxtNilai1.Text >= TxtNilai2.Text
Case "Like"
    TxtHasil.Text = TxtNilai1.Text Like TxtNilai2.Text
Case "Not"
    TxtHasil.Text = Not TxtNilai1.Text = TxtNilai2.Text
Case "And"
    TxtHasil.Text = TxtNilai1.Text = "UNISTEKOM" And
TxtNilai2.Text = "UNISTEKOM"
Case "Or"
    TxtHasil.Text = TxtNilai1.Text = "UNISTEKOM" Or
TxtNilai2.Text = "UNISTEKOM"
Case "Xor"
    TxtHasil.Text = TxtNilai1.Text = "UNISTEKOM" Xor
TxtNilai2.Text = "UNISTEKOM"
Case "+P"
    TxtHasil.Text = TxtNilai1.Text + TxtNilai2.Text
Case "&"
    TxtHasil.Text = TxtNilai1.Text & TxtNilai2.Text
End Select

```

6. Jalankan aplikasi dengan menekan tombol **F5** (di keyboard), atau melalui ikon Start Debugging di toolbar, atau melalui menu **Debug > Start Debugging**.
7. Simpan aplikasi **Anda**.

## BAB V PENCABANGAN

### 5.1 Tujuan Pembelajaran

- Memahami dan mengenal Pencabangan (if ... then dan Select Case ....)
- Dapat membuat aplikasi sederhana dengan menggunakan Pencabangan (if ... then dan Select Case ...)

### 5.2 Teori Pencabangan

#### Pencabangan

Pencabangan adalah merupakan perintah yang dapat memberikan pilihan suatu kondisi, program akan menjalankan perintah apabila suatu kondisi memenuhi syarat tertentu. Pencabangan dapat dibedakan menjadi :

#### a. IF ... THEN

If ... then merupakan pencabangan yang mempunyai satu pencabangan atau satu blok perintah. Format penulisannya :

```
If kondisi then
[perintah]
End if
```

#### b. IF ... THEN ... ELSE

Suatu perintah pencabangan bersarang (Nested If) yang merupakan perkembangan dari perintah pencabangan IF.. THEN, yang dapat menjalankan satu blok perintah, yang memiliki dua nilai atau syarat bahkan lebih yang akan diuji untuk menjalankan suatu kondisi tertentu. Bila kondisi pertama benar maka jalankan perintah blok pertama, jika kondisi pertama salah maka jalankan perintah blok kedua dan selanjutnya.

Format Penulisannya :

```
IF (kondisi) then
[perintah]
ELSE
[perintah]
End if
```

Dan

```
If (kondisi2) then
[perintah]
ELSEIF (kondisi2) Then
[perintah]
...
ELSE
[perintah]
End If
```

#### c. Select ... Case ...

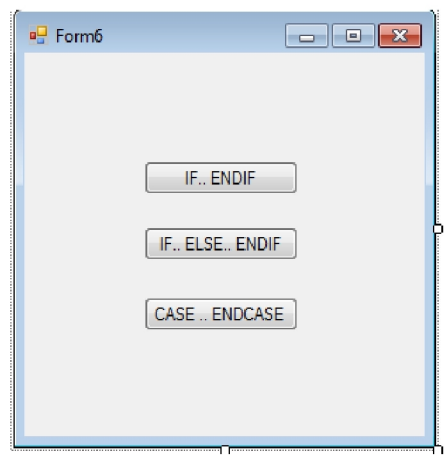
Select Case adalah control percabangan yang mempunyai fungsi hampir sama dengan percabangan if ... then... else. Select mempunyai penulisan dan pembacaan yang lebih mudah, efektif dan efisien. Namun mempunyai kelemahan yaitu tidak dapat menguji lebih dari satu ekspresi atau ungkapan. Format penulisannya adalah sebagai berikut :

```

Select Case kondisi
Case | Case is = ekspresi1
    [perintah1]
Case | Case is = ekspresi2
    [perintah2]
Case | Case is = ekspresi3
    [perintah3]
...
Case Else
    [perintah]
End Select
    
```

**5.3 Prosedur Praktikum**  
**Pencabangan IF.. , IF.. ELSE, CASE..**

1. Buat **form baru**.
2. Desain Tampilan form seperti gambar berikut ini :



3. Tambahkan kontrol ke form seperti pada gambar diatas dan atur propertinya seperti tabel berikut:

Object	Properties	Nilai
Form1	Name Text	Form1 Pencabangan IF
Button1	Name Text	Btn_if IF.. ENDIF

Button2	Name Text	Btn_if_else IF.. ELSE.. ENDIF
Button3	Name Text	Btn_case CASE.. ENDCASE

4. Klik ganda pada Btn\_if, kemudian ketikkan kode program berikut ini :

```

Private Sub Btn_if_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Btn_if.Click
    Dim usia As String, nilai As Single, thn As Integer
    usia = InputBox("Usia Pegawai saat ini?", "Batasan usia pensiun", 0)
    nilai = Val(usia)
    If nilai >= 55 Then
        thn = nilai - 55
        MessageBox.Show("usia pegawai : " & usia & " Tahun", "Konfirmasi",
        MessageBoxButtons.OK, MessageBoxIcon.Information)
        MessageBox.Show("Keterangan : Pegawai harus pensiun " & thn & " tahun yang
        lalu", "Konfirmasi", MessageBoxButtons.OK, MessageBoxIcon.Information)
    End If
End Sub

```

5. Klik ganda pada Btn\_if\_else, kemudian ketikkan kode program berikut ini :

```

Private Sub Btn_if_else_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles Btn_if_else.Click
    Dim usia As String, nilai As Single, thn As Integer
    usia = InputBox("Usia Pegawai saat ini?", "Batasan usia
pensiun", 0)
    nilai = Val(usia)
    If nilai >= 55 Then
        thn = nilai - 55
        MessageBox.Show("usia pegawai : " & usia & " Tahun",
"Konfirmasi", MessageBoxButtons.OK,
MessageBoxIcon.Information)
        MessageBox.Show("Keterangan : Pegawai harus pensiun " &
thn & " tahun yang lalu", "Konfirmasi", MessageBoxButtons.OK,
MessageBoxIcon.Information)
    Else
        thn = 55 - nilai
        MessageBox.Show("usia pegawai : " & usia & " Tahun",
"Konfirmasi", MessageBoxButtons.OK,
MessageBoxIcon.Information)
        MessageBox.Show("Keterangan : Pegawai akan pensiun " &
thn & " lagi", "Konfirmasi", MessageBoxButtons.OK,
MessageBoxIcon.Information)
    End If
End Sub

```

6. Klik ganda pada Btn\_case, kemudian ketikkan kode program berikut ini :

```

Private Sub Btn_case_Click(ByVal sender As System.Object, ByVal
e As System.EventArgs) Handles Btn_case.Click
    Dim isian As String, usia As Integer
    isian = InputBox("Usia anda saat ini?", "Status Anda")
    usia = Val(isian)
    MessageBox.Show("Usia Anda saat ini : " & usia & " tahun",
"Konfirmasi", MessageBoxButtons.OK,
MessageBoxIcon.Information)
    Select Case usia
        Case 1 To 4
            MessageBox.Show("Anda sekarang tergolong balita",
"Konfirmasi", MessageBoxButtons.OK,
MessageBoxIcon.Information)
        Case 5 To 16
            MessageBox.Show("Anda sekarang tergolong anak-anak",
"Konfirmasi", MessageBoxButtons.OK,
MessageBoxIcon.Information)
        Case 17 To 24
            MessageBox.Show("Anda sekarang tergolong remaja",
"Konfirmasi", MessageBoxButtons.OK,
MessageBoxIcon.Information)
        Case 25 To 55
            MessageBox.Show("Anda sekarang tergolong dewasa",
"Konfirmasi", MessageBoxButtons.OK,
MessageBoxIcon.Information)
        Case Is >= 56
    
```

```

        MessageBox.Show("Anda sekarang tergolong manula",
        "Konfirmasi", MessageBoxButtons.OK,
        MessageBoxIcon.Information)
    Case Else
        MessageBox.Show("Anda salah memasukkan usia",
        "Konfirmasi", MessageBoxButtons.OK,
        MessageBoxIcon.Information)
    End Select

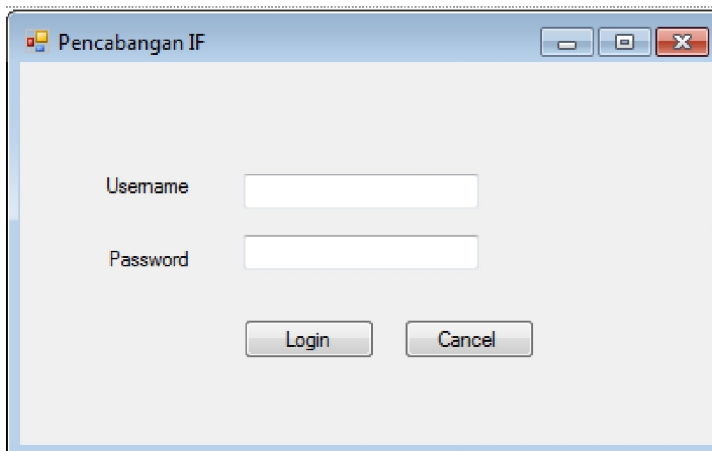
End Sub

```

7. Jalankan aplikasi dengan menekan tombol **F5** (di keyboard), atau melalui ikon Start Debugging di toolbar, atau melalui menu **Debug > Start Debugging**.
8. Simpan aplikasi **Anda**.

**Pencabangan Bersarang (NESTED IF)**

1. Buat **project baru**, dengan nama **Prak4**.
2. Desain Tampilan form seperti gambar berikut ini :



3. Tambahkan kontrol ke form seperti pada gambar diatas dan atur propertinya seperti tabel berikut:

Object	Properties	Nilai
Form1	Name Text	Form1 Pencabangan IF
Label1	Text	User Name
Label2	Text	Password
TextBox1	Name	TxtUserName

TextBox2	Name PasswordChar	TxtPassword *
Button1	Name Text	BtnLogin Login
Button2	Name Text	BtnCancel Cancel

4. Klik ganda pada BtnLogin, kemudian ketikkan kode program berikut ini :

```

If TxtPassword.Text = "" And TxtUserName.Text = "" Then
    MessageBox.Show("User name dan Password harus diisi",
    "Konfirmasi", MessageBoxButtons.OK, MessageBoxIcon.Question)
    TxtUserName.Focus()
ElseIf TxtPassword.Text = "" Then
    MessageBox.Show("Password harus diisi", "Konfirmasi",
    MessageBoxButtons.OK, MessageBoxIcon.Question)
    TxtPassword.Focus()
ElseIf TxtUserName.Text = "" Then
    MessageBox.Show("User name harus diisi", "Konfirmasi",
    MessageBoxButtons.OK, MessageBoxIcon.Question)
    TxtUserName.Focus()
ElseIf TxtUserName.Text = "STIKOM" And TxtPassword.Text
= "1234" Then
    MessageBox.Show("Password dan User Name Benar",
    "Konfirmasi", MessageBoxButtons.OK, MessageBoxIcon.Question)
Else
    MessageBox.Show("Password dan User Name Anda Salah,
    ulangi lagi", "Konfirmasi", MessageBoxButtons.OK,
    MessageBoxIcon.Question)
    TxtUserName.Text = ""
    TxtPassword.Text = ""
    TxtUserName.Focus()
End If
End Sub

```

5. Klik ganda pada BtnCancel, kemudian ketikkan kode program berikut ini :

```
End
```

6. Jalankan aplikasi dengan menekan tombol **F5** (di keyboard), atau melalui ikon Start
7. Debugging di toolbar, atau melalui menu **Debug > Start Debugging**.
8. Simpan aplikasi **Anda**.

## BAB VI PERULANGAN

### 6.1 Tujuan Pembelajaran

- Memahami dan mengenal Perulangan (For ... Next, While, Do While)
- Dapat membuat aplikasi sederhana dengan menggunakan Perulangan (For ... Next, While, Do While)

### 6.2 Teori Perulangan

#### PERULANGAN

Perintah perulangan adalah merupakan control perulangan yang berfungsi untuk melakukan perulangan (iterasi) dari suatu blok program secara berulang-ulang.

#### For ... Next

For ... Next merupakan suatu control perulangan yang mempunyai nilai tetap, jumlah perulangan telah diketahui sebelumnya. Format penulisannya adalah sebagai berikut :

```
For Counter=niali_awal to nilai_akhir  
  [ Perintah ]
```

#### Next (Counter)

#### For Each ... In ... Next

Merupakan suatu control yang digunakan untuk perulangan yang jumlah pengulangannya sesuai dengan jumlah elemen dari suatu koleksi objek dan biasanya tidak diketahui dengan pasti berapa jumlahnya. Koleksi merupakan kumpulan data atau objek yang didefinisikan dengan kode program.

Format Penulisannya :

```
For Each elemen In Group
```

```
  [ Perintah ]
```

```
  [ Perintah ]
```

```
Next [elemen]
```

#### While ...

**While** merupakan suatu control perulangan yang berfungsi untuk melakukan perulangan yang memiliki satu syarat tertentu. Dan akan terus dijalankan selama syarat tersebut terpenuhi dan begitupun sebaliknya jika syarat tidak terpenuhi maka pernyataan tidak akan dijalankan.

Format Penulisannya :

```
While (kondisi)
```

```
  [perintah1]
```

```
  [perintah2]
```

```
  ...
```

```
End While
```

#### DO While ... LOOP

Merupakan suatu control perulangan yang melakukan perulangan terlebih dahulu dan bentuk pengujian dilakukan belakangan atau sebaliknya.

Format penulisannya :

```

DO While (kondisi)
    [perintah1]
    [Exit Do]
    [perintah2]
LOOP

```

Bentuk 2

```

DO
    [perintah1]
    [Exit Do]
    [perintah2]
LOOP While (kondisi)

```

**DO UNTIL**

Jika Do While melakukan perulangan selama kondisi memenuhi until akan melakukan perulangan hingga kondisi memenuhi. Selama kondisi tidak memenuhi, loop akan dijalankan terus.

Format penulisannya :

```

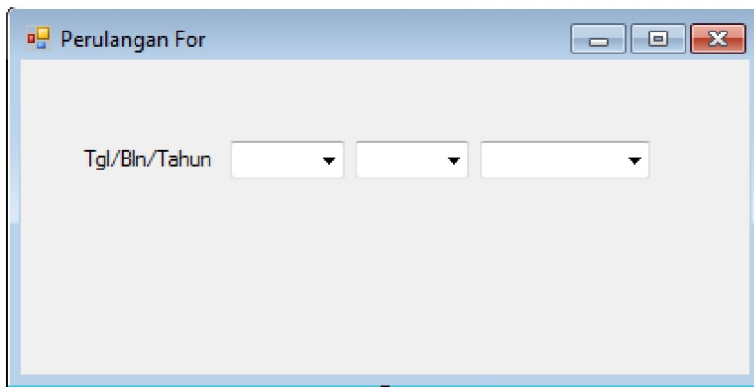
DO
    [perintah1]
LOOP UNTIL

```

**6.3 Praktikum**

**Perulangan FOR**

1. Buat project baru dengan nama **Prak8**.
2. Desain Tampilan form seperti gambar berikut ini :



3. Tambahkan kontrol ke form seperti pada gambar diatas dan atur propertinya seperti tabel berikut:

Object	Properties	Nilai
Form1	Name Text	Form1 Perulangan FOR
Label1	Text	Tgl/Bln/Thn
ComboBox1	Name	CmbTgl
ComboBox2	Name	CmbBln

ComboBox3	Name	CmbThn
-----------	------	--------

- Klik ganda pada Form1, kemudian ketikkan kode program berikut ini :

```

Dim tgl, bln, thn As Integer

For tgl = 1 To 31
    cmbTgl.Items.Add(tgl)
Next tgl

For bln = 1 To 12
    cmbBln.Items.Add(bln)
Next bln

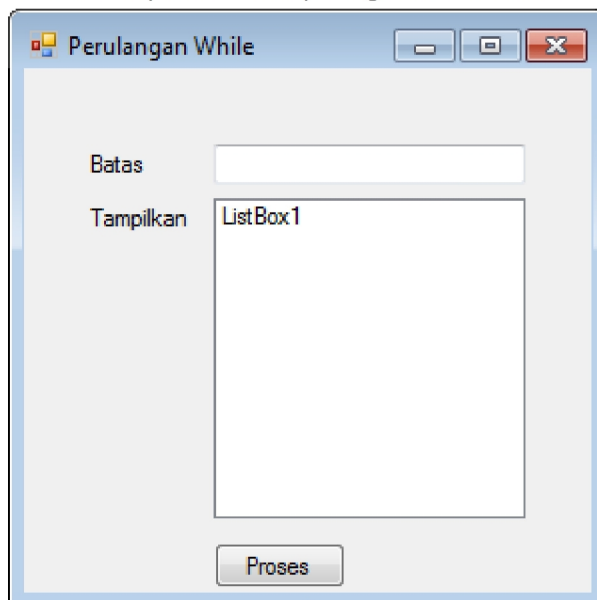
For thn = 1900 To 9999
    CmbThn.Items.Add(thn)
Next thn

```

- Jalankan aplikasi dengan menekan tombol **F5** (di keyboard), atau melalui ikon Start Debugging di toolbar, atau melalui menu **Debug > Start Debugging**.
- Simpan aplikasi **Anda**.

### Perulangan WHILE

- Tambahkan **Form baru**, dengan nama **Form3**.
- Desain Tampilan form seperti gambar berikut ini :



- Tambahkan kontrol ke form seperti pada gambar diatas dan atur propertinya seperti tabel berikut:

Object	Properties	Nilai
Form1	Name Text	Form1 Perulangan WHILE
Label1	Text	Batas
Label2	Text	Tampilkan
TextBox1	Name	TxtBatas
ListBox1	Name	ListBox1
Button1	Name Text	Button1 Proses

4. Klik ganda pada Proses, kemudian ketikkan kode program berikut ini :

```

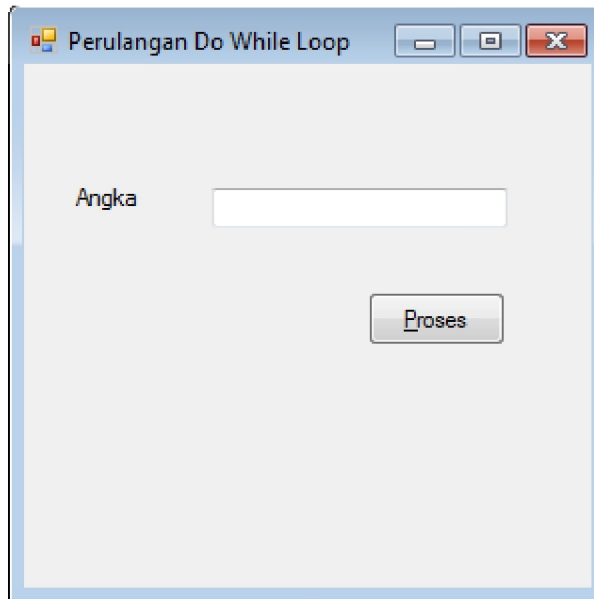
Dim angka As Integer
ListBox1.Items.Clear()
angka = 1
While angka <= Val(TxtBatas.Text)
    ListBox1.Items.Add(angka)
    angka = angka + 1
End While

```

5. Jalankan aplikasi dengan menekan tombol **F5** (di keyboard), atau melalui ikon Start Debugging di toolbar, atau melalui menu **Debug > Start Debugging**.
6. Simpan aplikasi **Anda**.

#### Perulangan DO WHILE...LOOP

1. Tambahkan **Form baru**, dengan nama **Form4**.
2. Desain Tampilan form seperti gambar berikut ini :



3. Tambahkan kontrol ke form seperti pada gambar diatas dan atur propertinya seperti tabel berikut:

Object	Properties	Nilai
Form1	Name Text	Form1 Perulangan DO WHILE ... LOOP
Label1	Text	Angka
TextBox1	Name	TxtAngka
Button1	Name Text	Button1 Proses

4. Klik ganda pada Proses, kemudian ketikkan kode program berikut ini :

```

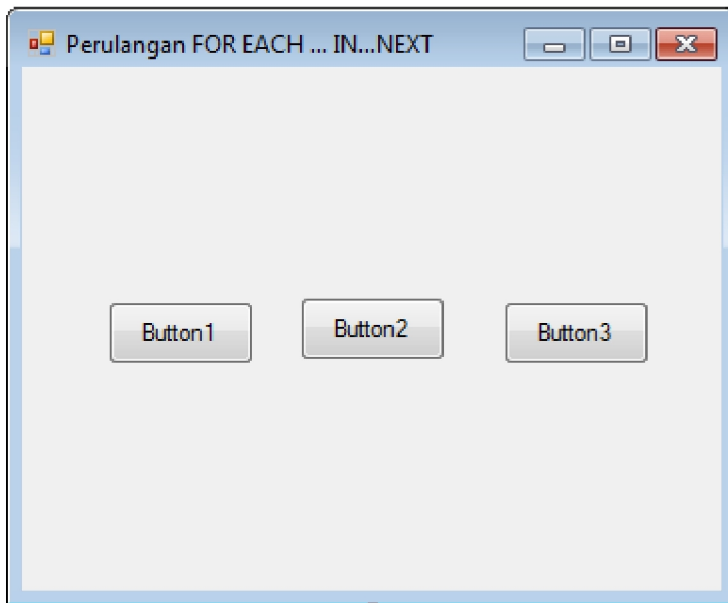
Dim X%
X = TxtAngka.Text
Do
  X = X Mod 2
  If X = 0 Then
    MsgBox("Bilangan Genap")
  Exit Do
  ElseIf X = 1 Then
    MsgBox("Bilangan Ganjil")
  Exit Do
  End If
Loop While Not X

```

5. Jalankan aplikasi dengan menekan tombol **F5** (di keyboard), atau melalui ikon Start Debugging di toolbar, atau melalui menu **Debug > Start Debugging**.
6. Simpan aplikasi **Anda**.

### Perulangan FOR EACH ... IN...NEXT

1. Tambahkan **Form baru**, dengan nama **Form5**.
2. Desain Tampilan form seperti gambar berikut ini :



3. Tambahkan kontrol ke form seperti pada gambar diatas dan atur propertinya seperti tabel berikut:

Object	Properties	Nilai
Form5	Name	Form5
	Text	Perulangan FOR EACH ... IN...NEXT
Button1	Name	Button1
	Text	Button1
Button2	Name	Button2
	Text	Button2
Button3	Name	Proses
	Text	Button3

4. Klik ganda pada Proses, kemudian ketikkan kode program berikut ini :

```
For Each ctrl In Controls  
    ctrl.Text = "Tombol"  
Next
```

5. Jalankan aplikasi dengan menekan tombol **F5** (di keyboard), atau melalui ikon Start Debugging di toolbar, atau melalui menu **Debug > Start Debugging**.
6. Simpan aplikasi **Anda**.

## BAB VII ARRAY

### 7.1 Tujuan Pembelajaran

- Memahami dan mengenal Array (array satu dimensi dan array multi dimensi)
- Dapat membuat aplikasi sederhana dengan menggunakan array satu dimensi dan array multi dimensi

### 7.2 Teori Array

Array adalah sekumpulan data yang memiliki tipe yang sama, sejumlah tetap, serta disusun secara terstruktur dan disimpan dalam satu variabel yang sama, dan diurutkan dengan index.

#### Array Satu Dimensi

Suatu array yang nilai dan ukurannya yang sudah ditentukan terlebih dahulu, dan memiliki satu dimensi.

Bentuk Umum :

**Dim Array[indeks] As Tipe\_Data**

Contoh :

**Dim Arr(2) As String**

Arr(0) = TxtNim.Text

Arr(1) = TxtNama.Text

Arr(2) = TxtProdi.Text

#### Array Multi Dimensi

Suatu array yang fungsinya hampir sama dengan array satu dimensi hanya saja pada array multi dimensi ini mewakili nilai table yang terdiri dari informasi yang diatur dalam baris dan kolom. Untuk mendefinisikan elemen table tertentu, kita harus menentukan dua indeks, pertama mengidentifikasi elemen baris dan yang mengidentifikasi elemen kolom. Array multidimensi memiliki lebih dari dua dimensi.

Bentuk Umum :

**Dim Array[indeks,Indeks] As Tipe\_Data**

**Dim Array[indeks,indeks,indeks] As Tipe\_Data**

Contoh :

**Dim Arr(3, 1) As String**

Arr(0, 0) = "NIM"

Arr(0, 1) = "NAMA"

Arr(1, 0) = "JENIS KELAMIN"

Arr(1, 1) = "PRODI"

Arr(2, 0) = "Laki-laki"

Arr(2, 1) = "Perempuan"

Arr(3, 0) = "Manajemen Informatika"

Arr(3, 1) = "Teknik Informatika"

### 7.3 PROSEDUR PRAKTIKUM

#### Array Satu Dimensi

1. Buat **project baru**, dengan nama **Prak9**.

2. Desain Tampilan form seperti gambar berikut ini :

3. Tambahkan kontrol ke form seperti pada gambar diatas dan atur propertinya seperti tabel berikut:

Object	Properties	Nilai
Form1	Name Text	Form1 Array Satu Dimensi
Label1	Text	NIM
Label2	Text	Nama
Label3	Text	Prodi
TextBox1	Name	TxtNIM
TextBox2	Name	TxtNama
TextBox3	Name	TxtProdi
ListView1	Name	ListView1
Button1	Name Text	BtnTambah Tambah

4. Klik ganda pada Form Load, kemudian ketikkan kode program berikut ini :

```

ListView1.GridLines = True
ListView1.View = View.Details
ListView1.Columns.Add("NIM")
ListView1.Columns.Add("NAMA", 115)
ListView1.Columns.Add("PRODI", 110)
TxtNim.Text = 1
TxtNama.Focus()

```

5. Klik ganda pada BtnTambah, kemudian ketikkan kode program berikut ini :

```

Dim Arr(2) As String
Arr(0) = TxtNim.Text
Arr(1) = TxtNama.Text
Arr(2) = TxtProdi.Text

Dim listitem As ListViewItem
listitem = New ListViewItem
listitem = ListView1.Items.Add(Arr(0))
listitem.SubItems.Add(Arr(1))
listitem.SubItems.Add(Arr(2))

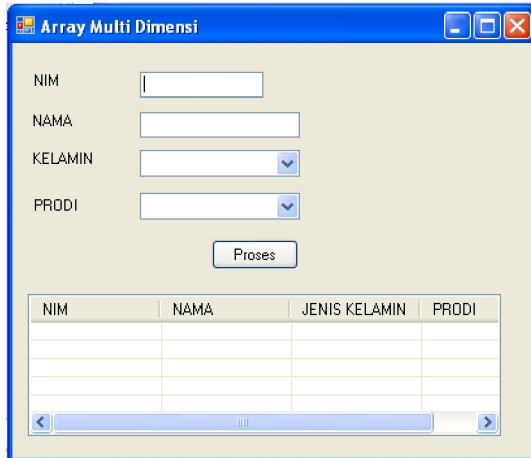
TxtNim.Text = TxtNim.Text + 1
TxtNama.Text = ""
TxtProdi.Text = ""
TxtNama.Focus()

```

6. Jalankan aplikasi dengan menekan tombol **F5** (di keyboard), atau melalui icon Start Debugging di toolbar, atau melalui menu **Debug > Start Debugging**.
7. Simpan aplikasi **Anda**.

### Array Multi Dimensi

1. Tambahkan **Form baru**, dengan nama **Form2**.
2. Desain Tampilan form seperti gambar berikut ini :



3. Tambahkan kontrol ke form seperti pada gambar diatas dan atur propertinya seperti tabel berikut:

Object	Properties	Nilai
Form1	Name Text	Form1 Array Multi Dimensi
Label1	Text	NIM
Label2	Text	Nama
Label3	Text	Kelamin
Label4	Text	Prodi
TextBox1	Name	TxtNIM
TextBox2	Name	TxtNama
Combo1	Name	CmbKelamin
Combo2	Name	CmbProdi
ListView1	Name	ListView1
Button1	Name Text	BtnProses Proses

4. Klik ganda pada **Form Load**, kemudian ketikkan kode program berikut ini :

```

Dim Arr(3, 1) As String
Arr(0, 0) = "NIM"
Arr(0, 1) = "NAMA"
Arr(1, 0) = "JENIS KELAMIN"
Arr(1, 1) = "PRODI"
Arr(2, 0) = "Laki-laki"
Arr(2, 1) = "Perempuan"
Arr(3, 0) = "Manajemen Informatika"
Arr(3, 1) = "Teknik Informatika"

ListView1.GridLines = True

```

```

ListView1.View = View.Details

For Baris = 0 To 1
    For Kolom = 0 To 1
        ListView1.Columns.Add(Arr(Baris, Kolom), 100)
    Next Kolom
Next Baris

For Baris = 2 To 2
    For Kolom = 0 To 1
        CmbKelamin.Items.Add(Arr(Baris, Kolom))
    Next Kolom
Next Baris

For Baris = 3 To 3
    For Kolom = 0 To 1
        CmbProdi.Items.Add(Arr(Baris, Kolom))
    Next Kolom
Next Baris

```

5. Klik ganda pada **BtnProses**, kemudian ketikkan kode program berikut ini :

```

Dim Arr(3) As String
Arr(0) = TxtNIM.Text
Arr(1) = TxtNAMA.Text
Arr(2) = CmbKelamin.Text
Arr(3) = CmbProdi.Text

Dim listitem As ListViewItem
listitem = New ListViewItem
listitem = ListView1.Items.Add(Arr(0))
listitem.SubItems.Add(Arr(1))
listitem.SubItems.Add(Arr(2))
listitem.SubItems.Add(Arr(3))
TxtNIM.Text = TxtNIM.Text + 1
TxtNAMA.Text = ""
CmbKelamin.Text = ""
CmbProdi.Text = ""
TxtNAMA.Focus()

```

6. Jalankan aplikasi dengan menekan tombol **F5** (di keyboard), atau melalui ikon Start Debugging di toolbar, atau melalui menu **Debug > Start Debugging**.
7. Simpan aplikasi **Anda**.

## BAB VIII PROCEDURE, FUNCTION DAN MODULE

### 8.1 Tujuan Pembelajaran

- Memahami dan mengenal Procedure, function dan modul
- Dapat membuat aplikasi sederhana dengan menggunakan Procedure, function dan modul

### 8.2 Teori Procedure, Function Dan Module

Sejauh ini, anda telah menuliskan sub procedure yang dibuat secara otomatis pada saat menggunakan event . Sub procedure merupakan blok kode yang mempunyai nama dan berisi perintah yang dapat dipanggil suatu waktu. Artinya, hanya sekali dibuat yang kemudian di panggil berkali-kali sesuai dengan kebutuhan.

Bentuk Umum :

```
[Private|Public] Sub subname[(argumentlist)]
    Statements
End sub
```

Sub Procedure dapat dipanggil dengan perintah :

```
[call] subname[(argumentlist)]
```

Keterangan :

Sub procedure (*Procedure*) sebaiknya menggunakan kata kerja (verb)

Pemanggilan procedure yang masih dalam satu kelas (class) atau modul (module) dapat dilakukan dengan kata kunci Me.

Procedure dapat menerima argument melalui nilai (by value) atau juga reference (by reference). Jika menggunakan by value, nilai pada argument tidak berubah. Sedangkan jika menggunakan by reference, nilai argument yang digunakan dapat berubah sesuai dengan perintah dalam procedure.

Secara default, argument dimasukkan melalui nilai (by value) dapat menggunakan kata kunci ByVal dan ByRef untuk By Reference.

Syntaks argument :

```
[ByVal|ByRef] variabelname as type
```

Contoh :

1. Procedure dengan argument di passing sebagai value

```
Sub HitungVal(ByVal as Decimal)
    A+=1
    textAakhir.text=a
end sub
```

2. Procedure dengan argument di passing sebagai value

```
Sub HitungRef(byRef as Decimal)
    A+=1
    textAakhir.text=a
end sub
```

3. pemanggilan procedure  
HitungVal(textAwal.Text)

### Function (Fungsi)

Fungsi mempunyai banyak kemiripan dengan sub procedure. Bedanya, fungsi selalu mengembalikan nilai (return value)

Sintaks fungsi dituliskan sebagai berikut :

```
[Private|Public] function functionname[(argumentlist)] [As type]
    Statements{
        Functionname=expression | {return expression}
    End function
```

Keterangan :

- Perbedaan antara fungsi dengan procedure hanyalah pada pengembalian nilai saja (return value). Sebagai contoh :  
Menggunakan argument berupa nilai (by value) dan reference (by reference)
- Pengembalian nilai dapat dilakukan dengan menggunakan operator assignment “=” pada fungsi atau biasanya dengan perintah return.

Contoh :

```
Private Function cekData() as Boolean
    if IsNumeric(TextAwal.text) then
        return true
    else
        return false
    endif
end function
```

Pemanggilan fungsi

```
if cekData()=true then
    HitungRef(textAwal.text)
else
    messageBox.Show(“Data yang dimasukkan bukan angka”,“Kesalahan data”,
    MessageBoxButtons.OK,
    MessageBoxIcon.Error)
endif
```

### Modul (Module)

Modul merupakan bagian yang sengaja dipisahkan untuk memudahkan pemrograman. Dalam module dapat dimasukkan procedure dan fungsi dan kemudian digunakan oleh beberapa form.

- Umumnya modul dideklarasikan secara public (public) agar dapat digunakan di dalam kode dimanapun
- Modul dapat diisi dengan main procedure yang akan dijalankan pertama kali saat program dimulai
- Modul merupakan salah satu argumentasi code reuse yang bertujuan agar program lebih efisien.

Sintaks pembuatan modul sebagai berikut :

```

Module moduleName
    Statement
End module

```

Contoh :

```

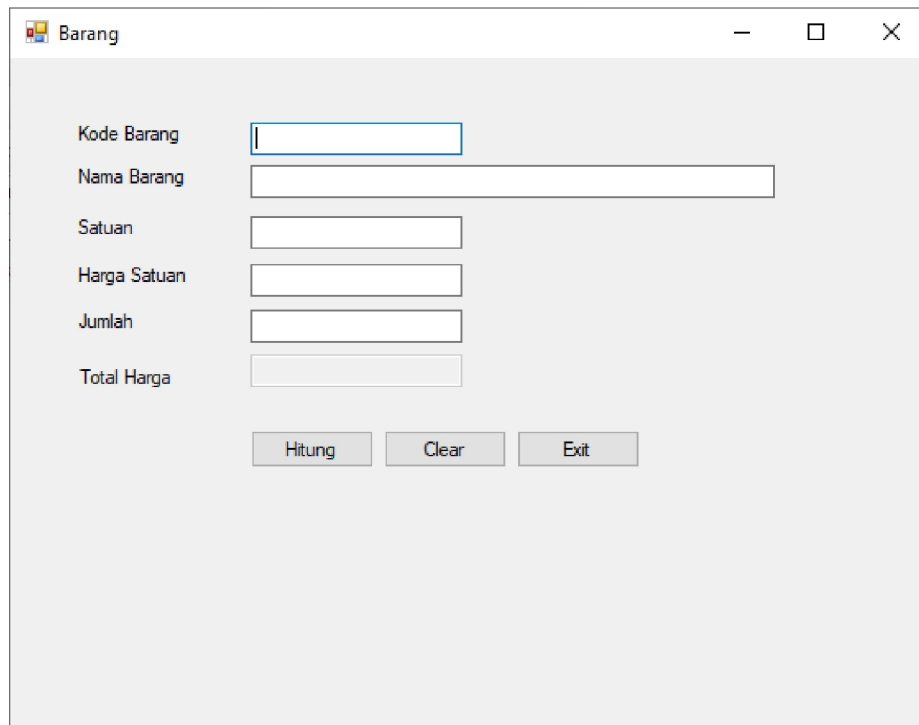
Module Module
    Public intResponse, RoleId, Msg As Integer
    Public Username, Password, Nama As String
End Module

```

### 8.3 Praktikum

#### Procedure dan Function

1. Buat **project baru**, dengan nama **Prak10**.
2. Desain Tampilan form seperti gambar berikut ini :



3. Tambahkan kontrol ke form seperti pada gambar diatas dan atur propertinya seperti tabel berikut:

Object	Properties	Nilai
Form1	Name Text	Form1 Barang
Label1	Text	Kode Barang
Label2	Text	Nama Barang
Label3	Text	Satuan

Label4	Text	Harga
Label5	Text	Jumlah
Label6	Text	Total Harga
TextBox1	Name	TxtKodeBrg
TextBox2	Name	TxtNamaBrg
TextBox3	Name	TxtSatuan
TextBox4	Name	TxtHrgSatuan
TextBox5	Name	TxtHarga
TextBox6	Name	TxtTot
Button1	Name Text	BtnHitung Hitung
Button2	Name Text	BtnClear Clear
Button3	Name Text	BtnExit Exit

4. Klik ganda pada Form, kemudian buatlah sub procedure **Bersih** berikut ini :

```

Sub Bersih()
    TxtKodeBrg.Text = ""
    TxtNamaBrg.Text = ""
    TxtSatuan.Text = ""
    TxtHarga.Text = ""
    TxtHrgSatuan.Text = ""
    TxtTot.Text = ""
    TxtKodeBrg.Focus()
    TxtTot.Enabled = False
End Sub

```

5. Panggil procedure yang anda buat diatas dengan cara, klik ganda pada form load ketikkan kode program berikut ini :

```
Call Bersih()
```

6. Panggil procedure yang anda buat diatas pada BtnClear dengan cara, klik ganda pada BtnClear kemudian ketikkan kode program berikut ini :

```
Call Bersih()
```

7. Jalankan aplikasi dengan menekan tombol **F5** (di keyboard), atau melalui icon Start Debugging di toolbar, atau melalui menu **Debug > Start Debugging**.

8. Buatlah Function Hitung, berikut ini :

```
Private Function Hitung()  
    Dim Harga, HrgSatuan, Total As Integer  
    Harga = TxtHarga.Text  
    HrgSatuan = TxtHrgSatuan.Text  
    Total = Harga * HrgSatuan  
    TxtTot.Text = Total  
    Return Total  
End Function
```

9. Buatlah **Procedure CekDataKosong** dan sekaligus memanggil **function Hitung**, berikut ini :

```
Sub CekDataKosong()  
    If TxtKodeBrg.Text = "" Then  
        MessageBox.Show("Kode Barang harus diisi", "Konfirmasi", MessageBoxButtons.OK,  
        MessageBoxIcon.Warning)  
        TxtKodeBrg.Focus()  
    ElseIf TxtNamaBrg.Text = "" Then  
        MessageBox.Show("Nama Barang harus diisi", "Konfirmasi", MessageBoxButtons.OK,  
        MessageBoxIcon.Warning)  
        TxtNamaBrg.Focus()  
    ElseIf TxtSatuan.Text = "" Then  
        MessageBox.Show("Satua Barang harus diisi", "Konfirmasi", MessageBoxButtons.OK,  
        MessageBoxIcon.Warning)  
        TxtSatuan.Focus()  
    ElseIf TxtHarga.Text = "" Then  
        MessageBox.Show("Harga Barang harus diisi", "Konfirmasi", MessageBoxButtons.OK,  
        MessageBoxIcon.Warning)  
        TxtHarga.Focus()  
    ElseIf TxtHrgSatuan.Text = "" Then  
        MessageBox.Show("Harga Satuan Barang harus diisi", "Konfirmasi", MessageBoxButtons.OK,  
        MessageBoxIcon.Warning)  
        TxtHrgSatuan.Focus()  
    Else  
        Call Hitung()  
    End If  
End Sub
```

10. Memanggil procedure CekDataKosong dengan cara, klik ganda pada tombol BtnHitung:

```
Call CekDataKosong()
```

11. Jalankan aplikasi dengan menekan tombol **F5** (di keyboard), atau melalui icon Start Debugging di toolbar, atau melalui menu **Debug > Start Debugging**.
12. Klik ganda pada tombol BtnExit, Kemudian ketikkan kode program berikut ini :

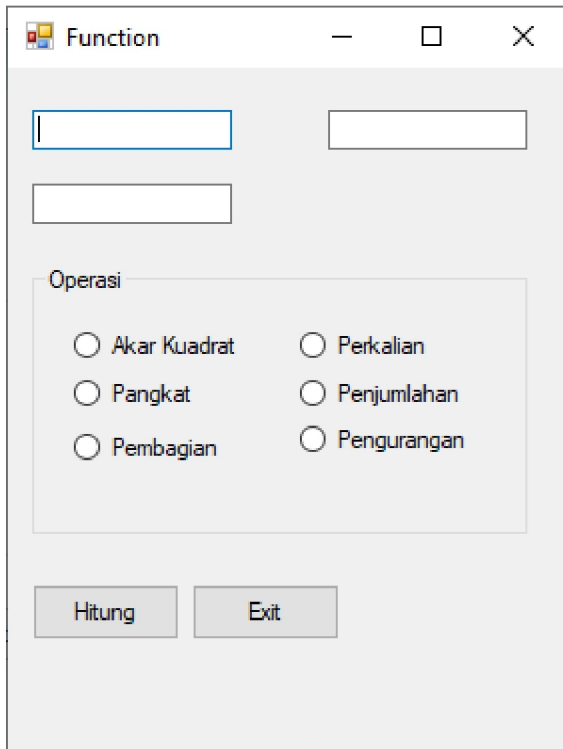
```
Dim Tutup As String

Tutup = MessageBox.Show("Yakin tutup form ini ?",
"Konfirmasi",
MessageBoxButtons.YesNo, MessageBoxIcon.Question)
If TUTUP = MsgBoxResult.Yes Then
    End
Else
    Exit Sub
End If
```

13. Jalankan aplikasi dengan menekan tombol **F5** (di keyboard), atau melalui icon Start Debugging di toolbar, atau melalui menu **Debug > Start Debugging**.
14. Simpan aplikasi **Anda**.

### Function

1. Tambahkan **Form baru**, dengan nama **Form2**.
2. Desain Tampilan form seperti gambar berikut ini :



3. Tambahkan kontrol ke form seperti pada gambar diatas dan atur propertinya seperti tabel berikut:

Object	Properties	Nilai
Form1	Name Text	Form1 Function
TextBox1	Name	TxtVar1
TextBox2	Name	TxtVar2
TextBox3	Name	TxtHasil
GroupBox1	Text	Operasi
RadioButton1	Name Text	RdbAkar Akar Kuadrat
RadioButton2	Name Text	RdbPangkat Pangkat
RadioButton3	Name Text	RdbPembagian Pembagian
RadioButton4	Name Text	RdbPerkalian Perkalian
RadioButton5	Name Text	RdbPenjumlahan Penjumlahan
RadioButton6	Name Text	RdbPengurangan Pengurangan
Button1	Name	BtnHitung

	Text	Hitung
Button2	Name	BtnExit
	Text	Exit

4. Klik kanan pada form, kemudian pilih menu view, kemudian ketik deklarasi masing-masing fungsi sebagai berikut ini :

```

Function Akar(ByVal a As Decimal) As String
    Return Math.Sqrt(a)
End Function

Function Pangkat(ByVal a As Decimal, ByVal b As Decimal) As String
    Return a ^ b
End Function

Function Pembagian(ByVal a As Decimal, ByVal b As Decimal) As String
    Return a / b
End Function

Function Perkalian(ByVal a As Decimal, ByVal b As Decimal) As String
    Return a * b
End Function

Function Pemjumlahan(ByVal a As Decimal, ByVal b As Decimal) As String
    Return a + b
End Function

Function Pengurangan(ByVal a As Decimal, ByVal b As Decimal) As String
    Return a - b
End Function

```

5. Klik ganda pada Tombol Hitung, kemudian ketik kode program berikut ini :

```

If RdbAkar.Checked = True Then
    TxtHasil.Text = Akar(TxtVar1.Text)
Elseif RdbPangkat.Checked = True Then
    TxtHasil.Text = Pangkat(TxtVar1.Text, TxtVar2.Text)
Elseif RdbPembagian.Checked = True Then
    TxtHasil.Text = Pembagian(TxtVar1.Text, TxtVar2.Text)
Elseif RdbPerkalian.Checked = True Then
    TxtHasil.Text = Perkalian(TxtVar1.Text, TxtVar2.Text)
Elseif RdbPenjumlahan.Checked = True Then
    TxtHasil.Text = Pemjumlahan(TxtVar1.Text, TxtVar2.Text)
Elseif RdbPengurangan.Checked = True Then
    TxtHasil.Text = Pengurangan(TxtVar1.Text, TxtVar2.Text)
End If

```

6. Jalankan aplikasi dengan menekan tombol **F5** (di keyboard), atau melalui icon Start Debugging di toolbar, atau melalui menu **Debug > Start Debugging**.
7. Simpan aplikasi **Anda**

## BAB IX DATABASE

### 9.1 Tujuan Pembelajaran

- Memahami dan mengenal database Microsoft SQL Server Compact (SQL Server CE)
- Dapat membuat database SQL Server CE
- Dapat membuat aplikasi database dengan akses data lewat komponen ADO.Net

### 9.2 Teori Database

Database adalah informasi yang tersimpan dan tersusun rapi di dalam suatu tempat, dan dapat dengan mudah dimanipulasi seperti menambah data, menghapus, mencari, mengatur informasi yang kita butuhkan.

#### Microsoft SQL Server Compact 3.5 atau SQL Server Compact Edition (SQL Server CE)

Microsoft SQL Server Compact 3.5 adalah aplikasi database yang dikeluarkan oleh Microsoft, SQL Server Compact 3.5 dapat digunakan untuk membuat aplikasi computer desktop, Smart Device dan Tablet PCs. Jenis file yang dihasilkan SQL Server CE adalah (\*.sdf).

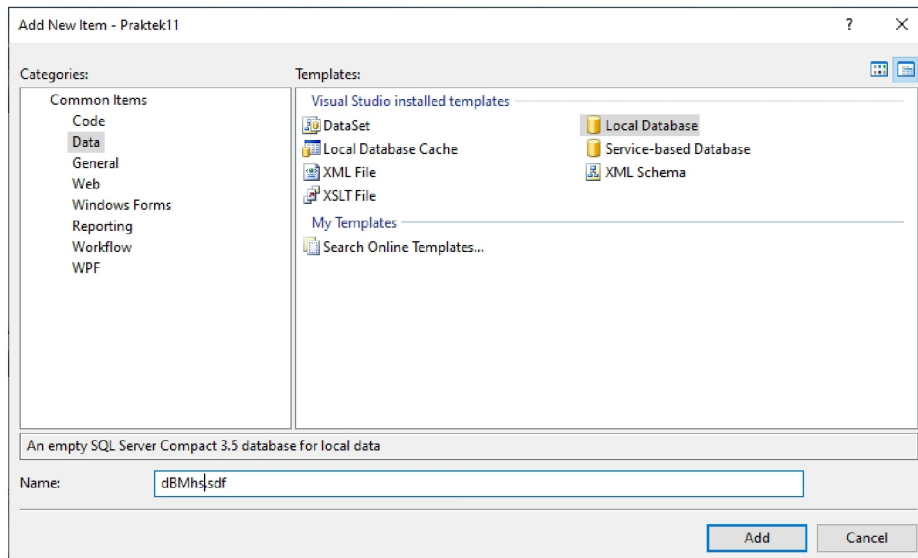
#### ADO.Net 3.5

**ADO.Net 3.5** merupakan hasil pengembangan dari ADO.Net 2.x sebelumnya, yang merupakan paket untuk mengakses dan mengelolah sumber data (database) atau sumber data lainnya. Berikut ini komponen .NET Data Provider dari ADO.Net.3.5 antara lain :

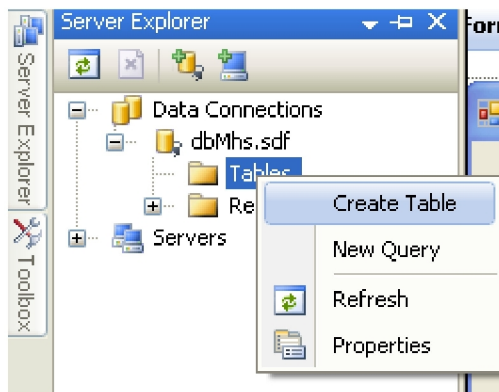
### 9.3 Praktikum

#### Latihan Membuat Database SQL Server CE

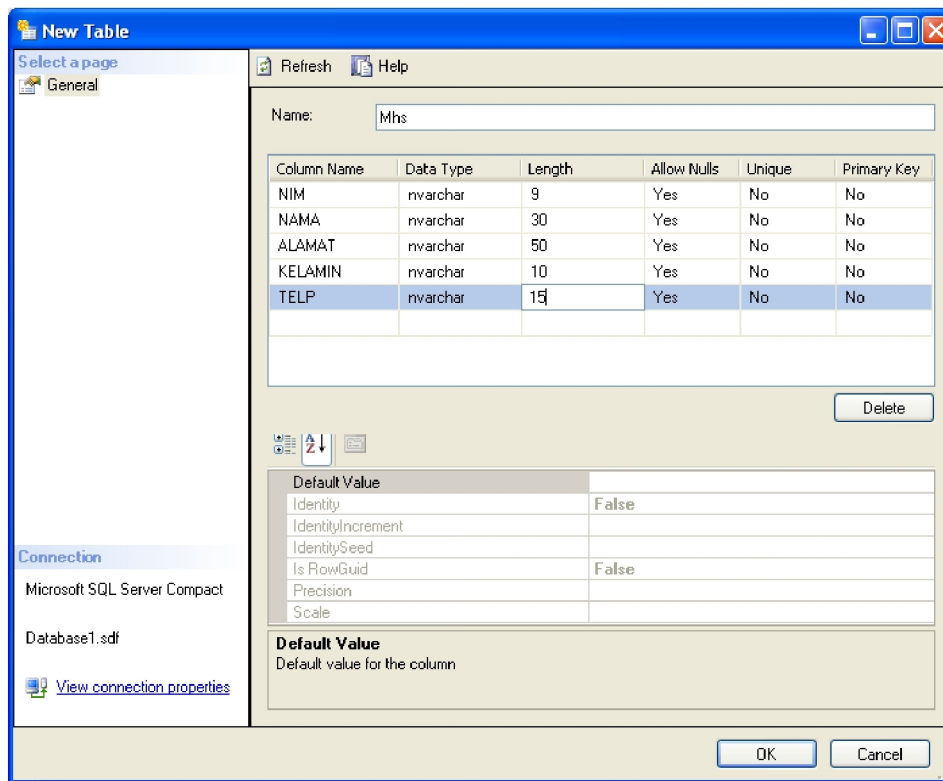
1. Buat **project baru**, dengan nama **Prak11**
2. Klik **Add New Item...**



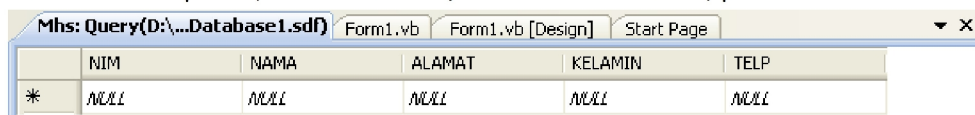
3. Pada Kotak Dialog **Add New Item**, klik **Local Database** dan pad Name : **dbMhs.sdf**, lalu klik tombol **Add**



4. Pada Server Explorer, klik kanan **Tables**, kemudian klik **Create Table**



5. Pada kotak dialog New Table di atas, diisi name :**Mhs**, kemudian buatlah design table seperti gambar di atas.
6. Setelah itu klik OK
7. Pada Server Explorer, klik kanan **Mhs**, kemudian klik kanan, pilih **show data table**



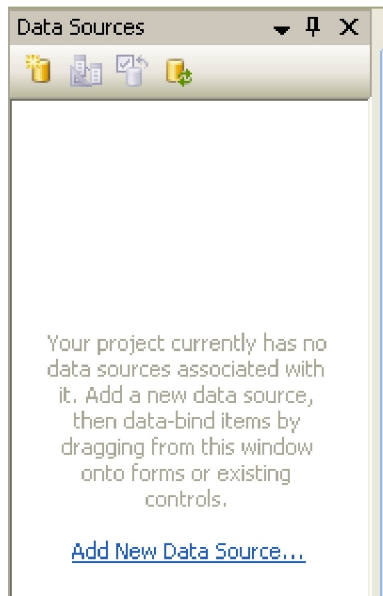
8. Isi data pada table yang anda desain diatas.

## Latihan Database Access dan Koneksi dengan ADO.Net 3.5

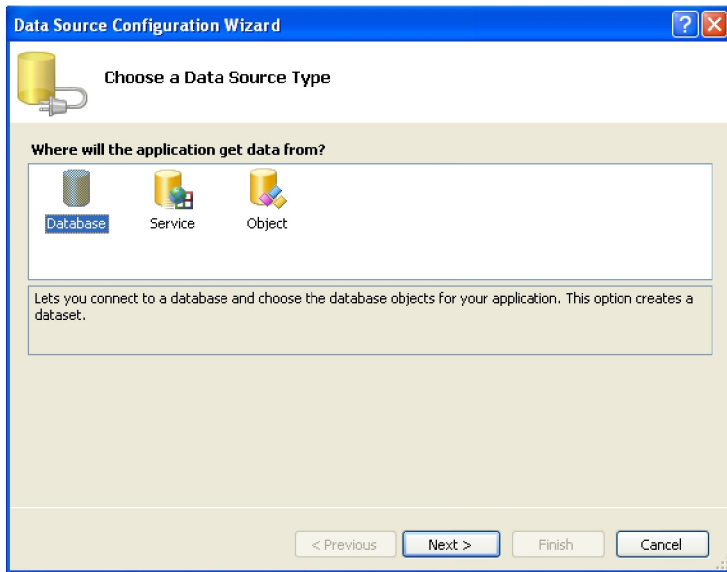
1. Buat Database pada Microsoft Access, dengan struktur sebagai berikut :

Field Name	Data Type	
NIM	Text	
NAMA	Text	
ALAMAT	Text	
TELP	Text	
KELAMIN	Text	

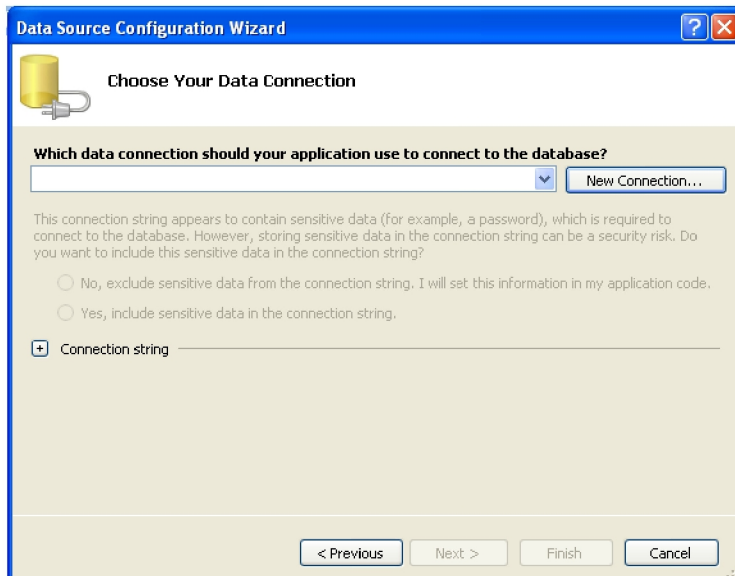
2. Buat **project baru**, dengan nama **Prak8**
3. Klik Tab **Data**, lalu klik **Show data source**



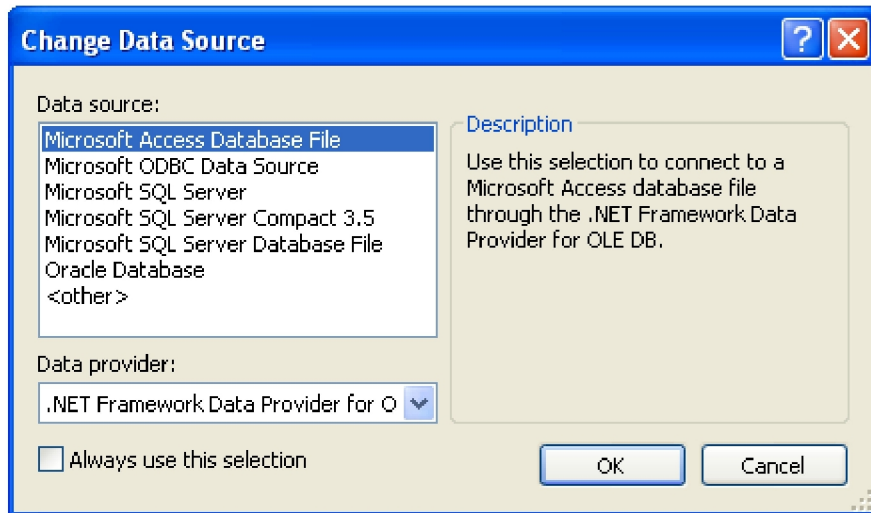
4. Klik **Add New Data Source ....**



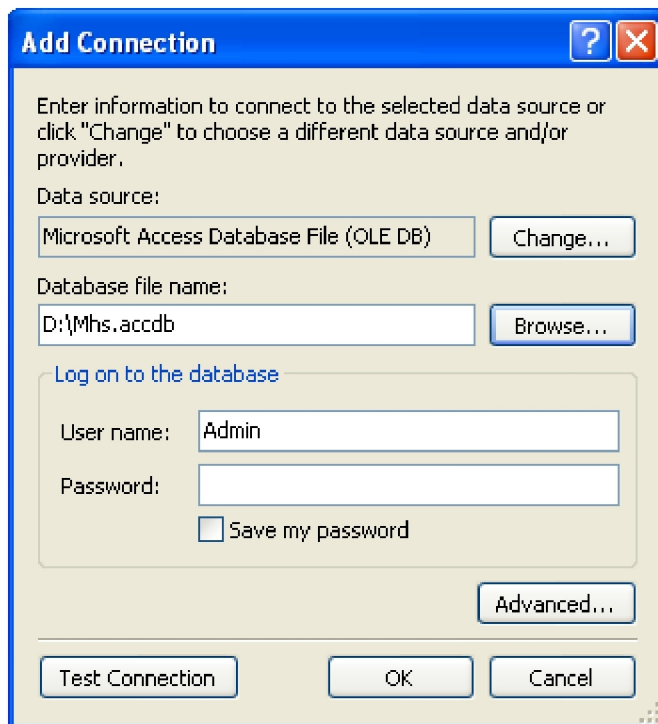
5. Pilih Database, Klik Next



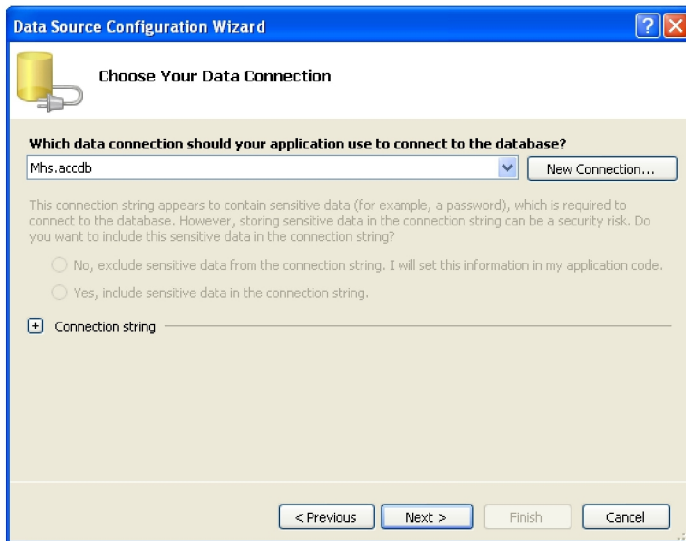
6. Klik New Connection, Pilih Microsoft Access Data File



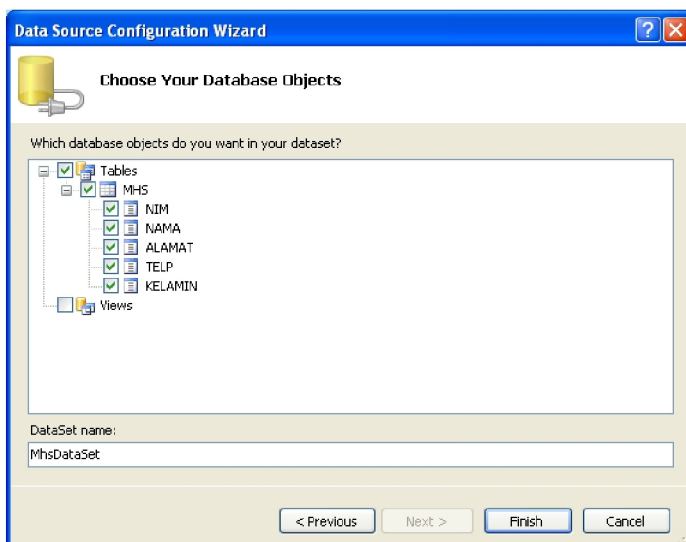
7. Klik **Continue**, pada kotak **dialog add connection**, klik tombol **Browse**



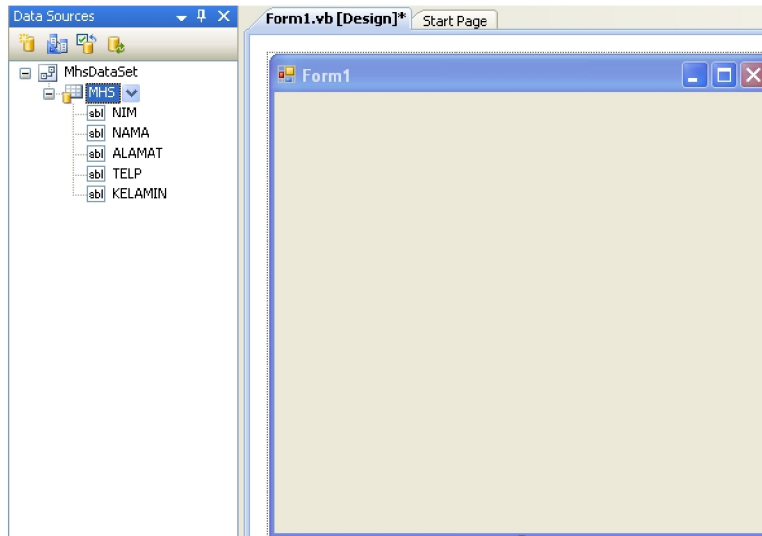
8. Pilih database access : **dbMhs**
9. Klik tombol **test connection**, kemudian klik **OK**



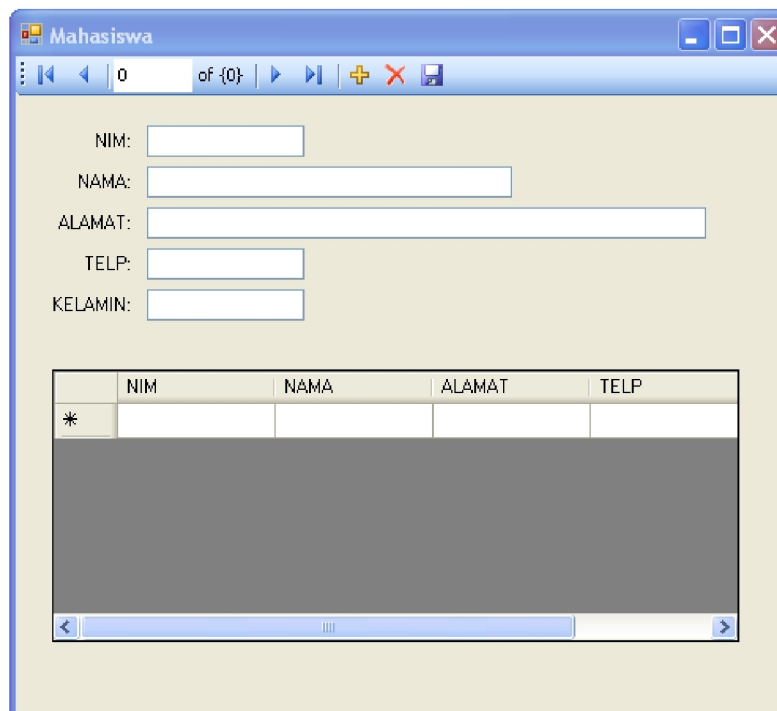
10. Klik **Next**, pada kotak diaolog **configuration wizard**, klik **next**



11. Beri tanda cek pada table, klik **finish**



12. Kemudian, **klik** dan **drag** ke area form Masing-masing **field** dari table **Mhs**
13. Dan hasil drag dari masing-masing field dari table mhs seperti gambar berikut ini :



14. Jalankan aplikasi dengan menekan tombol **F5** (di keyboard), atau melalui icon Start Debugging di toolbar, atau melalui menu **Debug > Start Debugging**.
15. Simpan aplikasi **Anda**.

## BAB X EXCEPTION HANDLING

### 10.1 Tujuan Pembelajaran

- Memahami dan mengenal Exception Handling
- Dapat membuat aplikasi sederhana dengan menggunakan Exception Handling

### 10.2 Teori Exception Handling

#### EXCEPTION HANDLING

Perintah penanganan kesalahan atau yang lebih dikenal dengan sebutan Exception Handling Perintah ini digunakan untuk menangani kesalahan dalam menjalankan aplikasi, dengan adanya perintah ini setiap kesalahan akan diatasi secara otomatis sesuai dengan perintah penanganan kesalahan yang telah dibuat, sehingga tidak terjadi kemacetan aplikasi yang sedang berjalan.

#### Struktur Exception Handling

Struktur Exception Handling atau penanganan kesalahan pada visual basic 2008 antara lain :

#### Try ... Catch ... Finally ... End Try

Merupakan perintah penanganan kesalahan yang berfungsi untuk menangani kesalahan dalam menjalankan aplikasi. Bentuk penulisan :

```
Try
    [ Perintah ]
Catch
    [Tampilkan kesalahan]
Finally
    [ Kode program setelah perintah Try dan Catch ]
End Try
```

Contoh :

```
Try
Dim X%
X = TxtAngka.Text
Catch ex As Exception
MsgBox(ex.ToString)
MsgBox("Input angka saja")
Finally
TxtAngka.Text=""
TxtAngka.Focus()
End Try
```

#### Throw

Throw merupakan perintah penanganan kesalahan yang berfungsi untuk menangani kesalahan apabila perintah try...catch ... Finally ... End. Try tidak dapat menangani suatu kesalahan dari jawaban aplikasi. Contoh penulisan :

```
Try
    [ Perintah ]
Catch ex As Exception
    [Tampilkan kesalahan]
Throw ex
    // mengembalikan ke kode pemanggil
Finally
    [ Kode program setelah perintah Try dan Catch ]
    Perintah ini bersifat optional bias dibuat bias tidak
End Try
```

Contoh :

```
Try
Dim X%
X = TxtAngka.Text
Catch ex As Exception
MsgBox(ex.ToString)
MsgBox("Input angka saja")
Throw ex
Finally
TxtAngka.Text=""
TxtAngka.Focus()
End Try
```

**On Error Resume Next**

Merupakan perintah penanganan kesalahan apabila terjadi kesalahan terhadap suatu baris tertentu maka baris tersebut akan di abaikan.

Contoh :

```
On Error Resume Next
Dim A As Integer
X = TextBox1.Text
TextBox1.Text="String"
MsgBox(ex.ToString)
MsgBox("Abaikan Kesalahan")
```

**On Error Goto**

Merupakan perintah penanganan kesalahan apabila terjadi kesalahan terhadap suatu baris tertentu maka perintah akan melompat ke baris yang dituju akan dijalankan.

Contoh :

```
On Error Goto Pesan
Dim A As Integer
X = TextBox1.Text
TextBox1.Text="String"
```

Exit Sub

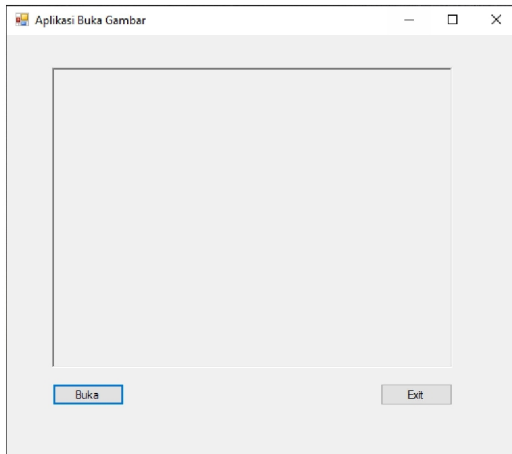
Pesan :

MsgBox("Abaikan Kesalahan")

### 10.3 Praktikum

#### Latihan 1

1. Buat **project baru**, dengan nama **Prak12**.
2. Desain Tampilan form seperti gambar berikut ini :



3. Tambahkan kontrol ke form seperti pada gambar diatas dan atur propertinya seperti tabel berikut:

Object	Properties	Nilai
Form1	Name Text	Form1 Aplikasi Buka Gambar
OpenFileDialog1	Text	ofdBuka
PictureBox1	Name	PcbGambar
Button1	Name Text	BtnBuka Buka
Button2	Name Text	BtnExit Exit

4. Klik ganda pada Tombol Buka, kemudian ketik kode program berikut ini :

```

If OfdBuka.ShowDialog() = Windows.Forms.DialogResult.OK Then
    Try
        PcbGambar.Image = Image.FromFile(OfdBuka.FileName)
        Me.Text = "Membuka File " + OfdBuka.FileName
    Catch ex As Exception
        MessageBox.Show("File Gagal dibuka", "Error",
        MessageBoxButtons.OK,
        MessageBoxIcon.Error)
    End Try
End If

```

5. Klik ganda pada tombol BtnExit, Kemudian ketikkan kode program berikut ini :

```

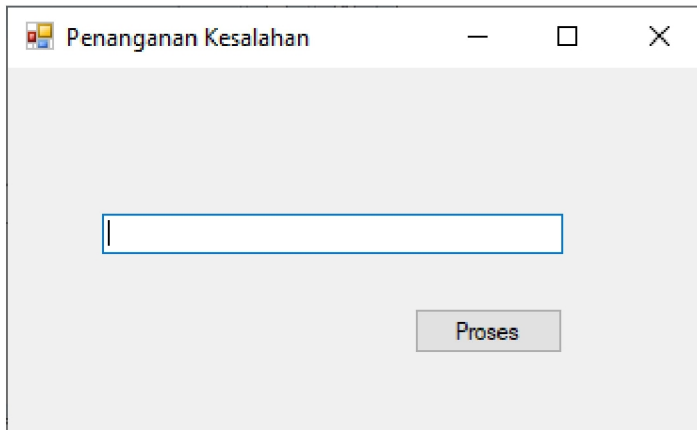
    Dim Tutup As String
    Tutup = MessageBox.Show("Yakin tutup form ini ?",
    "Konfirmasi",
    MessageBoxButtons.YesNo, MessageBoxIcon.Question)
    If Tutup = MsgBoxResult.Yes Then
        End
    Else
        Exit Sub
    End If

```

6. Jalankan aplikasi dengan menekan tombol **F5** (di keyboard), atau melalui icon Start Debugging di toolbar, atau melalui menu **Debug > Start Debugging**.
7. Simpan aplikasi **Anda**.

## Latihan 2

1. Tambahkan **Form baru**, dengan nama **Form2**.
2. Desain Tampilan form seperti gambar berikut ini :



3. Tambahkan kontrol ke form seperti pada gambar diatas dan atur propertinya seperti tabel berikut:

Object	Properties	Nilai
Form1	Name Text	Form1 Penanganan Kesalahan
TextBox1	Name	TxtAngka
Button1	Name Text	BtnProses Proses

4. Double klik pada tombol BtnProses, kemudian ketik kode Program berikut ini :

```

Try
  Dim X As Integer
  X = TxtAngka.Text
  Do
    X = X Mod 2
    If X = 0 Then
      MsgBox.Show("Bilangan Genap", "Informasi")
    Exit Do
    ElseIf X = 1 Then
      MsgBox.Show("Bilangan Ganjil", "Informasi")
    Exit Do
  End If
Loop While Not X

Catch ex As Exception
  MsgBox("Salah menginput data / input angka")
Finally
  TxtAngka.Text=""
  TxtAngka.Focus()

```

End Try

5. Jalankan aplikasi dengan menekan tombol **F5** (di keyboard), atau melalui icon Start Debugging di toolbar, atau melalui menu **Debug > Start Debugging**.
6. Simpan aplikasi **Anda**

## **BAB XI**

### **KEAMANAN DAN ADMINISTRASI DATABASE**

#### **12.1 Tujuan Pembelajaran :**

- Dapat menjelaskan Database Security
- Dapat menjelaskan ukuran untuk control berbasis computer Countermeasures-Computer-Based Controls
- Dapat menjelaskan dan memahami DBMS and Web Security
- Dapat menjelaskan Database Administration

#### **12.2 Dasar Teori**

Mirip dengan sumber daya perusahaan, data merupakan sumber daya yang sangat berharga yang perlu diatur dan dikelola dengan hati-hati. Dengan kata lain, DBMS harus menjamin keamanan database. Perlindungan database dari akses yang tidak diinginkan, termasuk pencurian dan perubahan, disebut sebagai keamanan. Pembahasan bagaimana keamanan basis data mencakup topik yang lebih luas terkait dengan perlindungan basis data dan lingkungannya selain layanan yang ditawarkan oleh DBMS.

#### **12.3 Database Securiti**

Teknik yang menjaga database dari ancaman jahat atau tidak disengaja adalah definisi keamanan database. Sebuah pelanggaran keamanan dapat mempengaruhi komponen lain dari sistem, yang pada gilirannya dapat mempengaruhi database, oleh karena itu pertimbangan keamanan tidak hanya berkaitan dengan data yang disimpan dalam database. Akibatnya, data, orang, perangkat lunak, dan perangkat keras adalah bagian dari keamanan basis data. Kontrol yang tepat, yang diuraikan dalam tujuan misi unik untuk sistem, diperlukan agar implementasi keamanan berhasil. Meskipun sering diabaikan atau diabaikan di masa lalu, organisasi saat ini menjadi semakin sadar akan kebutuhan akan keamanan. Meningkatnya jumlah data bisnis penting yang disimpan di komputer dan kesadaran bahwa kehilangan atau ketidakterseediaannya dapat menimbulkan efek bencana adalah kekuatan pendorong di balik perubahan ini.

Database adalah sumber daya bisnis penting yang perlu dilindungi secara aman dengan pengamanan yang sesuai. Konsekuensinya, keadaan berikut terhubung ke keamanan database:

1. Pencurian dan penipuan.
2. Kehilangan kerahasiaan.
3. Kehilangan privasi.
4. Kehilangan integritas.
5. Kehilangan ketersediaan.

Skenario ini berfungsi sebagai contoh situasi yang mana organisasi mestinya bekerja untuk menurunkan risiko, atau kemungkinan menderita kerugian atau kerusakan. Dalam beberapa keadaan, wilayah-wilayah ini terhubung sangat erat sehingga tindakan yang merugikan satu wilayah dapat juga merugikan wilayah lainnya. Disamping itu, situasi seperti penipuan atau pelanggaran privasi dapat diakibatkan oleh tindakan berbahaya atau tidak merusak dan tidak selalu diikuti oleh perubahan yang dapat dilihat di database atau sistem komputer.

Ancaman, yang merupakan bagian integral dari diskusi keamanan database, tidak dapat dipisahkan. Ancaman (threat) adalah keadaan atau kejadian apa pun, berbahaya atau tidak.

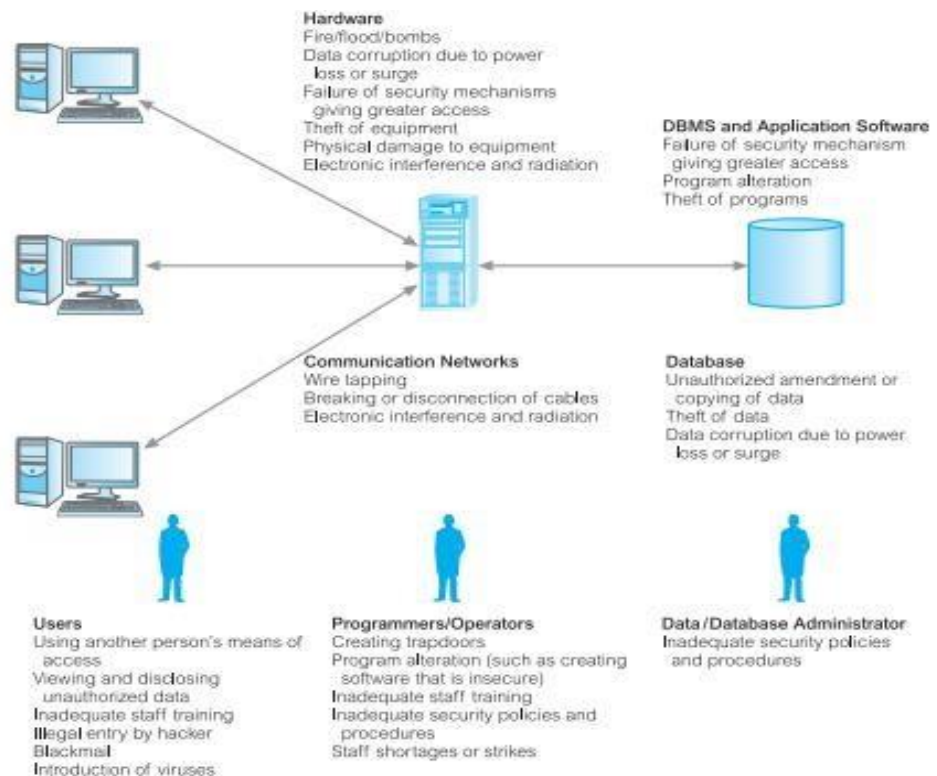
Setiap ancaman harus dianggap sebagai potensi pelanggaran keamanan yang akan berdampak jika berhasil. Tabel 10.1 berikut ini menyajikan contoh berbagai jenis *threat*, yang tercantum di bawah area di mana mereka mungkin memiliki dampak.

Tabel 10.1 Contoh Threat dan Area yang Terdampak

THREAT	Pencurian dan Penipuan	Hilangnya Kerahasiaan	Hilangnya Privasi	Hilangnya Integritas	Hilangnya Ketersediaan
Menggunakan sarana	✓	✓	✓		
Amandemen yang tidak sah atau menyalin data	✓			✓	
Perubahan program	✓			✓	✓
Kebijakan dan prosedur yang tidak memadai yang memungkinkan	✓	✓	✓		
campuran output rahasia dan normal					
Penyadapan	✓	✓	✓		
Entri ilegal oleh peretas	✓	✓	✓		
Pemerasan	✓	✓	✓		
Menciptakan "trapdoor"	✓	✓	✓		
Pencurian data, program	✓	✓	✓		✓
Kegagalan mekanisme keamanan, memberikan akses yang lebih besar dari		✓	✓	✓	
Kekurangan staf atau				✓	✓
memperoleh Pelatihan staf yang tidak memadai		✓	✓	✓	✓

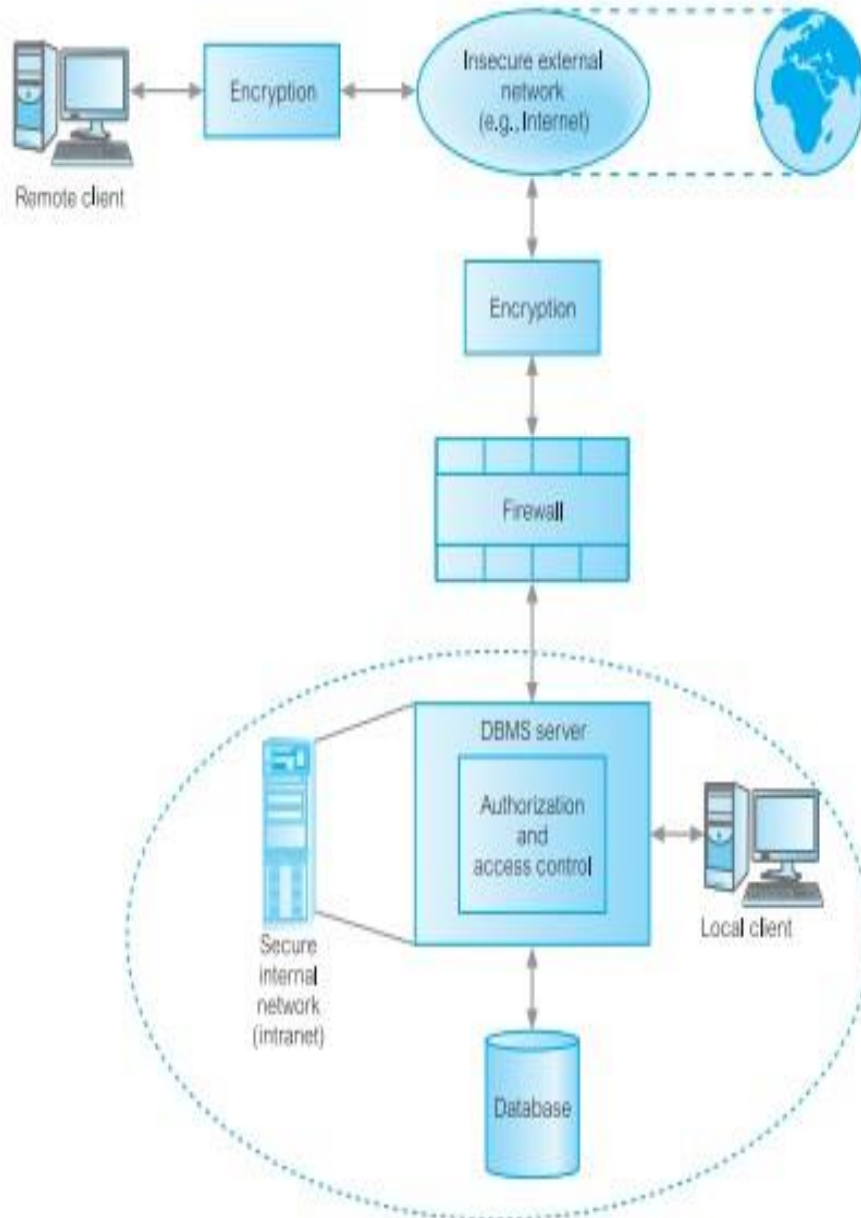
Melihat dan mengungkapkan data yang tidak sah	✓	✓	✓		
Gangguan elektronik dan radiasi				✓	✓
Korupsi data karena kehilangan daya atau lonjakan arus				✓	✓
Api (gangguan listrik, sambaran petir, kebakaran), banjir, bom				✓	✓
Kerusakan fisik pada peralatan				✓	✓

Organisasi perlu mengidentifikasi ancaman yang dapat mereka targetkan dan mulai mempersiapkan dan melindunginya dengan tepat dengan biaya tertentu. Bisnis organisasi juga dapat memengaruhi jenis ancaman yang harus dipertimbangkan, beberapa di antaranya mungkin jarang terjadi. Namun, kejadian langka harus diperhitungkan, terutama jika dampaknya akan signifikan. Ringkasan dari potensi ancaman terhadap sistem komputer diwakili dalam Gambar 10.1.



## 12.4 Countermeasures-Computer-Based Controls

Jenis tindakan balasan terhadap ancaman terhadap sistem komputer berkisar dari kontrol fisik hingga prosedur administratif. Dalam sub bagian ini kami fokus pada kontrol keamanan berbasis komputer untuk lingkungan multi-pengguna. Representasi lingkungan komputer multi-user yang khas ditunjukkan pada gambar 10.2. berikut ini



Di bagian ini kita akan fokus pada kontrol keamanan berbasis komputer untuk lingkungan multi-pengguna (beberapa diantara sangat mungkin tdk tersedia di lingkungan Porsenal Computer) berikut:

1. Otorisasi,

Otorisasi merujuk pada proses memberikan hak atau hak istimewa yang memungkinkan subjek yang memiliki akses sah ke suatu sistem atau objek sistem. Kontrol otorisasi dapat diimplementasikan dalam perangkat lunak dan mengatur tidak hanya sistem atau objek apa yang dapat diakses oleh pengguna tertentu, tetapi juga apa yang dapat dilakukan pengguna dengan objek tersebut. Proses otorisasi melibatkan otentikasi subjek yang meminta akses ke objek. Di sini, "subjek" dapat berupa pengguna atau program, dan "objek" dapat berupa tabel database, tampilan, prosedur, pemicu, atau objek lain yang dapat dibuat dalam sistem.

Administrator sistem biasanya bertanggung jawab untuk memungkinkan pengguna memiliki akses ke sistem komputer dengan membuat akun pengguna individual. Namun, hal ini tidak selalu berarti bahwa pengguna telah diberikan otorisasi untuk mengakses DBMS atau program aplikasi terkait. Sebagai gantinya, administrator sistem harus mengatur dan memastikan bahwa setiap pengguna diberikan otorisasi yang sesuai sesuai dengan kebutuhan bisnis dan prinsip keamanan yang berlaku. Dengan demikian, kontrol otorisasi yang tepat sangat penting untuk memastikan keamanan basis data dan mencegah akses yang tidak sah oleh pihak yang tidak berwenang.

2. Kontrol akses,

Salah satu metode umum yang digunakan untuk memberikan kontrol akses pada sistem basis data adalah melalui pemberian dan pencabutan hak istimewa. Dengan adanya hak istimewa ini, pengguna diberikan akses untuk membuat atau mengakses objek basis data tertentu seperti relasi, tampilan, atau indeks, dan juga untuk menjalankan utilitas DBMS tertentu. Hak istimewa ini diberikan kepada pengguna agar mereka dapat menyelesaikan tugas-tugas yang diperlukan untuk pekerjaan mereka. Pencabutan hak istimewa juga dapat dilakukan ketika pengguna tidak lagi membutuhkan akses ke objek atau utilitas tertentu.

Melihat kelemahan itu maka jelas diperlukan pendekatan keamanan tambahan untuk menghilangkan celah tersebut, dan persyaratan ini dipenuhi dalam pendekatan yang disebut *Mandatory Access Control (MAC)*.

Untuk menerapkan kebijakan MAC dalam DBMS relasional, kelas keamanan harus ditugaskan untuk setiap objek database. Situasi ini mengarah pada konsep hubungan multitingkat, yang merupakan relasi yang mengungkapkan tupel berbeda kepada pengguna dengan jarak keamanan yang berbeda. Masalah inferensi ini dapat diselesaikan dengan memasukkan atribut klasifikasi keamanan sebagai bagian dari kunci utama untuk suatu relasi. Kehadiran objek data yang tampaknya memiliki nilai berbeda untuk pengguna dengan jarak bebas yang berbeda pula disebut *polyinstantiation*.

Walaupun kontrol akses harus mengatasi kelemahan utama DAC, kelemahan utama MAC adalah kekakuan lingkungan MAC. Sebagai contoh, kebijakan MAC sering dibuat oleh administrator database atau sistem, dan mekanisme klasifikasi terkadang dianggap tidak fleksibel

3. View,

Tampilan atau "View" adalah hasil operasi dinamis dari satu atau lebih operasi relasional yang dijalankan pada relasi dasar untuk menghasilkan relasi baru. Tampilan tersebut adalah relasi virtual yang sebenarnya tidak tersimpan di dalam database, melainkan dibuat secara langsung saat diminta oleh pengguna tertentu.

4. Cadangan dan pemulihan,

Backup adalah proses menyalin basis data dan file log secara berkala (dan mem-backup program-program yang mungkin) ke media penyimpanan offline.

Journaling adalah proses menyimpan dan memelihara file log (atau jurnal) dari semua perubahan yang dilakukan ke database untuk memungkinkan pemulihan dilakukan secara efektif jika terjadi kegagalan.

5. Integritas,

Kendala integritas juga berkontribusi untuk menjaga sistem basis data yang aman dengan mencegah data menjadi tidak valid, dan karenanya memberikan hasil yang menyesatkan atau salah.

6. Enkripsi

Enkripsi adalah penyandian data melalui algoritme khusus yang membuat data terenkripsi tidak dapat dibaca oleh program apa pun tanpa kunci dekripsi.

Ada sejumlah teknik untuk menyandikan data untuk menyembunyikan informasi; beberapa disebut "irreversible" dan yang lain "reversibel." Teknik ireversibel, seperti namanya, tidak mengizinkan data asli untuk diketahui. Namun, data dapat digunakan untuk mendapatkan informasi statistik yang valid. Teknik reversibel lebih umum digunakan.

Untuk mentransmisikan data secara aman melalui jaringan yang tidak aman membutuhkan penggunaan cryptosystem, yang meliputi:

- a. Kunci enkripsi, untuk mengenkripsi data (*plaintext*);
- b. Algoritme enkripsi dengan menggunakan kunci enkripsi dapat mengubah *plaintext* menjadi *ciphertext*;
- c. Kunci dekripsi, digunakan untuk untuk mendekripsi ciphertext
- d. Algoritma dekripsi menggunakan kunci dekripsi mengubah ciphertext kembali menjadi plaintext.

Terdapat dua teknik enkripsi yaitu enkripsi simetris (*symmetric encryption*) dan enkripsi asimetris (*asymmetric encryption*). Enkripsi simetris, menggunakan kunci yang sama untuk enkripsi dan dekripsi dan bergantung pada jalur komunikasi yang aman untuk bertukar kunci, sedangkan enkripsi asimetris menggunakan kunci yang berbeda untuk enkripsi dan deskripsi.

7. Teknologi RAID

RAID (Redundant Array of Independent Disks) adalah teknologi penyimpanan data yang menggabungkan beberapa disk fisik menjadi satu sistem penyimpanan. Tujuan dari teknologi RAID adalah meningkatkan kinerja, keamanan, dan/atau kapasitas penyimpanan. Terdapat beberapa level RAID yang berbeda, yaitu:

RAID 0: disk digabungkan tanpa redundant data, sehingga memberikan kinerja tinggi dan kapasitas penyimpanan yang besar, namun jika satu disk rusak maka seluruh data hilang.

RAID 1: setiap data disimpan di dua disk yang berbeda (mirroring), sehingga memberikan tingkat keamanan yang tinggi, namun kapasitas penyimpanan hanya setengah dari total kapasitas disk yang digunakan.

RAID 5: data dibagi menjadi blok-blok dan diletakkan di disk yang berbeda, dan ditambahkan data parity untuk memungkinkan pemulihan data jika satu disk rusak. RAID 5 membutuhkan minimal tiga disk dan memberikan kinerja yang baik serta kapasitas penyimpanan yang efisien. RAID 6: mirip dengan RAID 5, namun data parity dihitung menggunakan dua bit parity sehingga memberikan keamanan yang lebih tinggi. RAID 6 membutuhkan minimal empat disk. RAID merupakan teknologi yang sangat umum digunakan pada server dan sistem penyimpanan data yang memerlukan kinerja tinggi, keamanan, dan/atau kapasitas penyimpanan yang besar.

## 12.5 DBMS and WEB Security

Komunikasi internet menggunakan TCP/IP untuk protokol dasarnya. Sedangkan, TCP/IP dan HTTP tdk dirancang dengan pertimbangan keamanan. Tanpa harus perangkat lunak khusus, semua lalu lintas Internet "asli" dan siapa pun yang menemukannya dapat membacanya. Bentuk serangan ini sangat mudah digunakan dengan perangkat lunak "packet sniffing" yang tersedia secara gratis, karena Internet selalu merupakan jaringan terbuka.

Oleh sebab itu tantangan yang ada adalah untuk mengirim dan menerima informasi melalui Internet sambil memastikan bahwa:

- 1) Informasi tidak dapat diakses oleh siapa pun kecuali transmittor dan receiver (privasi);
- 2) Informasi belum berubah selama pengiriman (integritas);
- 3) Penerima mendapatkan keyakinan itu berasal dari pengirim (keaslian);
- 4) Pengirim dapat memastikan bahwa penerima asli (nonfabrikasi);
- 5) Pengirim tidak dapat menyangkal dia mengirimnya (nonrepudiation).

Namun, melindungi transaksi hanya menyelesaikan sebagian dari masalah. Setelah informasi telah mencapai server Web, itu juga harus dilindungi di sana. Dengan arsitektur threethier yang populer di lingkungan Web, kita juga memiliki kompleksitas untuk memastikan akses aman ke, dan dari, database. Saat ini, sebagian besar arsitektur semacam itu dapat diamankan, tetapi umumnya membutuhkan produk dan mekanisme yang berbeda.

Salah Satu aspek keamanan lainnya yang harus ditangani di lingkungan Web merupakan yang menyatakan bahwa informasi yang dikirimkan ke mesin klien mungkin memiliki konten yang dapat dieksekusi (*executable content*). Konten yang dapat dieksekusi dapat melakukan tindakan jahat berikut, dan tindakan berikut harus diambil untuk mencegahnya melakukan:

- 1) MengKorup data atau status pelaksanaan program;
- 2) Memformat ulang disk lengkap;
- 3) Melakukan shutdown sistem total;
- 4) Mengumpulkan dan unduh data rahasia, seperti file atau kata sandi, ke situs lain;
- 5) Mengubah identitas dan tiru pengguna atau komputer pengguna untuk menyerang target lain di jaringan;
- 6) Mengunci sumber daya sehingga tidak tersedia untuk pengguna dan program yang sah;
- 7) Menyebabkan efek yang tidak fatal tetapi tidak diinginkan, terutama pada perangkat output.

Di sesi ini kita akan membahas beberapa masalah yang berkaitan dengan keamanan basis data yang berada dalam lingkungan berikut ini.

### 1. Proxy Server

Di lingkungan Web, server proxy adalah komputer yang berada di antara browser Web dan server Web. Server proxy memiliki dua tujuan utama yaitu :

a. Meningkatkan kinerja

Karena server proksi menyimpan hasil dari semua permintaan untuk waktu tertentu, itu dapat secara signifikan meningkatkan kinerja untuk kelompok pengguna. Selain itu karena server proxy sering berada di jaringan yang sama dengan pengguna, maka operasi dapat dilakukan dengan jauh lebih cepat. Server proxy nyata, seperti yang digunakan oleh CompuServe dan America Online, dapat support ribuan user.

b. Memfilter *request*

Server proxy juga dapat digunakan untuk memfilter permintaan. Sebagai contoh, sebuah organisasi mungkin menggunakan server proxy untuk mencegah karyawannya mengakses satu set situs Web tertentu.

## 2. Firewalls

Firewall adalah sebuah sistem keamanan jaringan yang bertindak sebagai penghalang antara jaringan internal (misalnya jaringan komputer kantor) dan jaringan eksternal (misalnya Internet). Tujuan dari firewall adalah untuk melindungi jaringan internal dari serangan yang berasal dari luar jaringan.

Firewall bekerja dengan memeriksa setiap paket data yang melewati batas jaringan, kemudian memutuskan apakah paket tersebut diteruskan atau diblokir. Firewall dapat melakukan ini dengan menggunakan beberapa metode, seperti memeriksa alamat IP dan port sumber dan tujuan, atau memeriksa isi dari paket data itu sendiri.

Firewall dapat dikonfigurasi dengan berbagai cara, termasuk whitelist (daftar akses yang diizinkan) dan blacklist (daftar akses yang diblokir). Firewall juga dapat ditempatkan pada berbagai titik dalam jaringan, termasuk di server, router, dan gateway.

Penggunaan firewall telah menjadi sangat umum dalam sistem keamanan jaringan modern, baik dalam lingkungan perusahaan maupun individual. Hal ini karena firewall dapat membantu mencegah serangan jaringan, mengurangi risiko kerentanan keamanan dan melindungi data sensitif yang disimpan di jaringan internal.

## 3. *Message Digest Algorithms* dan *Digital Signatures* (Tandatangan Digital)

Message Digest Algorithms (MDA) adalah teknik kriptografi yang digunakan untuk menghasilkan nilai hash atau fingerprint unik dari data yang diberikan. Nilai hash ini kemudian digunakan untuk memverifikasi integritas data, yaitu memastikan bahwa data tersebut tidak diubah atau dimanipulasi dalam perjalanan.

Digital Signature (Tandatangan Digital) adalah metode yang digunakan untuk memberikan keamanan pada dokumen digital dengan menggunakan teknik kriptografi. Tandatangan digital mirip dengan tandatangan fisik, namun dibuat menggunakan teknologi kriptografi, yang membuatnya jauh lebih sulit untuk dipalsukan. Tandatangan digital digunakan untuk memverifikasi keaslian dokumen digital, melindungi dokumen dari perubahan yang tidak sah dan memastikan bahwa hanya pihak yang ditunjuk yang dapat mengakses dan memodifikasi dokumen tersebut.

MDA dan digital signature biasanya digunakan bersama-sama untuk memberikan keamanan tambahan pada dokumen digital. Sebelum dokumen ditandatangani digital, nilai hash dari dokumen tersebut dihasilkan menggunakan algoritma MDA. Nilai hash ini kemudian dienkripsi dengan menggunakan kunci pribadi pengirim untuk menghasilkan digital signature. Penerima kemudian dapat memverifikasi digital signature dengan menggunakan kunci publik

pengirim, dan membandingkan nilai hash dari dokumen yang diterima dengan nilai hash yang ditandatangani digital oleh pengirim. Jika kedua nilai hash sama, maka dokumen dianggap tidak diubah dan asli.

#### **4. Digital Certificates (Sertifikat Digital)**

Sertifikat digital adalah dokumen elektronik yang mengonfirmasikan identitas dari entitas tertentu, seperti individu, organisasi, atau situs web. Sertifikat digital umumnya berisi informasi yang mengidentifikasi pemiliknya, termasuk nama, alamat, nomor identitas, dan informasi lain yang relevan.

Sertifikat digital biasanya digunakan untuk menjamin keamanan komunikasi elektronik dengan memastikan bahwa entitas yang berkomunikasi adalah yang sebenarnya dan mencegah orang yang tidak berwenang mengakses informasi atau merubah data. Sertifikat digital juga dapat digunakan untuk mengamankan transaksi online, seperti pembelian online atau transfer dana.

Sertifikat digital dikeluarkan oleh otoritas sertifikasi (CA), yang adalah lembaga tepercaya yang bertanggung jawab untuk memverifikasi identitas entitas yang meminta sertifikat dan memastikan bahwa informasi yang disediakan akurat. Ketika sertifikat digital diterbitkan, informasi tentang sertifikat, termasuk informasi identitas dan kunci publik, dimasukkan ke dalam database CA. Sertifikat ini kemudian dapat digunakan oleh entitas yang membutuhkan keamanan komunikasi, seperti server web atau klien email, untuk mengotentikasi identitas dan mengenkripsi komunikasi.

#### **5. Kerberos**

Kerberos adalah protokol autentikasi jaringan yang digunakan untuk memberikan keamanan terhadap serangan seperti sniffing dan replay attack. Protokol ini digunakan untuk mengotentikasi pengguna dan memberikan izin akses ke sumber daya jaringan seperti file, printer, atau aplikasi.

Kerberos bekerja dengan cara memberikan tiket kepada pengguna setelah mereka memasukkan kredensial yang benar. Tiket ini kemudian digunakan untuk memperoleh izin akses ke sumber daya jaringan yang diminta oleh pengguna. Tiket ini memiliki masa berlaku tertentu dan hanya dapat digunakan oleh pengguna yang sah.

Kerberos juga menggunakan server yang disebut Key Distribution Center (KDC) untuk mengelola tiket dan otentikasi. KDC memiliki dua komponen utama: Authentication Server (AS) dan Ticket Granting Server (TGS). AS digunakan untuk mengotentikasi pengguna dan memberikan tiket awal, sedangkan TGS digunakan untuk menghasilkan tiket kerberos baru setelah pengguna memasukkan tiket awal.

Kerberos digunakan di banyak sistem operasi seperti Windows, Linux, dan macOS dan merupakan salah satu protokol autentikasi jaringan yang paling umum digunakan.

#### **6. Secure Sockets Layer dan Secure HTTP**

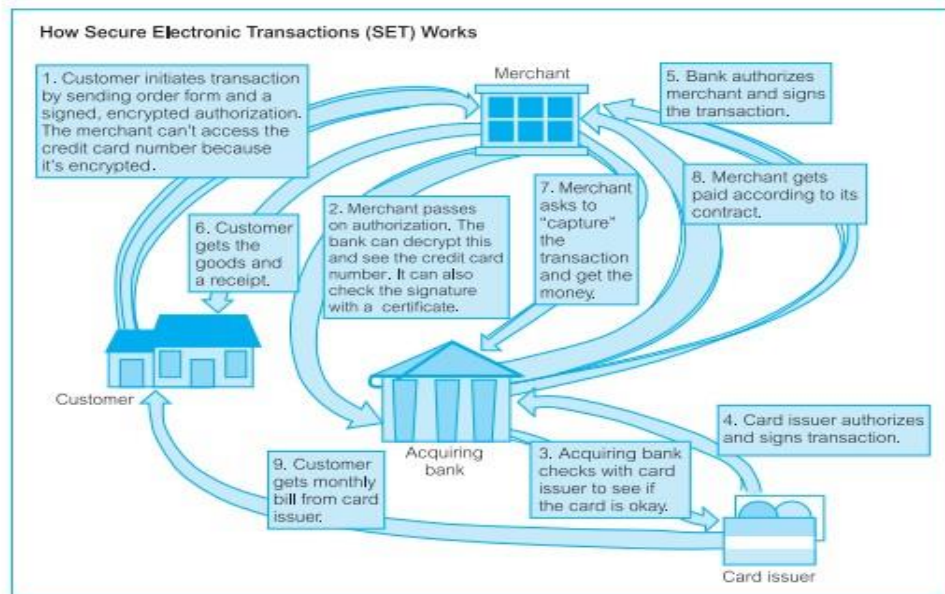
Secure Sockets Layer (SSL) dan Secure HTTP (HTTPS) adalah protokol keamanan untuk memastikan keamanan komunikasi antara klien dan server di internet. SSL dan HTTPS mengenkripsi data yang ditransfer antara server web dan browser klien menggunakan teknologi enkripsi kunci publik dan kunci simetris.

SSL memastikan bahwa data yang ditransfer antara klien dan server dienkripsi dan tidak dapat dibaca oleh pihak ketiga yang mencoba mengintersep komunikasi. SSL juga menggunakan tanda tangan digital untuk memverifikasi identitas server web, sehingga klien

dapat yakin bahwa mereka terhubung ke server yang sebenarnya dan bukan server palsu yang mencoba menipu mereka.

HTTPS, singkatan dari Hypertext Transfer Protocol Secure, adalah protokol yang digunakan untuk memastikan keamanan komunikasi di web melalui SSL. Ketika browser klien terhubung ke server web menggunakan HTTPS, browser klien dan server web akan saling bertukar sertifikat digital dan melakukan negosiasi kunci enkripsi yang akan digunakan selama sesi. Setelah koneksi aman didirikan, browser klien dan server web akan berkomunikasi dengan menggunakan enkripsi kunci simetris yang digunakan untuk mengenkripsi dan mendekripsi data yang ditransfer.

## 7. Keamanan Electronic Transaction and Transaction Keamanan Technology



Meskipun Microsoft dan Visa International adalah peserta utama dalam spesifikasi SET, mereka saat ini menyediakan protokol Secure Transaction Technology (STT), yang telah dirancang untuk menangani pembayaran bank yang aman melalui Internet. STT menggunakan enkripsi informasi DES, enkripsi RSA informasi kartu kredit, dan otentikasi yang kuat dari semua pihak yang terlibat dalam transaksi.

## 8. Java Security

Sand box adalah bagian integral dari desain Java untuk memastikan keselamatan dan keamanan, dengan cara mencegah aplikasi yang tidak dipercayai dan berpotensi berbahaya untuk tidak dapat mengakses sumber daya sistem. Ada tiga komponen yang digunakan untuk menerapkan "sandbox" ini:

### a. *a class loader*,

Kelas loader, selain memuat setiap kelas yang diperlukan dan memeriksa itu dalam format yang benar, juga memeriksa apakah aplikasi / applet melanggar keamanan sistem dengan mengalokasikan *namespace*.

### b. *a bytecode verifier*/ pemverifikasi bytecode,

Pemeriksaan umum termasuk memverifikasi bahwa:

- kode yang dikompilasi diformat dengan benar;
- tumpukan internal tidak akan meluap / melimpah;
- tidak ada konversi data "ilegal" yang akan terjadi (misalnya, integer ke pointer) - ini memastikan bahwa variabel tidak akan diberikan akses ke area memori yang terbatas;
- instruksi bytecode diketik secara tepat;
- semua akses anggota kelas valid.

### c. *a security manager*/ seorang manajer keamanan.

Umumnya, Security Manager melakukan verifikasi runtime metode yang berpotensi "berbahaya", yaitu metode yang meminta I / O, akses jaringan, atau upaya untuk mendefinisikan loader kelas baru.

Fitur keamanan disediakan oleh bahasa Java dan Java Virtual Machine (JVM), dan ditegakkan oleh compiler dan sistem runtime; keamanan adalah kebijakan yang dibangun di atas lapisan keamanan ini.

Dua fitur keamanan Java berhubungan dengan pengetikan yang kuat dan pengumpulan sampah otomatis

## 9. ActiveX Security

ActiveX adalah teknologi yang dikembangkan oleh Microsoft untuk memungkinkan komponen software dapat digunakan kembali (reusable) pada platform Windows. Meskipun ActiveX menyediakan berbagai keuntungan dalam pengembangan perangkat lunak, seperti komponen yang dapat digunakan kembali dan fungsionalitas yang diperluas, ia juga dapat membuka celah keamanan dalam sistem.

Untuk mengatasi masalah keamanan dalam ActiveX, Microsoft mengembangkan teknologi ActiveX Security yang memungkinkan administrator sistem untuk mengatur perizinan untuk mengakses dan menggunakan komponen ActiveX. Pengguna juga dapat mengatur tingkat keamanan ActiveX melalui Internet Explorer untuk memblokir atau mengizinkan instalasi ActiveX pada komputer mereka. Selain itu, ActiveX digunakan dalam lingkungan sandbox untuk membatasi akses terhadap sumber daya sistem yang lebih sensitif, seperti registry dan sistem file.

## 12.6 Data Administrasi and Database Administrasi

Administrasi Data (DA) dan Database Administrator (DBA) bertanggung jawab untuk mengelola dan mengendalikan kegiatan yang terkait dengan data perusahaan dan database perusahaan, masing-masing.

Pengertian dari *data administration* (DA/ Administrasi data) adalah manajemen sumber data, yang meliputi perencanaan, pengembangan, dan pemeliharaan standar, kebijakan dan prosedur database, dan desain database konseptual dan logis. Tugas yang terkait dengan administrasi data dijelaskan pada Tabel 10.3. berikut ini.

Tabel 10.3. Tugas yang Berkaitan dengan DA

Memilih alat produktivitas yang tepat.
Membantu dalam pengembangan perusahaan IT / IS dan strategi perusahaan.
Melakukan studi kelayakan dan perencanaan untuk pengembangan basis data.
Mengembangkan model data perusahaan.
Menentukan persyaratan data organisasi.
Menetapkan standar pengumpulan data dan menetapkan format data.
Memperkirakan volume data dan kemungkinan pertumbuhan.
Menentukan pola dan frekuensi penggunaan data.
Menentukan persyaratan akses data dan perlindungan untuk persyaratan hukum dan perusahaan.
Melaksanakan desain database konseptual dan logis.
Bekerja sama dengan staf administrasi database dan pengembang aplikasi untuk memastikan aplikasi
memenuhi semua persyaratan yang disebutkan.
Mendidik pengguna tentang standar data dan tanggung jawab hukum.
Tetap up to date dengan IT / IS dan perkembangan perusahaan.
Memastikan dokumentasi mutakhir dan lengkap, termasuk standar, kebijakan, prosedur, penggunaan kamus data, dan kontrol pada pengguna akhir.
Mengelola kamus data.

Sedangkan Administrasi basis data adalah manajemen realisasi fisik dari sistem basis data, yang mencakup perancangan dan penerapan basis data fisik, pengaturan kontrol keamanan dan integritas, pemantauan kinerja sistem, dan pengaturan ulang basis data, jika diperlukan. Tugas administrasi basis data dijelaskan pada Tabel 10.4. berikut ini.

Tabel 10.4. Tugas yang Berkaitan dengan DBA

Mengevaluasi dan memilih produk DBMS.
Melakukan perancangan basis data fisik.

Menerapkan desain database fisik menggunakan DBMS target.
Menentukan batasan keamanan dan integritas.
Berhubungan dengan pengembang aplikasi database.
<i>Mengembangkan strategi tes.</i>
Pelatihan pengguna.
Bertanggung jawab untuk "sign off" pada sistem database yang diimplementasikan.
Memantau kinerja sistem dan menyetel basis data sebagaimana mestinya.
Melakukan backup secara rutin.
Memastikan bahwa mekanisme dan prosedur pemulihan sudah ada.
Memastikan dokumentasi itu lengkap, termasuk material yang diproduksi di rumah.
Tetap up to date dengan berkembangnya software dan perangkat keras serta biaya, dan instal terbaru pembaruan seperlunya.

Setelah memeriksa tujuan dan tugas yang terkait dengan administrasi data dan administrasi basis data, maka pada bagian akhir ini kita akan secara singkat membandingkan area-area fungsional yang ada. Tabel 10.5 merangkum perbedaan tugas utama administrasi data dan administrasi basis data.

Tabel 10.5 Administrasi data dan administrasi basis data — perbedaan tugas utama.

<b>Data Administration</b>	<b>Database Administration</b>
Terlibat dalam perencanaan strategis IS	Mengevaluasi DBMS baru
Menentukan tujuan jangka panjang	Melaksanakan rencana untuk mencapai tujuan
Menegakkan standar, kebijakan, dan prosedur	Menegakkan standar, kebijakan, dan prosedur
Menentukan persyaratan data	Mengimplementasikan persyaratan data
Mengembangkan desain database konseptual dan logis	Mengembangkan desain database logis dan fisik
Mengembangkan dan memelihara model data perusahaan	Mengimplementasikan desain database fisik
Mengkoordinasikan pengembangan sistem	Mengawasi dan mengontrol basis data
Orientasi manajerial	Orientasi teknis
DBMS-independen	DBMS-dependen

## 12.7 Web Technologies and DBMS

### 1. REQUIREMENTS FOR WEB–DBMS INTEGRATION

Meskipun banyak vendor DBMS bekerja untuk menyediakan solusi konektivitas database berpemilik untuk Web, sebagian besar organisasi memerlukan solusi yang lebih umum untuk mencegah mereka dari terikat ke dalam satu teknologi. Di bagian ini, kami secara singkat mencantumkan beberapa persyaratan paling penting untuk integrasi aplikasi basis data dengan Web. Persyaratan ini adalah cita-cita dan tidak sepenuhnya dapat dicapai pada saat ini, dan beberapa mungkin perlu ditukar dengan yang lain. Tidak dalam urutan peringkat apa pun, persyaratannya adalah sebagai berikut:

1. Solusi konektivitas yang memanfaatkan semua fitur DBMS organisasi.
2. Pendekatan arsitektur terbuka untuk memungkinkan interoperabilitas dengan berbagai sistem dan teknologi.
3. Solusi hemat biaya
4. Dukungan untuk transaksi yang menjangkau beberapa permintaan HTTP.
5. Dukungan untuk otentikasi berbasis sesi dan aplikasi.
6. Kinerja yang dapat diterima.
7. Biaya administrasi minimal.
8. Seperangkat alat produktivitas tingkat tinggi untuk memungkinkan aplikasi dikembangkan, dipelihara, dan digunakan dengan relatif mudah dan cepat.

### 2. Kelebihan dan Kekurangan dari THE WEB–DBMS APPROACH

Web sebagai platform untuk sistem basis data dapat memberikan solusi inovatif untuk operasi bisnis antar dan antar perusahaan. Sayangnya, ada juga kelemahan yang terkait dengan pendekatan ini. Keuntungan dari pendekatan Web-DBMS adalah Kesederhanaan, Kemandirian platform, Antarmuka pengguna grafis, Standardisasi, Dukungan lintas platform, Akses jaringan transparan, Penempatan scalable, dan Inovasi.

Kerugian dari pendekatan Web-DBMS adalah kurangnya Keandalan, Keamanan, Biaya, Skalabilitas, Fungsionalitas HTML yang terbatas, *statelessness*, Bandwidth, Performa, Ketidakmatangan alat pengembangan.

### 3. Introduction to XML

Sebagian besar dokumen di Web saat ini disimpan dan dikirim dalam HTML. Kami telah berkomentar bahwa salah satu kekuatan HTML adalah kesederhanaannya, memungkinkannya untuk digunakan oleh berbagai macam pengguna. Namun, kesederhanaannya bisa dibalang juga salah satu kelemahannya, dengan meningkatnya kebutuhan dari pengguna yang menginginkan tag untuk menyederhanakan beberapa tugas dan membuat dokumen HTML lebih menarik dan dinamis. Dalam upaya untuk memenuhi permintaan ini, vendor telah memperkenalkan beberapa tag HTML khusus browser. Namun, ini menyulitkan untuk mengembangkan dokumen Web yang canggih dan dapat dilihat secara luas. Untuk mencegah perpecahan ini, W3C menghasilkan standar yang disebut eXtensible Markup Language (XML), yang dapat mempertahankan kemandirian aplikasi umum yang menjadikan HTML portabel dan kuat. XML 1.0 (Edisi

Kedua) menjadi Rekomendasi W3C pada Oktober 2000 (W3C, 2000b) dan XML 1.1 (Edisi Kedua), dengan dukungan Unicode 3, menjadi Rekomendasi W3C pada Agustus 2006 (W3C, 2006a). XML 1.0 (Edisi Kelima) menjadi Rekomendasi W3C pada bulan November 2008 dan cenderung menjadi versi yang disarankan kecuali diperlukan fitur khusus XML 1.1.

## **12.8** Beberapa contoh aplikasi yang menggunakan database :

### **1. Mengambil asuransi**

Kapan pun Anda ingin mengambil asuransi — misalnya asuransi pribadi, asuransi properti, atau asuransi mobil.

### **2. Menyewa DVD**

Ketika Anda ingin menyewa DVD dari perusahaan rental DVD, Anda mungkin akan menemukan bahwa perusahaan menyimpan database yang terdiri dari judul DVD yang disewakan, perincian pada salinannya untuk setiap judul, apakah salinannya tersedia untuk disewa atau saat ini sedang dipinjamkan, perincian para anggotanya (para penyewa), dan DVD mana yang saat ini mereka sewa dan tanggal mereka kembalikan. Basis data bahkan dapat menyimpan informasi yang lebih rinci tentang setiap DVD, seperti sutradara dan para aktornya.

### **3. Menggunakan Internet**

Toko buku memungkinkan Anda menelusuri buku dalam berbagai kategori, seperti komputasi atau manajemen, atau dengan nama penulis. Dalam kasus apa pun, ada basis data di server Web organisasi yang terdiri atas detail buku, ketersediaan, informasi pengiriman, tingkat stok, dan riwayat pesanan. Detail buku mencakup judul buku, ISBN, penulis, harga, riwayat penjualan, penerbit, ulasan, dan deskripsi terperinci. Basis data memungkinkan buku-buku menjadi referensi silang: misalnya, sebuah buku dapat terdaftar di bawah beberapa kategori, seperti komputasi, bahasa pemrograman, buku terlaris, dan judul yang direkomendasikan. Referensi silang juga memungkinkan Amazon memberi Anda informasi tentang buku lain yang biasanya dipesan bersama dengan judul yang Anda minati. Seperti contoh sebelumnya, Anda dapat memberikan detail kartu kredit untuk membeli satu atau lebih buku secara online. Amazon.com mempersonalisasikan layanannya bagi pelanggan yang kembali ke situsnya dengan mencatat semua transaksi sebelumnya, termasuk barang yang dibeli, pengiriman, dan rincian kartu kredit. Ketika Anda kembali ke situs, Anda mungkin akan disambut oleh nama dan disajikan dengan daftar judul yang direkomendasikan berdasarkan pembelian sebelumnya.

## SIMPULAN

Kesimpulan dari materi ini adalah:

1. Keamanan basisdata merupakan mekanisme yang sangat melindungi data base terhadap ancaman yg disengaja atau tidak disengaja.
2. Threat (ancaman) adalah keadaan atau peristiwa apa saja, apakah disengaja atau tdk disengaja, yang mempunyai dampak negatif di sistem dan pengaruh organisasi.
3. Kontrol keamanan berbasis komputer untuk lingkungan banyak pengguna meliputi: otorisasi, kontrol akses, tampilan, cadangan dan pemulihan, integritas, enkripsi, dan teknologi RAID.
4. Oracle menyediakan dua jenis keamanan yaitu keamanan sistem dan keamanan data.
5. Langkah-langkah keamanan yang terkait dengan DbMSs di Web.
6. Algorithman dan digital signatures serta sejenisnya.
7. Tingkat konseptual mewakili semua entitas, atribut mereka, dan hubungan mereka, serta batasan pada data, dan informasi keamanan dan integritas. tingkat internal adalah tampilan komputer dari database: ia menentukan bagaimana data direpresentasikan, bagaimana catatan diurutkan, indeks dan pointer apa yang ada, dan sebagainya.
8. Model data adalah kumpulan konsep yang dapat digunakan untuk mendeskripsikan sekumpulan data, operasi untuk memanipulasi data, dan satu set batasan integritas untuk data. Model data dibagi ke dalam tiga kategori besar: model data berbasis objek, model data berbasis-rekaman, dan model data fisik. Dua yang pertama digunakan untuk menggambarkan data pada tingkat konseptual dan eksternal; yang terakhir digunakan untuk menggambarkan data pada tingkat internal.
9. Pemodelan konseptual adalah proses membangun arsitektur terperinci untuk database yang tidak bergantung pada detail implementasi, seperti target DBMS, program aplikasi, bahasa pemrograman, atau pertimbangan fisik lainnya

## BAB XII KONSEP DATA WAREHOUSE DAN SCHEMA

### 12.1 Tujuan Pembelajaran :

- Mahasiswa diharapkan mampu mendeskripsikan konsep data warehouse dalam organisasi bisnis

### 12.2 Dasar Teori

Definisi Data Warehouse, Berikut ini adalah define dari data warehouse :

#### 1. Berorientasi subyek

Data warehouse dibangun berdasarkan entitas data utama atau subjek organisasi, seperti siswa, guru, program studi di Universitas, atau pekerja, magang, jasa, dan barang di perusahaan.

#### 2. Terintegrasi

Data warehouse menggabungkan data dari berbagai sistem untuk memberikan pandangan yang lebih luas terhadap data perusahaan yang ada, sehingga disebut 'terintegrasi'.

#### 3. Time variant

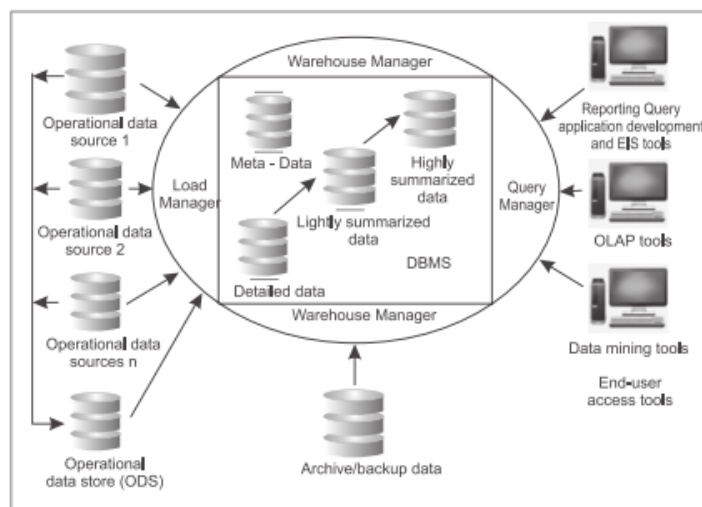
Data dalam data warehouse tidak selalu terupdate karena berisi data historis yang valid atau akurat sampai pada waktu tertentu atau interval waktu.

#### 4. Non-volatile

Data warehouse merupakan memori perusahaan jangka panjang karena sifatnya yang tidak mudah menguap. Data tidak diperbarui secara real-time, tetapi lebih sering diperbarui secara berkala. Oleh karena itu, data baru ditambahkan ke dalam database seperti suplemen, bukan pengganti, dan secara bertahap diintegrasikan dengan data sebelumnya. Arsitektur Data Warehouse

### 12.3 Arsitektur data warehouse

Berikut ini adalah gambar arsitektur data warehouse :



Gambar 12.1. Arsitektur dari data warehouse

### 1. Load manager

Load manager bertanggung jawab atas pengumpulan data dari sistem operasional. Dia juga melakukan konversi data menjadi beberapa bentuk yang dapat digunakan lebih lanjut oleh pengguna. Ini mencakup semua program dan antarmuka aplikasi yang diperlukan untuk mengekstraksi data dari sistem operasional, persiapannya, dan akhirnya memuat data ke dalam data warehouse itu sendiri. Load manager harus melakukan tugas-tugas berikut ini:

- a. Identifikasi Data
- b. Validasi Data untuk akurasi
- c. Ekstraksi Data dari sumber aslinya
- d. Pembersihan data
- e. Pemformatan data
- f. Standarisasi data (yaitu membawa data agar sesuai dengan beberapa format standar)
- g. Mengonsolidasikan data dari berbagai sumber ke satu tempat
- h. Pembentukan Integritas Data menggunakan Integrity constraint

### 2. Warehouse manager

Warehouse Manager adalah bagian utama dari sistem Data Warehousing karena menyimpan sejumlah besar informasi dari berbagai sumber. Bagian ini mengatur data sedemikian rupa sehingga menjadi mudah bagi siapa saja untuk menganalisis atau menemukan informasi yang diperlukan. Warehouse manager adalah inti dari data warehouse itu sendiri. Dia mempertahankan tiga tingkat informasi, yaitu, details, lightly summarized dan highly summarized. Dia juga bertugas memelihara data meta, yaitu data about data.

### 3. Query manager

Akhirnya Query manager adalah antarmuka yang menghubungkan pengguna akhir dengan informasi yang disimpan di data warehouse melalui penggunaan alat pengguna akhir khusus. Alat-alat ini dikenal sebagai alat akses data mining. Pasar saat ini dibanjiri dengan alat yang memiliki fungsi umum dan ketentuan untuk menyesuaikan lebih banyak fitur khusus untuk suatu perusahaan. Alat-alat ini memiliki berbagai kategori seperti kueri dan pelaporan, statistik, data discovery, dll.

## 12.4 Keuntungan Data Warehouse

Sebuah data warehouse dapat melakukan keajaiban jika diimplementasikan dengan sukses; beberapa keunggulan dan manfaat yang dapat diberikan dapat dilihat di bawah ini.

### 1. Potensi ROI tinggi (Pengembalian Investasi)

Berinvestasi di Data Warehousing itu sendiri merupakan investasi yang sangat besar, tetapi laporan sebelumnya menunjukkan pertumbuhan ROI hingga 400% pada organisasi yang menggunakan Data Warehousing, hal ini menjadikannya pilihan berharga untuk bisnis.

### 2. Keunggulan kompetitif yang tak terkalahkan

Implementasi Data Warehousing dapat memberikan keunggulan kompetitif bagi perusahaan atas para pesaingnya. Dengan Data Warehousing, perusahaan dapat menemukan fakta dan angka yang sebelumnya tidak tersedia, tren, dan informasi yang

belum dimanfaatkan. Pengungkapan baru semacam itu akan meningkatkan kualitas keputusan.

3. Produktivitas Tinggi dalam pengambilan keputusan perusahaan dan business intelligence. Data Warehousing menggabungkan data dari berbagai sumber menjadi informasi yang berarti yang dapat dianalisis dan dirujuk oleh manajer untuk meningkatkan keputusan mereka bagi organisasi.
4. Hemat biaya  
Dengan Data Warehousing memungkinkan untuk merampingkan organisasi sehingga mengurangi biaya overhead dan mengurangi biaya produk.
5. Layanan pelanggan yang ditingkatkan  
Data Warehousing memberikan dukungan penting saat berkomunikasi dengan pelanggan dan dengan demikian membantu meningkatkan kepuasan pelanggan dan mempertahankannya.

#### 12.5 Masalah dan Keterbatasan Data Ware House

Terkait dengan pengembangan dan pengelolaan data warehousing ada beberapa masalah yang harus di hadapi adalah sebagai berikut :

1. Meremehkan sumber daya untuk data ETL  
Cukup sering pengguna meremehkan total waktu pemrosesan yang diperlukan untuk melakukan operasi ekstraksi, pembersihan, dan pemuatan data sebelum melakukan data warehousing. Jadi, dalam implementasi real time banyak proses lain yang terganggu dan operasi dapat terganggu sementara hal itu terjadi.
2. Sistem sumber yang salah  
Ada banyak masalah tersembunyi yang terkait dengan sistem sumber yang digunakan untuk memasukkan data.
3. Data yang diperlukan tidak diambil  
Data warehouse menyimpan informasi rinci, namun sering kali dengan sengaja melewati beberapa detail kecil yang nantinya dapat digunakan untuk analisis atau tugas lainnya. Misalnya, tanggal pendaftaran saat menambahkan detail properti baru mungkin tidak digunakan dalam sistem sumber tetapi mungkin sangat berguna selama analisis.
4. Meningkatnya permintaan pengguna akhir dan demands  
Dari sisi pengguna akhir, kueri tidak pernah berakhir. Bahkan setelah pertanyaan awal telah ditangani dengan memuaskan, rantai pertanyaan lebih lanjut dapat mengikuti.
5. Hilangnya informasi selama homogenisasi data  
Ketika data dari berbagai sumber digabungkan, kehilangan informasi dapat terjadi saat mengubah format.
6. Permintaan sumber daya yang tinggi  
Sumber daya seperti ruang disk yang besar diperlukan untuk menyimpan sejumlah besar data yang terakumulasi setiap hari.
7. Kepemilikan data  
Cukup sering ketika satu departemen dimintai beberapa data, departemen tersebut enggan atau lambat dalam memproses permintaan tersebut. Mereka mungkin takut kehilangan kepemilikan data atau kendali atas data. Hal ini menyebabkan kelambatan dan gangguan dalam proses Data Warehousing.

8. Perawatan yang diperlukan

Sebuah data warehouse membutuhkan perawatan yang tinggi. Setiap perubahan dalam proses bisnis atau reorganisasi sistem sumber dapat mempengaruhi data warehouse dan semua hal ini berkontribusi pada biaya pemeliharaan yang sangat tinggi.

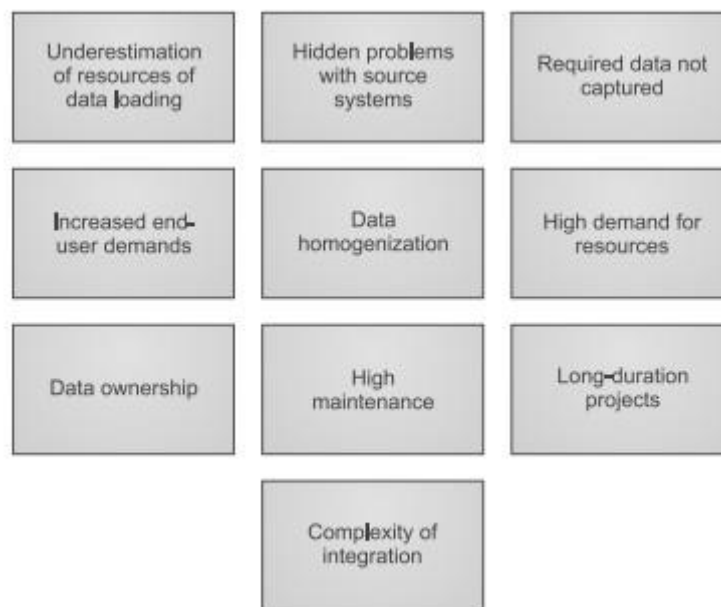
9. Proyek berdurasi Panjang

Mulai dari perencanaan hingga implementasi aktual, data warehouse menghabiskan banyak waktu dan uang; maka banyak organisasi yang pada awalnya enggan.

10. Kompleksitas integrasi

Kinerja data warehouse dapat ditentukan berdasarkan kemampuannya integrasinya. Dengan demikian, sebuah organisasi menghabiskan banyak waktu secara signifikan, dalam menentukan seberapa baik beberapa alat data warehousing dapat berjalan bersama (atau terintegrasi) untuk menghasilkan solusi, yang merupakan tugas yang sangat sulit karena banyak alat yang tersedia.

Isu-isu data warehouse diringkas dalam Gambar 12.2.berikut.



Gambar 12.2. Limitasi Data Warehouse

## 12.6 Data Marts

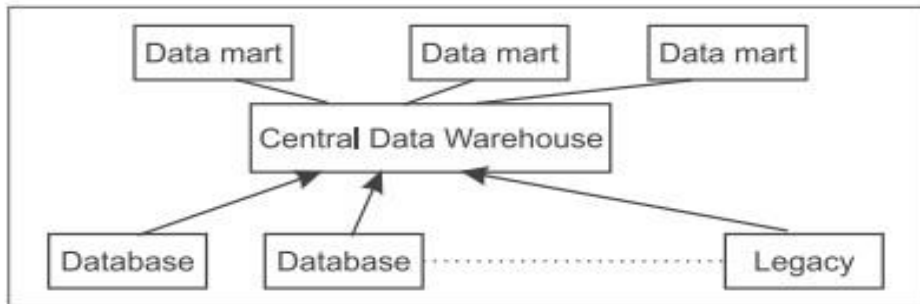
Data warehouse khusus departemen disebut sebagai 'data mart', yang merupakan data warehouse lokal kecil yang dibangun untuk satu tujuan. Biasanya dibangun untuk memenuhi kebutuhan sekelompok pengguna atau departemen dalam suatu organisasi.

Misalnya, sebuah organisasi dapat memiliki banyak departemen, termasuk keuangan, departemen TI, dan lainnya. Masing-masing departemen ini dapat memiliki gudang data mereka sendiri, yang tidak lain adalah data mart dari departemen tertentu.

Dengan demikian, data mart dapat didefinisikan sebagai 'penyimpanan data yang terspesialisasi, berorientasi subjek, terintegrasi, time-variant, dan volatile untuk mendukung subset tertentu dari keputusan manajemen'. Dengan cara yang disederhanakan, kita dapat

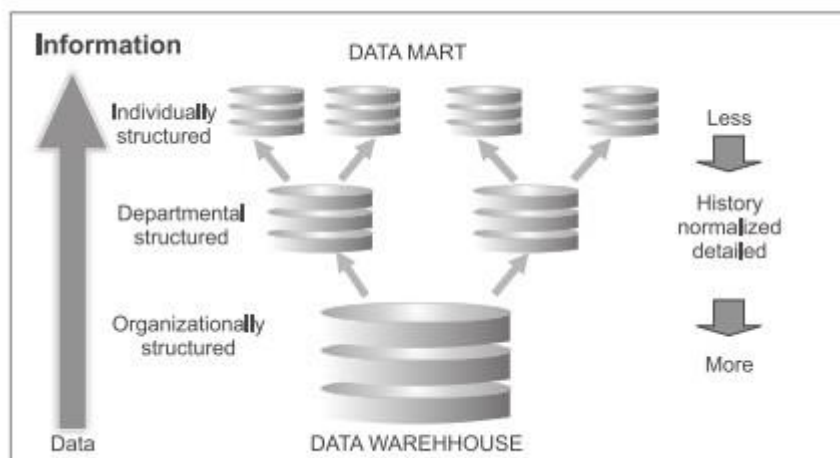
mendefinisikan data mart sebagai 'Subset data warehouse yang mendukung persyaratan departemen atau fungsi bisnis tertentu'.

Sebuah organisasi dapat memelihara baik data warehouse (mewakili data semua departemen dengan cara yang terpadu) dan data mart departemen individual. Oleh karena itu, data mart dapat berdiri sendiri (individual) atau terhubung secara terpusat ke data warehouse perusahaan seperti yang ditunjukkan pada Gambar 12.3.



Gambar 12.3 Data Mart dan Data Warehouse

Sering terlihat bahwa dengan bertumbuhnya data warehouse menjadi lebih besar dengan lebih banyak data, kemampuannya untuk melayani kebutuhan yang berbeda dari setiap perusahaan akan terganggu. Dalam kasus seperti itu, data mart datang untuk menyelamatkan, seperti pada perusahaan besar, data mart cenderung menjadi cara untuk membangun data warehouse dalam pendekatan berurutan atau bertahap. Kumpulan data mart dapat membentuk data warehouse di seluruh perusahaan. Sebaliknya, data warehouse dapat divisualisasikan sebagai kumpulan subset data mart seperti yang diilustrasikan pada Gambar 12.4.



Gambar 12.4. Hubungan antara data mart dan data warehouse

Berikut ini adalah ciri-ciri yang membedakan data mart dan data warehouse.

1. Data mart biasanya berfokus pada kebutuhan data dari beberapa departemen tertentu daripada keseluruhan organisasi.
2. Informasi rinci tidak terkandung oleh data mart (tidak seperti gudang data)

3. Mereka mudah dinavigasi, ditransfer, dan dijelajahi dibandingkan dengan data warehouse yang bekerja pada volume data yang besar.

Berikut ini kita akan menjabarkan kelebihan dan kekurangan dari data mart. Kelebihan data mart yaitu :

1. Dengan data mart, pengguna mendapatkan data yang relevan, to the point. .
2. Data mart merespons dengan cepat.
3. Operasi data seperti pembersihan data, loading, transformasi, dan integrasi jauh lebih mudah dan lebih murah karena data mart bekerja dengan volume data yang rendah. Menerapkan dan menyiapkan data mart lebih sederhana daripada menerapkan gudang data untuk seluruh perusahaan.
4. Dibandingkan dengan data warehouse, menerapkan data mart jauh lebih ekonomis.
5. Pengguna potensial dari data mart dapat dikelompokkan dengan cara yang lebih baik daripada melibatkan banyak anggota yang tidak perlu.
6. Data mart dirancang atas dasar bahwa tidak perlu melayani seluruh perusahaan. Oleh karena itu, departemen dapat meringkas, memilih, dan menyusun data departemen mereka sendiri secara mandiri.
7. Data mart dapat memungkinkan setiap departemen untuk mengerjakan bagian tertentu dari data historis daripada keseluruhan data.
8. Departemen dapat menyesuaikan perangkat lunak untuk data mart mereka sesuai kebutuhan mereka
9. Data mart lebih hemat biaya.

Sedangkan limitasi atau kekurangan dari data mart yaitu :

1. Setelah beroperasi, karena keterbatasan dalam desain akan menjadi hal yang rumit ketika akan memperluas ruang lingkup ke departemen lain.
2. Masalah Integrasi Data sering ditemui.
3. Ketika data mart mengembangkan beberapa dimensi maka masalah skalabilitas menjadi hal yang umum.

## 12.7 Data Ware House Schema

Deskripsi logis dari database dikenal sebagai Schema. Schema adalah cetak biru dari seluruh database. Schema ini mendefinisikan bagaimana data diatur dan bagaimana hubungan di antara mereka terkait. Schema data warehouse terdiri dari nama dan deskripsi dari record termasuk item data terkait dan agregat. Basis data menggunakan model relasional sedangkan data warehouse menggunakan berbagai jenis skema, yaitu, Star, Snow Flake, dan Fact Constellation.

Istilah 'dimensi' dalam data warehousing adalah kumpulan referensi informasi tentang suatu event yang terukur. Event ini disimpan dalam fact table/fact table dan dikenal sebagai fact. Dimensi umumnya entitas yang organisasi ingin disimpan dalam record. Atribut deskriptif diatur sebagai kolom dalam tabel dimensi oleh data warehouse. Misalnya, atribut dimensi siswa dapat terdiri dari nama depan dan belakang, nomor urut, usia, jenis kelamin, atau dimensi alamat yang akan menyertakan atribut nama jalan, negara bagian, dan negara.

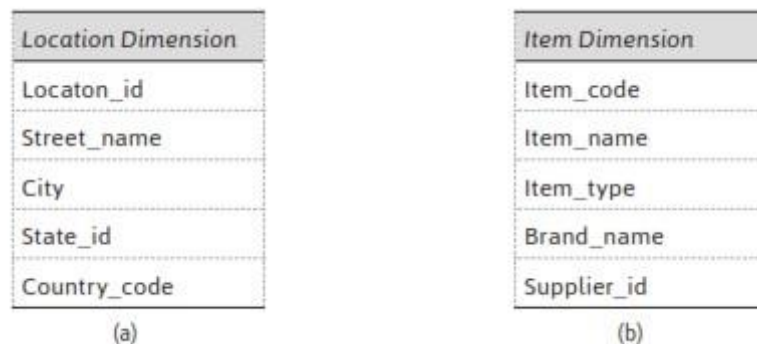
Tabel dimensi terdiri dari kolom kunci utama (Primary Key Column) yang secara unik mengidentifikasi setiap record (baris/row) dimensi. Dimensi adalah kerangka kerja yang

terdiri dari satu atau lebih hierarki yang mengklasifikasikan data. Biasanya dimensi adalah tabel yang de-normalized dan mungkin memiliki data yang berlebihan.

Mari kita rekap cepat konsep normalisasi dan de-normalisasi, seperti yang akan digunakan dalam bab ini. Normalisasi adalah proses memecah tabel yang lebih besar menjadi tabel yang lebih kecil bebas dari kemungkinan anomali insert, update atau delete. Tabel yang dinormalisasi telah mengurangi adanya redundansi data. Untuk mendapatkan informasi lengkap, tabel-tabel ini biasanya digabungkan.

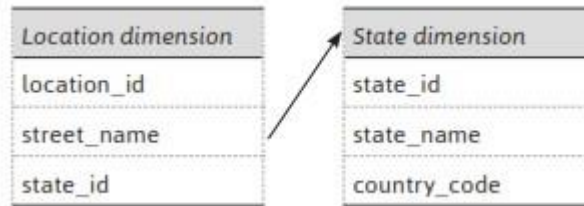
Dalam de-normalisasi, tabel yang lebih kecil digabungkan untuk membentuk tabel yang lebih besar untuk mengurangi operasi penggabungan. De-normalisasi terutama dilakukan dalam kasus-kasus di mana pengambilan merupakan persyaratan utama dan operasi insert, update, dan delete minimal, seperti dalam kasus data historis atau data warehouse. Tabel de-normalisasi ini akan memiliki redundansi data. Misalnya, dalam kasus database EMP-DEPT, akan ada dua tabel yang dinormalisasi sebagai EMP(eno, ename, job, sal, deptno) dan DEPT(deptno, dname), sedangkan dalam kasus de-normalisasi kita akan memiliki satu tabel EMP\_DEPT dengan atribut eno, ename, job, sal, deptno, dname.

Mari kita pertimbangkan dimensi Location dan Item seperti yang ditunjukkan pada Gambar 12.5. Di sini, dalam dimensi Location, location\_id adalah primary key / kunci utama dengan street\_name, city, state\_id and country\_code sebagai atributnya. Gambar 12.5 (b) menunjukkan dimensi lain yaitu Item yang memiliki item\_code sebagai primary key dan item\_name, item\_type, brand\_name, dan supplier\_id sebagai atribut lainnya.



Gambar 12.5. (a) Dimensi Location dan (b) Dimensi Item

Penting untuk dicatat bahwa dimensi tersebut mungkin tabel de-normalisasi karena dimensi Location dapat diturunkan dari tabel location\_detail (location\_id, street\_name, state\_id) dan state\_detail (state\_id, state\_name, country\_code) seperti yang ditunjukkan pada Gambar 12.6.



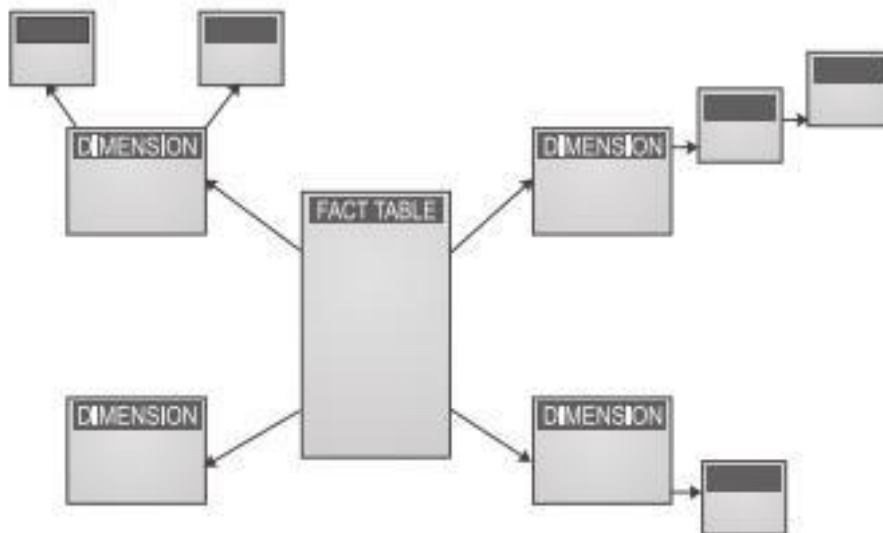
Gambar 12.6. View Normalisasi

### 12.8 Measure

Measure/ukuran adalah istilah yang digunakan untuk nilai yang bergantung pada dimensi. Misalnya, jumlah yang terjual, jumlah yang terjual, dll.

### 12.9 Fact Table

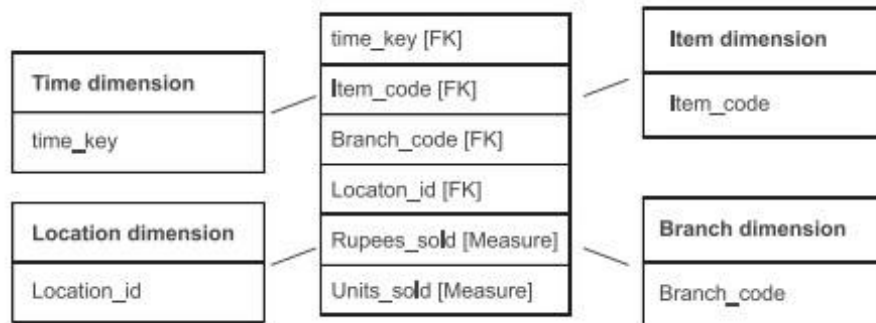
Sebuah 'fact table' adalah sekelompok item data yang terkait. Ini terdiri dari nilai-nilai dimensi dan ukuran. Ini berarti bahwa fact table dapat didefinisikan dari dimensi dan measure yang diberikan. Fact table biasanya terdiri dari dua jenis kolom yaitu foreign key dan measure. Foreign key terkait dengan tabel dimensi dan measure terdiri dari fakta numerik seperti yang ditunjukkan pada Gambar 12.7. Fact table umumnya berukuran lebih besar daripada tabel dimensi.



Gambar 12.7 Representasi Fact Table dan Tabel Dimensi

Fact table dapat menampung fact dataset pada tingkat detail atau agregat. Mari kita perhatikan fact table penjualan seperti yang ditunjukkan pada Gambar 12.4. Fact table ini memiliki time\_key, item\_code, branch\_code dan location\_id sebagai kunci asing dari table

dimensi dan rupee\_sold dan unit\_sold sebagai measure atau agregasi. Di sini, FK menunjukkan kunci asing



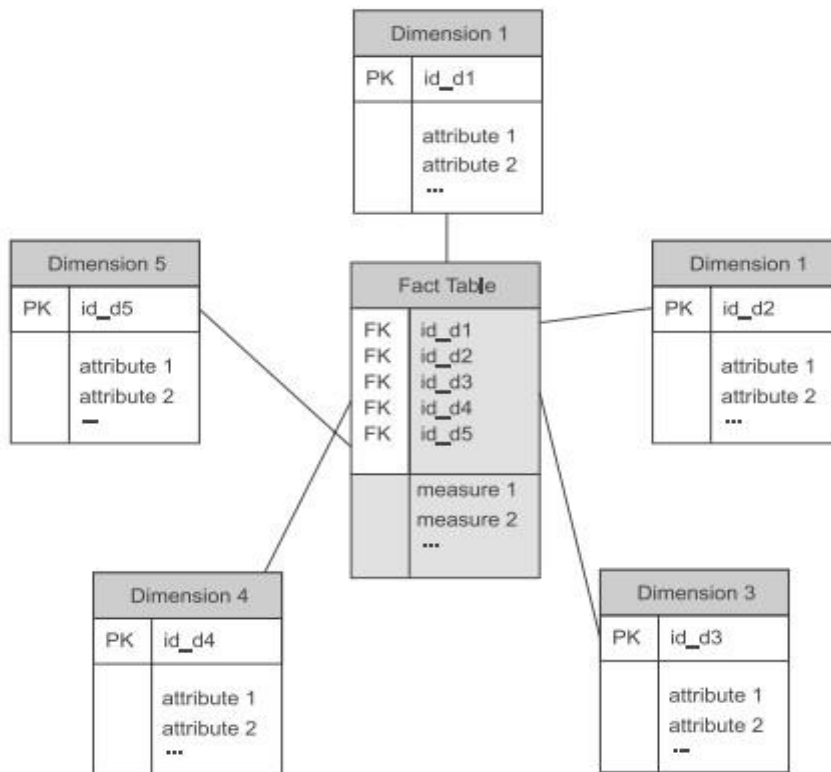
Gambar 12.8. Fact Table Sales

## 12.10 Star Schema

Star chema/Skema bintang adalah salah satu skema data warehouse yang paling sederhana. Disebut bintang karena tampak seperti bintang dengan titik-titik yang membesar dari pusatnya. Gambar 12.3 merupakan skema bintang di mana fact table berada di tengah dan tabel dimensi di node bintang.

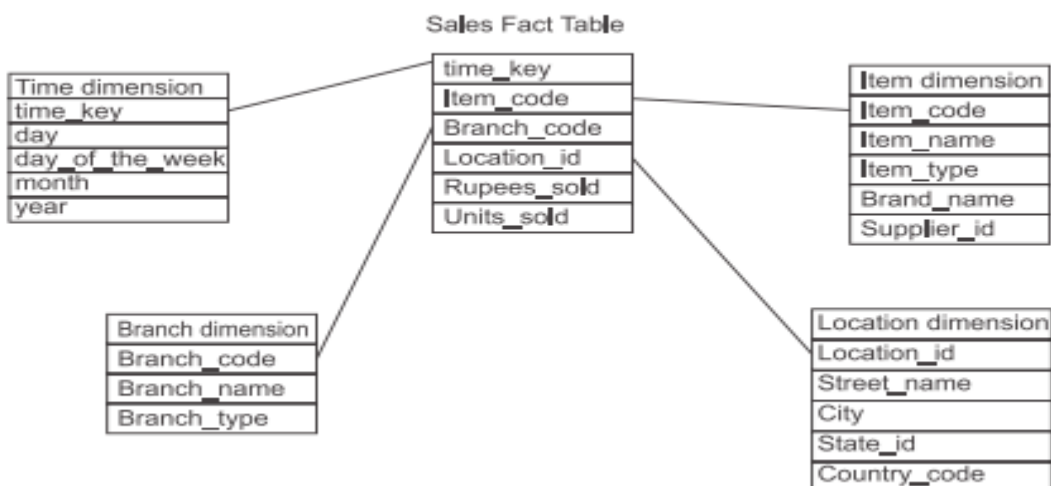
Setiap dimensi dalam skema bintang hanya mewakili satu tabel dimensi dan tabel dimensi terdiri dari sekumpulan atribut. Tabel dimensi terdiri dari record yang berjumlah relatif kecil dibandingkan dengan fact table, tetapi setiap record dapat terdiri dari sejumlah besar atribut untuk menggambarkan fact data . Fact table biasanya terdiri dari fakta numerik dan foteign key untuk data dimensi.

Umumnya, fact table dalam bentuk normal ketiga (3NF) dalam kasus skema bintang sedangkan tabel dimensi dalam bentuk de-normalisasi. Meskipun skema bintang adalah salah satu struktur paling sederhana, namun skema ini masih banyak digunakan saat ini dan direkomendasikan oleh Oracle. Gambar 12.9 secara grafis menyajikan skema bintang



Gambar 12.9. Representasi grafis dari skema Bintang (Star Schema)

Skema bintang untuk analisis penjualan suatu perusahaan ditunjukkan pada Gambar 12.6. Dalam skema ini, fakta penjualan disajikan di tengah bintang yang terdiri dari Foreign key yang terhubung ke masing-masing dari empat tabel dimensi yang sesuai. Fact table penjualan juga mencakup measure seperti rupee\_sold dan unit\_sold. Penting untuk dicatat bahwa di sini table dimensi berbentuk de-normalisasi. Misalnya, tabel dimensi Location terdiri dari kumpulan atribut seperti {location\_id, street\_name, city, state\_id, country\_code} dan mungkin memiliki redundansi data karena de-normalisasi. Contohnya, kedua kota 'Patiala' dan 'Amritsar' terletak di negara bagian Punjab di India. Catatan untuk kota-kota ini dapat mengakibatkan redundansi data yang sesuai dengan atribut negara dan negara bagian.



Gambar 12.10. Skema Star/bintang untuk analisis penjualan

Karakteristik utama dari star schema/skema bintang adalah sebagai berikut:

1. Memiliki kinerja kueri yang tinggi sebab memerlukan lebih sedikit operasi joint karena de-normalisasi data.
2. Memiliki struktur sederhana yang mudah dimengerti.
3. Dibutuhkan waktu yang relatif lebih lama untuk load data ke dalam tabel dimensi karena de-normalisasi, yang juga menyebabkan redundansi data yang dapat meningkatkan ukuran table data.
4. Merupakan struktur yang paling umum digunakan dan paling sederhana di data warehouse dan didukung oleh sejumlah besar tools.

Keuntungan dari skema bintang adalah:

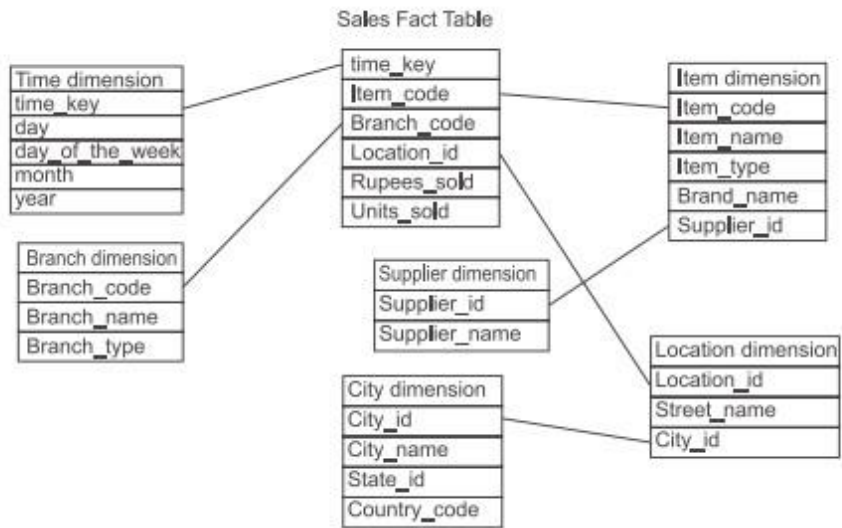
1. Memiliki kueri sederhana karena tidak memerlukan operasi joint karena de-normalisasi data.
2. Dibandingkan dengan skema yang sangat dinormalisasi, skema bintang memiliki logika pelaporan bisnis dasar.
3. Memiliki kinerja kueri yang tinggi karena lebih sedikit operasi joint. Memiliki agregasi cepat karena kuerinya yang lebih sederhana.

Kekurangan dari skema bintang adalah:

1. Integritas data tidak dapat diterapkan dalam skema bintang karena basis datanya tidak dinormalisasi( de-normalisasi) dan berisi data yang redundan.
2. Insert dan update dalam skema bintang dapat mengakibatkan anomali data karena redundansi data yang dirancang untuk dihindari oleh skema yang de-normalisasi.
3. Tidak fleksibel dalam hal kebutuhan analitis jika dibandingkan dengan model data yang dinormalisasi (normalisasi).
4. Terlalu spesifik karena dirancang untuk fixed analisis data dan tidak memungkinkan low complex analytic.

### 12.11 Snowflake Schema

Perbedaan utama antara skema star dan snowflake adalah bahwa skema snowflake dapat terdiri dari dimensi yang dinormalisasi (normalized) sedangkan skema star selalu terdiri dari dimensi yang tidak dinormalisasi (de-normalisasi). Dengan demikian, skema snowflake merupakan modifikasi dari skema star/bintang yang mendukung normalisasi tabel dimensi. Beberapa tabel dimensi dinormalisasi dalam skema snowflake sehingga membagi data menjadi tabel tambahan. Jadi, 'Snowflaking' adalah proses normalisasi tabel dimensi dalam skema bintang. Struktur yang dihasilkan tampak seperti kepingan salju yang memiliki tabel fakta di tengah ketika semua tabel dimensi dinormalisasi sepenuhnya. Misalnya, tabel dimensi Item yang ditunjukkan pada Gambar 12.11 dinormalisasi dengan membaginya menjadi dua tabel dimensi yaitu Item dan Supplier



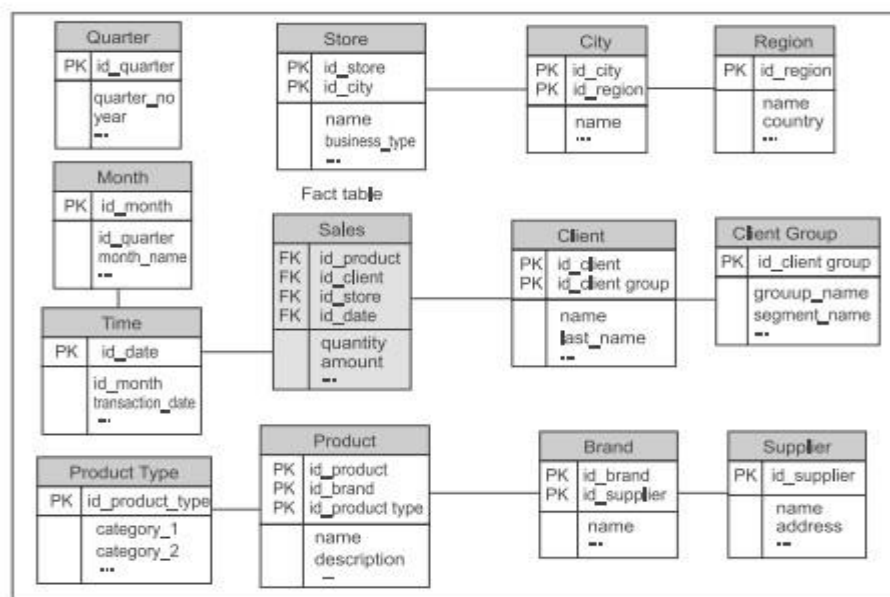
Gambar 12.11. Skema Snowflake untuk analisis penjualan

Di sini, tabel dimensi item terdiri dari kumpulan atribut {item\_code, item\_name, item\_type, brand\_name, supplier\_id}. Selanjutnya supplier\_id diasosiasikan dengan tabel dimensi supplier yang terdiri dari atribut supplier\_id dan supplier\_type.

Demikian pula, tabel dimensi location terdiri dari kumpulan atribut {location\_id, street\_name, city\_id}. Selanjutnya, city\_id diasosiasikan dengan tabel dimensi city yang terdiri dari atribut-atribut yaitu, city\_id, city\_name, state\_id dan country\_code.

Tampilan lain dari skema kepingan salju diilustrasikan pada Gambar 12.8. Di sini, dimensi shop telah dinormalisasi menjadi dimensi shop, city dan region. Demikian pula, dimensi time telah dinormalisasi menjadi dimensi time, month dan quarter. Dimensi client telah dinormalisasi menjadi dimensi client dan client group. Dan dimensi product telah dinormalisasi menjadi dimensi product type, brand dan supplier seperti yang ditunjukkan pada Gambar 12.12.

Penting untuk dicatat bahwa redundansi berkurang dalam skema snowflake karena adanya normalisasi.



Gambar 12.12. Skema Snowflake

Selanjutnya kita akan melihat kelebihan dan kekurangan yang ada pada skema snowflake.

Kelebihan dari skema snowflake yaitu :

1. Skema snowflake menghasilkan penghematan ruang penyimpanan karena atribut yang dinormalisasi meskipun ada kompleksitas tambahan dalam kueri joint dari sumber.
2. Beberapa tools pemodelan database multidimensi OLAP dioptimalkan untuk skema snowflake.

Sedangkan kekurangan dari skema snowflake adalah :

1. Skema ini memiliki kueri kompleks karena operasi joint karena adanya dimensi yang dinormalisasi.
2. Ini memiliki kinerja kueri yang buruk karena diperlukan operasi joint untuk data yang dinormalisasi.

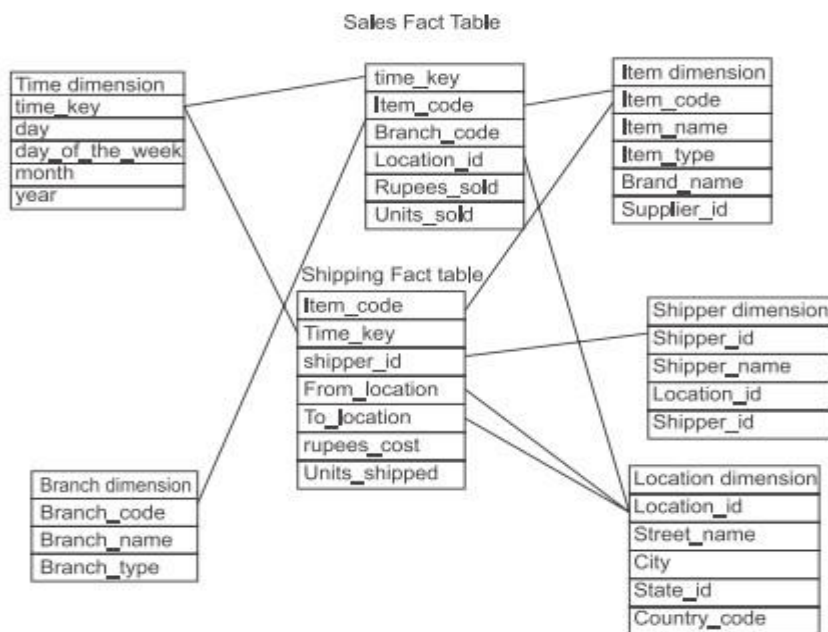
Kesimpulannya, tujuan utama skema snowflake adalah untuk menyimpan data yang dinormalisasi secara efisien tetapi dalam melakukannya menghabiskan kinerja kueri yang signifikan.

### 12.12 Fact Constellation Schema / Galaxy Schema

Perbedaan utama antara skema star/snowflake dan fact constellation adalah bahwa skema star/snowflake hanya terdiri dari satu fact table sedangkan skema fact constellation selalu terdiri dari beberapa fact table. Oleh karena itu, fact constellation juga dikenal sebagai skema galaksi (sejumlah fact table dipandang sebagai sekelompok bintang).

Fact constellation adalah measure dari pemrosesan analitik online, yang merupakan kumpulan beberapa fact table yang berbagi tabel dimensi. Skema ini adalah peningkatan dari skema Bintang.

Gambar 12.9 menunjukkan contoh skema Fact constellation karena memiliki dua fact table, yaitu sales dan shipping.



Gambar 12.9. Skema fact constellation untuk analisis penjualan

Fact table sales dalam skema fact constellation mirip dengan yang ada di skema bintang. Fact table sales di sini, juga terdiri dari empat atribut, yaitu, time\_key, item\_code, branch\_code, location\_id dan dua measure yaitu rupee\_sold dan unit\_sold seperti yang ditunjukkan pada Gambar 12.9. Dan fact table shipping terdiri dari lima atribut yaitu item\_code, time\_key, shipper\_id, from\_location, to\_location dan dua measure seperti rupee\_cost dan units\_sold. Bahkan di skema konstelasi, tabel dimensi juga dapat dibagi antara fact table. Misalnya, tabel dimensi, yaitu, location, item dan time dibagi antara act tables shipping dan sales seperti yang ditunjukkan pada Gambar 12.9.

Keuntungan utama dari skema ini adalah memberikan dukungan yang lebih baik kepada end user karena memiliki beberapa fact table. Sedangkan kerugian utama dari skema ini adalah desainnya yang rumit karena pertimbangan varian agregasi yang berbeda.

### 12.13 Comparison among Star, Snowflake and Galaxy Schema

Perbandingan antar skema data warehouse berdasarkan parameter yang berbeda dapat dilihat pada Tabel 12.1. berikut ini.

Tabel 12.1. Perbandingan antara Skema Star, Snowflake dan Fact Constellation

Parameter	Star	Snowflake	Fact Constellation
Kueri join	Membutuhkan simple join	Membutuhkan complicated join	Membutuhkan complicated join
Struktur data	De-normalized data structure	Normalized data structure	Normalized data structure
Jumlah fact table	Single fact table	Single fact table	Multiple fact table
Kinerja query	Memberikan hasil kueri yang lebih cepat karena lebih sedikit operasi joint.	Lambat dalam pemrosesan kueri karena operasi joint yang lebih besar.	Lambat dalam pemrosesan kueri karena operasi joint yang lebih besar.
Dimensi	Tabel dimensi tidak dibagi menjadi beberapa bagian.	Tabel dimensi dibagi menjadi beberapa bagian.	Tabel dimensi dibagi menjadi beberapa bagian.
Data redundansi	Data redundan karena de-normalisasi.	Data tidak redundan karena dimensi dinormalisasi.	Data tidak redundan karena dimensi dinormalisasi.
Data integrity	Sulit untuk menerapkan data integrity karena redundansi data.	Mudah untuk menegakkan integritas data karena tidak ada redundansi data	Mudah untuk menegakkan integritas data karena tidak ada redundansi

## DAFTAR PUSTAKA

1. [https://monalisa.denpasarkota.go.id/storage/proposals/Proposal\\_191002110453.pdf](https://monalisa.denpasarkota.go.id/storage/proposals/Proposal_191002110453.pdf)
2. <http://comlab.ilkom.unsri.ac.id/wp-content/uploads/2015/03/MODUL-PEMOGRAMAN-VISUAL.doc>
3. <http://comlab.ilkom.unsri.ac.id/wp-content/uploads/2015/03/MODUL-PEMOGRAMAN-VISUAL.doc>
4. Thomas Connolly and Carolyn Begg. (2015). *Database Systems: A Practical Approach To Design, Implementation, and Management*. 06. Pearson Education. USA. ISBN: 978-1-292-06118-4.



Rusito adalah seorang dosen di Universitas Sains dan Teknologi Komputer (STEKOM) Semarang. Selain menjadi dosen di Stekom juga telah menjadi tenaga pengajar di beberapa kampus ternama di Indonesia. Mulai meniti karier di bidang IT sebagai staff IT di PT. Makro Indonesia dan sekarang menjadi Lottemart. Di Stekom berada di progdi S1 Sistem Komputer yang sangat erat kaitannya dengan dunia IT. Pendidikan S1 Sistem Komputer dan S2 Sistem Informasi telah diselesaikan dengan tepat waktu. Dengan gelar pendidikan tersebut maka dapat menyandang gelar Rusito S.Kom M.Kom. Buku ini di buat untuk memenuhi tugas sebagai dosen yang menghasilkan karya ilmiah berupa buku ajar.

Sedangan isi dari buku ini adalah terdiri dari bebrpa bab yaitu :

- BAB I MENGENAL VISUAL BASIC.NET 2008
- BAB II FORM DAN OBJEK KONTROL
- BAB III FORM OBJEK KONTROL LANJUTAN
- BAB IV TIPE DATA, VARIABLE, KONSTANTA, DAN OPERATOR
- BAB V PENCABANGAN
- BAB VI PERULANGAN
- BAB VII ARRAY
- BAB VIII PROCEDURE, FUNCTION DAN MODULE
- BAB IX DATABASE
- BAB X EXCEPTION HANDLING
- BAB XI KEAMANAN DAN ADMINISTRASI DATABASE
- BAB XII KONSEP DATA WAREHOUSE DAN SCHEMA

## PEMPROGRAMAN VISUAL VB.NET 2008

ISBN 978-623-8120-23-9 (PDF)



9 786238 120239