Dr. Budi Raharjo, S.Kom, M.Kom, MM.

PENGEMBANGAN WEB

MySQL

(My Structured Query Language)

YAYASAN PRIMA AGUS TEKNIK

Dr. Budi Raharjo, S.Kom, M.Kom, MM.

PENGEMBANGAN WEB MySQL

(My Structured Query Language)

YAYASAN PRIMA AGUS TEKNIK

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

ISBN 978-634-7227-22-5 (PDF)
9 786347 227225

PENGEMBANGAN WEB MySQL
(My Structured Query Language)

Penulis :
Dr. Budi Raharjo, S.Kom, M.Kom, MM.

ISBN : 978-634-7227-22-5

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

Penyunting :
Dr. Joseph Teguh Santoso, M.Kom.

Desain Sampul dan Tata Letak :
Irdha Yunianto, S.Ds., M.Kom.

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

Anggota IKAPI No: 279 / ALB / JTE / 2023

Redaksi :
Jl. Majapahit no 605 Semarang
Telp. (024) 6723456
Fax. 024-6710144
Email : penerbit_ypat@stekom.ac.id

Distributor Tunggal :
Universitas STEKOM
Jl. Majapahit no 605 Semarang
Telp. (024) 6723456
Fax. 024-6710144
Email : info@stekom.ac.id

KATA PENGANTAR

Puji syukur ke hadirat Tuhan Yang Maha Esa atas segala rahmat dan karunia-Nya, sehingga buku berjudul *Pengembangan WEB MySQL* ini dapat disusun dan hadir di hadapan pembaca. Buku ini dirancang sebagai panduan komprehensif bagi siapa saja yang ingin memahami, membangun, dan mengembangkan aplikasi web berbasis MySQL secara efektif dan aman.

Di era digital saat ini, kebutuhan akan pengelolaan data yang efisien dan aman menjadi sangat penting, terutama dalam pengembangan aplikasi web dan e-commerce. Oleh karena itu, buku ini hadir untuk memberikan pemahaman menyeluruh mulai dari konsep dasar basis data relasional, perancangan basis data web, hingga implementasi tingkat lanjut yang mencakup keamanan, autentikasi, dan transaksi.

Buku ini dimulai dengan pembahasan konsep dasar basis data relasional, teknik desain basis data web, serta arsitektur basis data web modern pada Bab 1. Selanjutnya, Bab 2 mengulas langkah-langkah praktis dalam membuat basis data web menggunakan MySQL, mulai dari instalasi, pembuatan pengguna, hingga pengelolaan hak istimewa dan tipe data. Pada Bab 3, pembaca diperkenalkan dengan dasar-dasar SQL, teknik join tabel, serta pengelompokan dan penggabungan data. Bab 4 membahas integrasi MySQL dengan PHP untuk mengakses dan memanipulasi basis data melalui web, termasuk pentingnya validasi dan pemilihan basis data yang tepat. Bab 5 mengupas fitur-fitur lanjutan MySQL, seperti sistem hak istimewa, keamanan basis data, optimasi kueri, dan berbagai jenis tabel. Kemudian, Bab 6 membahas penerapan basis data dalam situs e-commerce, jenis-jenis situs komersial, serta risiko dan strategi pengelolaannya. Bab 7 fokus pada isu-isu keamanan e-commerce, mulai dari ancaman keamanan, kehilangan data, hingga upaya menyeimbangkan kegunaan, performa, biaya, dan keamanan server. Bab 8 membahas penerapan autentikasi menggunakan PHP dan MySQL, teknik penyimpanan dan enkripsi kata sandi, serta kontrol akses pengguna. Terakhir, Bab 9 membahas transaksi aman dalam aplikasi web, penggunaan Secure Sockets Layer (SSL), dan penyimpanan data yang aman di internet.

Setiap bab dilengkapi dengan penjelasan yang mudah dipahami, contoh kasus nyata, dan langkah-langkah praktis yang dapat langsung diterapkan oleh pembaca. Buku ini diharapkan dapat menjadi referensi utama bagi mahasiswa, pengembang web, serta siapa saja yang ingin memperdalam pengetahuan tentang pengelolaan basis data web menggunakan MySQL.

Akhir kata, penulis mengucapkan terima kasih kepada semua pihak yang telah membantu dalam penyusunan buku ini. Semoga buku *Pengembangan WEB MySQL* dapat memberikan manfaat dan menjadi inspirasi dalam pengembangan aplikasi web yang handal dan aman.

Semarang, Juli 2025
Penulis

Dr. Budi Raharjo, S.Kom, M.Kom, MM.

DAFTAR ISI

BAB 1
MERANCANG BASIS DATA

Sekarang setelah Anda memahami dasar-dasar PHP, kita akan mulai membahas cara mengintegrasikan basis data ke dalam skrip Anda. Seperti yang mungkin Anda ingat, "Menyimpan dan Mengambil Data," kita membahas tentang keuntungan menggunakan basis data relasional daripada berkas datar. Keuntungan tersebut meliputi:

- RDBMS dapat menyediakan akses data yang lebih cepat daripada berkas datar.
- RDBMS dapat dengan mudah dikueri untuk mengekstrak kumpulan data yang sesuai dengan kriteria tertentu.
- RDBMS memiliki mekanisme bawaan untuk menangani akses bersamaan sehingga Anda sebagai programmer tidak perlu mengkhawatirkannya.
- RDBMS menyediakan akses acak ke data Anda.
- RDBMS memiliki sistem hak istimewa bawaan.

Dalam istilah yang lebih konkret, menggunakan basis data relasional memungkinkan Anda menjawab pertanyaan tentang asal pelanggan Anda, produk mana yang paling laku, atau jenis pelanggan yang paling banyak berbelanja dengan cepat dan mudah. Informasi ini dapat membantu Anda meningkatkan situs untuk menarik dan mempertahankan lebih banyak pengguna. Basis data yang akan kita gunakan di bagian ini adalah MySQL. Sebelum kita membahas secara spesifik MySQL di bab berikutnya, kita perlu membahas

- Konsep dan terminologi basis data relasional
- Desain basis data web
- Arsitektur basis data web Bab-bab berikut mencakup
- Bab 2, "Membuat Basis Data Web Anda," membahas konfigurasi dasar yang Anda perlukan untuk menghubungkan basis data MySQL Anda ke Web.
- Bab 3, "Bekerja dengan Basis Data MySQL Anda," menjelaskan cara melakukan kueri pada basis data, menambahkan dan menghapus rekaman, semuanya dari baris perintah.
- Bab 4, "Mengakses Basis Data MySQL Anda dari Web dengan PHP," menjelaskan cara menghubungkan PHP dan MySQL bersama-sama sehingga Anda dapat menggunakan dan mengelola basis data Anda dari antarmuka Web.
- Bab 5, "MySQL Lanjutan," membahas beberapa fitur lanjutan MySQL yang dapat berguna saat mengembangkan aplikasi berbasis Web yang lebih menuntut.

1.1 KONSEP BASIS DATA RELASIONAL

Basis data relasional sejauh ini merupakan jenis basis data yang paling umum digunakan. Basis data ini bergantung pada dasar teori yang kuat dalam aljabar relasional. Anda tidak perlu memahami teori relasional untuk menggunakan basis data relasional (yang merupakan hal yang baik), tetapi Anda perlu memahami beberapa konsep dasar basis data.

Tabel

Basis data relasional terdiri dari relasi, yang lebih umum disebut tabel. Tabel adalah seperti namanya—tabel data. Jika Anda pernah menggunakan spreadsheet elektronik, Anda telah menggunakan tabel relasional.

Pada Gambar 1.1, Anda dapat melihat contoh tabel. Tabel ini berisi nama dan alamat pelanggan toko buku, Book-O-Rama.

Pelanggan			
CustomerID	**Name**	**Alamat**	**Kota**
1	Julie Smith	25 Oak Street	Airport West
2	Alan Wong	1/47 Haines Avenue	Box Hill
3	Michelle Arthur	357 North Road	Yarraville

Gambar 1.1 Rincian pelanggan Book-O-Rama disimpan dalam tabel.

Tabel tersebut memiliki nama (Pelanggan), sejumlah kolom, yang masing-masing terkait dengan bagian data yang berbeda, dan baris yang terkait dengan masing-masing pelanggan.

Kolom

Setiap kolom dalam tabel memiliki nama yang unik dan berisi data yang berbeda. Setiap kolom memiliki tipe data terkait. Misalnya, dalam tabel Pelanggan pada Gambar 1.1, Anda dapat melihat bahwa IDPelanggan adalah bilangan bulat dan tiga kolom lainnya adalah string. Kolom terkadang disebut bidang atau atribut.

Baris

Setiap baris dalam tabel mewakili pelanggan yang berbeda. Karena format tabel, semuanya memiliki atribut yang sama. Baris juga disebut rekaman atau tupel.

Nilai

Setiap baris terdiri dari sekumpulan nilai individual yang terkait dengan kolom. Setiap nilai harus memiliki tipe data yang ditentukan oleh kolomnya.

Kunci

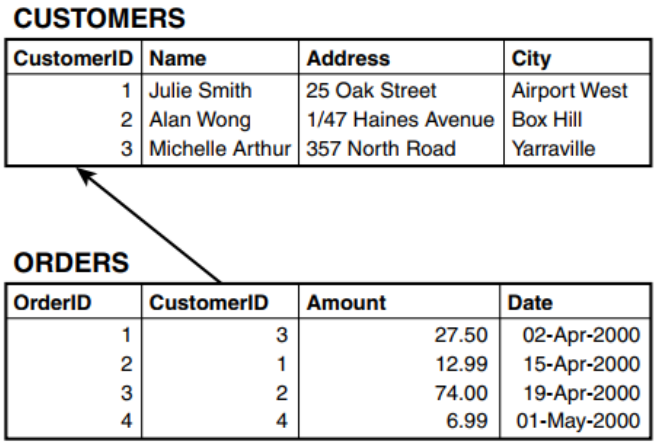
Kita perlu memiliki cara untuk mengidentifikasi setiap pelanggan tertentu. Nama biasanya bukan cara yang baik untuk melakukan ini—jika Anda memiliki nama yang umum, Anda mungkin akan mengerti alasannya. Ambil contoh Julie Smith dari tabel Pelanggan. Jika saya membuka direktori telepon saya, ada terlalu banyak daftar nama itu untuk dihitung.

Kita dapat membedakan Julie dengan beberapa cara. Kemungkinannya, dia adalah satu-satunya Julie Smith yang tinggal di alamatnya. Membicarakan tentang "Julie Smith, dari 25 Oak Street, Airport West" cukup merepotkan dan terdengar seperti bahasa hukum. Itu juga memerlukan penggunaan lebih dari satu kolom dalam tabel.

Apa yang telah kita lakukan dalam contoh ini, dan apa yang mungkin akan Anda lakukan dalam aplikasi Anda, adalah menetapkan CustomerID yang unik. Ini adalah prinsip yang sama yang mengarah pada Anda memiliki nomor rekening bank atau nomor keanggotaan klub yang unik. Itu membuat penyimpanan detail Anda dalam basis data menjadi lebih mudah. Nomor identifikasi yang ditetapkan secara artifisial dapat dijamin unik. Beberapa informasi nyata, bahkan jika digunakan dalam kombinasi, memiliki properti ini.

Kolom identifikasi dalam tabel disebut kunci atau kunci utama. Kunci juga dapat terdiri dari beberapa kolom. Misalnya, jika kita memilih untuk menyebut Julie sebagai "Julie Smith, dari 25 Oak Street, Airport West," kuncinya akan terdiri dari kolom Nama, Alamat, dan Kota dan tidak dapat dijamin unik.

Basis data biasanya terdiri dari beberapa tabel dan menggunakan kunci sebagai referensi dari satu tabel ke tabel lainnya. Pada Gambar 1.2, kita telah menambahkan tabel kedua ke basis data. Tabel ini menyimpan pesanan yang dilakukan oleh pelanggan. Setiap baris dalam tabel Pesanan mewakili satu pesanan, yang dilakukan oleh satu pelanggan. Kita mengetahui siapa pelanggan tersebut karena kita menyimpan ID Pelanggan mereka. Kita dapat melihat pesanan dengan IDPesanan 2, misalnya, dan melihat bahwa pelanggan dengan IDPelanggan 1 yang memesannya. Jika Anda kemudian melihat tabel Pelanggan, Anda dapat melihat bahwa IDPelanggan 1 merujuk kepada Julie Smith.



CUSTOMERS

CustomerID	Name	Address	City
1	Julie Smith	25 Oak Street	Airport West
2	Alan Wong	1/47 Haines Avenue	Box Hill
3	Michelle Arthur	357 North Road	Yarraville

ORDERS

OrderID	CustomerID	Amount	Date
1	3	27.50	02-Apr-2000
2	1	12.99	15-Apr-2000
3	2	74.00	19-Apr-2000
4	4	6.99	01-May-2000

Gambar 1.2 Setiap pesanan dalam tabel Pesanan merujuk ke pelanggan dari tabel Pelanggan.

Istilah basis data relasional untuk hubungan ini adalah kunci asing. CustomerID adalah kunci utama dalam Customers, tetapi ketika muncul di tabel lain, seperti Orders, maka disebut sebagai kunci asing.

Anda mungkin bertanya-tanya mengapa kami memilih untuk memiliki dua tabel terpisah—mengapa tidak menyimpan alamat Julie saja di tabel Orders? Kami akan membahasnya lebih rinci di bagian berikutnya.

Skema

Rangkaian lengkap desain tabel untuk basis data disebut skema basis data. Skema ini mirip dengan cetak biru untuk basis data. Skema harus menunjukkan tabel beserta kolomnya, tipe data kolom, dan menunjukkan kunci utama setiap tabel dan kunci asing apa pun. Skema tidak menyertakan data apa pun, tetapi Anda mungkin ingin menunjukkan contoh data dengan skema Anda untuk menjelaskan kegunaannya. Skema dapat ditampilkan sebagaimana adanya dalam diagram yang kami gunakan, dalam diagram hubungan entitas (yang tidak dibahas dalam buku ini), atau dalam bentuk teks, seperti

```
Customers(CustomerID, Name, Address, City)
```

```
Orders(OrderID, CustomerID, Amount, Date)
```

Istilah yang digarisbawahi dalam skema adalah kunci utama dalam relasi tempat istilah tersebut digarisbawahi. Istilah yang digarisbawahi dengan titik-titik adalah kunci asing dalam relasi tempat istilah tersebut muncul dengan garis bawah titik-titik.

Relasi

Kunci asing mewakili relasi antara data dalam dua tabel. Misalnya, tautan dari Pesanan ke Pelanggan mewakili relasi antara baris dalam tabel Pesanan dan baris dalam tabel Pelanggan.

Ada tiga jenis relasi dasar dalam basis data relasional. Relasi tersebut diklasifikasikan menurut jumlah hal di setiap sisi relasi. Relasi dapat berupa satu-ke-satu, satu-ke-banyak, atau banyak-ke-banyak.

Relasi satu-ke-satu berarti bahwa ada satu dari setiap hal dalam relasi tersebut. Misalnya, jika kita meletakkan alamat dalam tabel terpisah dari Pelanggan, akan ada relasi satu-ke-satu di antara keduanya. Anda dapat memiliki kunci asing dari Alamat ke Pelanggan atau sebaliknya (keduanya tidak diperlukan). Dalam hubungan satu ke banyak, satu baris dalam satu tabel ditautkan ke banyak baris di tabel lain. Dalam contoh ini, satu Pelanggan mungkin memesan banyak Pesanan. Dalam hubungan ini, tabel yang berisi banyak baris akan memiliki kunci asing ke tabel dengan satu baris. Di sini, kami telah memasukkan CustomerID ke dalam tabel Pesanan untuk menunjukkan hubungan tersebut.

Dalam hubungan banyak ke banyak, banyak baris dalam satu tabel ditautkan ke banyak baris di tabel lain. Misalnya, jika kita memiliki dua tabel, Buku dan Penulis, Anda mungkin menemukan bahwa satu buku telah ditulis oleh dua rekan penulis, yang masing-masing telah menulis buku lain, sendiri atau mungkin dengan penulis lain. Jenis hubungan ini biasanya mendapatkan tabel untuk dirinya sendiri, jadi Anda mungkin memiliki Buku, Penulis, dan Buku_Penulis. Tabel ketiga ini hanya akan berisi kunci tabel lain sebagai kunci asing berpasangan, untuk menunjukkan penulis mana yang terlibat dengan buku mana.

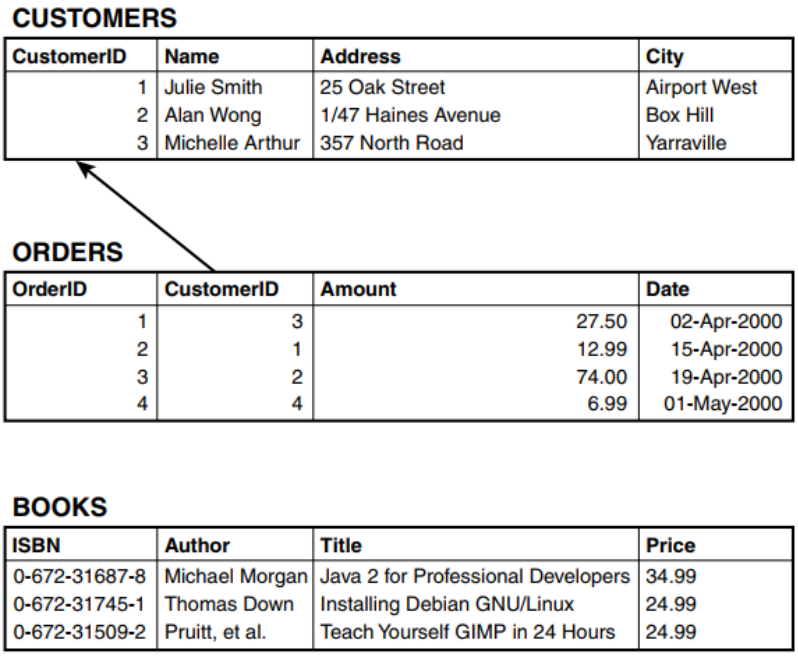
1.2 CARA MENDESAIN BASIS DATA WEB

Mengetahui kapan Anda memerlukan tabel baru dan apa kuncinya bisa menjadi semacam seni. Anda dapat membaca banyak sekali informasi tentang diagram hubungan entitas dan normalisasi basis data, yang berada di luar cakupan buku ini. Namun, sebagian besar waktu, Anda dapat mengikuti beberapa prinsip desain dasar. Mari kita pertimbangkan ini dalam konteks Book-O-Rama.

Pikirkan Objek Dunia Nyata yang Anda Modelkan

Saat Anda membuat basis data, Anda biasanya memodelkan item dan hubungan dunia nyata serta menyimpan informasi tentang objek dan hubungan tersebut. Umumnya, setiap kelas objek dunia nyata yang Anda modelkan akan memerlukan tabelnya sendiri. Pikirkan tentang hal ini: Kita ingin menyimpan informasi yang sama tentang semua pelanggan kita. Jika ada sekumpulan data yang memiliki "bentuk" yang sama, kita dapat dengan mudah membuat tabel yang sesuai dengan data tersebut.

Dalam contoh Book-O-Rama, kita ingin menyimpan informasi tentang pelanggan kita, buku yang kita jual, dan detail pesanan. Semua pelanggan memiliki nama dan alamat. Pesanan memiliki tanggal, jumlah total, dan sekumpulan buku yang dipesan. Buku-buku tersebut memiliki ISBN, penulis, judul, dan harga. Ini menunjukkan bahwa kita memerlukan setidaknya tiga tabel dalam basis data ini: Pelanggan, Pesanan, dan Buku. Skema awal ini ditunjukkan pada Gambar 1.3.



CUSTOMERS

CustomerID	Name	Address	City
1	Julie Smith	25 Oak Street	Airport West
2	Alan Wong	1/47 Haines Avenue	Box Hill
3	Michelle Arthur	357 North Road	Yarraville

ORDERS

OrderID	CustomerID	Amount	Date
1	3	27.50	02-Apr-2000
2	1	12.99	15-Apr-2000
3	2	74.00	19-Apr-2000
4	4	6.99	01-May-2000

BOOKS

ISBN	Author	Title	Price
0-672-31687-8	Michael Morgan	Java 2 for Professional Developers	34.99
0-672-31745-1	Thomas Down	Installing Debian GNU/Linux	24.99
0-672-31509-2	Pruitt, et al.	Teach Yourself GIMP in 24 Hours	24.99

Gambar 1.3 Skema awal terdiri dari Pelanggan, Pesanan, dan Buku.

Saat ini, kita tidak dapat mengetahui dari model buku mana yang dipesan dalam setiap pesanan. Kita akan membahasnya sebentar lagi.

Hindari Menyimpan Data yang Berlebihan

Sebelumnya, kita mengajukan pertanyaan: "Mengapa tidak menyimpan alamat Julie Smith saja di tabel Pesanan?"

Jika Julie memesan dari Book-O-Rama beberapa kali, yang kami harap akan dilakukannya, kita akan menyimpan datanya beberapa kali. Anda mungkin akan mendapatkan tabel Pesanan yang tampak seperti yang ditunjukkan pada Gambar 1.4.

Ada dua masalah dasar dengan ini.

Yang pertama adalah pemborosan ruang. Mengapa menyimpan detail Julie tiga kali jika kita hanya perlu menyimpannya sekali?

ORDERS

OrderID	Amount	Date	CustomerID	Name	Address	City
12	199.50	25-Apr-2000	1	Julie Smith	28 Oak Street	Airport West
13	43.00	29-Apr-2000	1	Julie Smith	28 Oak Street	Airport West
14	15.99	30-Apr-2000	1	Julie Smith	28 Oak Street	Airport West
15	23.75	01-May-2000	1	Julie Smith	28 Oak Street	Airport West

Gambar 1.4 Desain basis data yang menyimpan data yang berlebihan akan menghabiskan ruang ekstra dan dapat menyebabkan anomali dalam data.

Masalah kedua adalah hal ini dapat menyebabkan anomali pembaruan, yaitu situasi saat kita mengubah basis data dan berakhir dengan data yang tidak konsisten. Integritas data dilanggar dan kita tidak lagi tahu data mana yang benar dan mana yang salah. Hal ini umumnya menyebabkan hilangnya informasi.

Tiga jenis anomali pembaruan perlu dihindari: anomali modifikasi, penyisipan, dan penghapusan. Jika Julie pindah rumah saat ia memiliki pesanan yang tertunda, kita perlu memperbarui alamatnya di tiga tempat, bukan satu, sehingga membutuhkan tiga kali lebih banyak pekerjaan. Fakta ini mudah diabaikan dan hanya mengubah alamatnya di satu tempat, yang menyebabkan data tidak konsisten dalam basis data (hal yang sangat buruk). Masalah ini disebut anomali modifikasi karena terjadi saat kita mencoba mengubah basis data.

Dengan desain ini, kita perlu memasukkan detail Julie setiap kali kita menerima pesanan, jadi setiap kali kita harus memeriksa dan memastikan bahwa detailnya konsisten dengan baris yang ada di tabel. Jika kita tidak memeriksa, kita mungkin akan mendapatkan dua baris informasi yang saling bertentangan tentang Julie. Misalnya, satu baris mungkin memberi tahu kita bahwa Julie tinggal di Airport West, dan baris lainnya mungkin memberi tahu kita bahwa dia tinggal di Airport. Ini disebut anomali penyisipan karena terjadi saat data sedang dimasukkan.

Jenis anomali ketiga disebut anomali penghapusan karena terjadi (kejutan, kejutan) saat kita menghapus baris dari basis data. Misalnya, bayangkan bahwa saat pesanan telah dikirim, kita menghapusnya dari basis data. Saat semua pesanan Julie saat ini telah dipenuhi, semuanya dihapus dari tabel Pesanan. Ini berarti bahwa kita tidak lagi memiliki catatan alamat Julie. Kita tidak dapat mengirimkan penawaran khusus kepadanya, dan lain kali dia ingin memesan sesuatu dari kita, kita harus mendapatkan detailnya lagi. Umumnya Anda ingin mendesain basis data Anda sehingga tidak ada anomali ini yang terjadi.

Gunakan Nilai Kolom Atomik

Ini berarti bahwa di setiap atribut di setiap baris, kita hanya menyimpan satu hal. Misalnya, kita perlu mengetahui buku apa saja yang termasuk dalam setiap pesanan. Ada beberapa cara untuk melakukannya.

Kita dapat menambahkan kolom ke tabel Pesanan yang mencantumkan semua buku yang telah dipesan, seperti yang ditunjukkan pada Gambar 1.5.

ORDERS

OrderID	CustomerID	Amount	Date	Books Ordered
1	3	27.50	02-Apr-2000	0-672-31697-8
2	1	12.99	15-Apr-2000	0-672-31745-1, 0-672-31509-2
3	2	74.00	19-Apr-2000	0-672-31697-8
4	3	6.99	01-May-2000	0-672-31745-1, 0-672-31509-2, 0-672-31697-8

Gambar 1.5 Dengan desain ini, atribut Buku yang Diurutkan di setiap baris memiliki beberapa nilai.

Ini bukan ide yang bagus karena beberapa alasan. Yang sebenarnya kita lakukan adalah menumpuk seluruh tabel di dalam satu kolom—tabel yang menghubungkan pesanan dengan buku. Jika Anda melakukannya dengan cara ini, akan lebih sulit untuk menjawab pertanyaan seperti "Berapa banyak salinan Java 2 for Professional Developers yang telah dipesan?" Sistem tidak dapat lagi hanya menghitung bidang yang cocok. Sebaliknya, sistem harus mengurai setiap nilai atribut untuk melihat apakah ada kecocokan di dalamnya.

Karena kita benar-benar membuat tabel di dalam tabel, kita harus membuat tabel baru itu saja. Tabel baru ini disebut Order_Items dan ditunjukkan pada Gambar 1.6.

ORDER_ITEMS

OrderID	ISBN	Quantity
1	0-672-31697-8	1
2	0-672-31745-1	2
2	0-672-31509-2	1
3	0-672-31697-8	1
4	0-672-31745-1	1
4	0-672-31509-2	2
4	0-672-31697-8	1

Gambar 1.6 Desain ini memudahkan pencarian buku-buku tertentu yang telah dipesan.

Tabel ini menyediakan tautan antara tabel Orders dan tabel Books. Jenis tabel ini umum digunakan jika terdapat hubungan many-to-many antara dua objek—dalam kasus ini, satu pesanan mungkin terdiri dari banyak buku, dan setiap buku dapat dipesan oleh banyak orang.

Pilih Kunci yang Masuk Akal

Pastikan kunci yang Anda pilih unik. Dalam kasus ini, kami telah membuat kunci khusus untuk pelanggan (`CustomerID`) dan untuk pesanan (`OrderID`) karena objek dunia nyata ini mungkin tidak memiliki pengenal yang dapat dijamin unik. Kami tidak perlu membuat pengenal unik untuk buku—ini telah dilakukan, dalam bentuk ISBN.

Untuk `Order_Item`, Anda dapat menambahkan kunci tambahan jika Anda mau, tetapi kombinasi dari dua atribut `OrderID` dan ISBN akan unik selama lebih dari satu salinan buku yang sama dalam pesanan diperlakukan sebagai satu baris. Karena alasan ini, tabel `Order_Items` memiliki kolom `Quantity`.

Pikirkan Pertanyaan yang Ingin Anda Ajukan ke Basis Data

Melanjutkan dari bagian terakhir, pikirkan pertanyaan apa yang ingin Anda jawab di basis data. (Pikirkan kembali pertanyaan-pertanyaan yang telah kami sebutkan di awal bab ini.

Misalnya, apa saja buku terlaris Book-O-Rama?) Pastikan bahwa basis data berisi semua data yang diperlukan, dan bahwa tautan yang sesuai tersedia di antara tabel untuk menjawab pertanyaan yang Anda miliki.

Hindari Desain dengan Banyak Atribut Kosong

Jika kita ingin menambahkan ulasan buku ke basis data, setidaknya ada dua cara yang dapat kita lakukan. Kedua pendekatan ini ditunjukkan pada Gambar 1.7.

BOOKS

ISBN	Author	Title	Price	Review
0-672-31687-8	Michael Morgan	Java 2 for Professional Developers	34.99	
0-672-31745-1	Thomas Down	Installing Debian GNU/Linux	24.99	
0-672-31509-2	Pruitt, et al.	Teach Yourself GIMP in 24 Hours	24.99	

BOOK_REVIEWS

ISBN	Review

Gambar 1.7 Untuk menambahkan ulasan, kita dapat menambahkan kolom Ulasan ke tabel Buku, atau menambahkan tabel khusus untuk ulasan.

Cara pertama berarti menambahkan kolom Ulasan ke tabel Buku. Dengan cara ini, ada kolom untuk Ulasan yang akan ditambahkan untuk setiap buku. Jika banyak buku ada di dalam basis data, dan pengulas tidak berencana untuk mengulas semuanya, banyak baris tidak akan memiliki nilai dalam atribut ini. Ini disebut memiliki nilai null.

Memiliki banyak nilai null dalam basis data Anda adalah ide yang buruk. Ini membuang-buang ruang penyimpanan dan menyebabkan masalah saat menghitung total dan fungsi lainnya pada kolom numerik. Saat pengguna melihat null dalam tabel, mereka tidak tahu apakah itu karena atribut ini tidak relevan, apakah ada kesalahan dalam basis data, atau apakah datanya belum dimasukkan.

Anda biasanya dapat menghindari masalah dengan banyak null dengan menggunakan desain alternatif. Dalam kasus ini, kita dapat menggunakan desain kedua yang diusulkan dalam Gambar 1.7. Di sini, hanya buku dengan ulasan yang dicantumkan dalam tabel Book_Reviews, beserta ulasannya.

Perlu diketahui bahwa desain ini didasarkan pada gagasan untuk memiliki satu pengulas internal. Kita juga dapat dengan mudah mengizinkan pelanggan menulis ulasan. Jika kita ingin melakukan ini, kita dapat menambahkan `CustomerID` ke tabel Book_Reviews.

Ringkasan Jenis Tabel

Biasanya Anda akan menemukan bahwa desain basis data Anda terdiri dari dua jenis tabel:

- Tabel sederhana yang menggambarkan objek dunia nyata. Tabel ini mungkin juga berisi kunci ke objek sederhana lainnya yang memiliki hubungan satu-ke-satu atau satu-ke-banyak. Misalnya, satu pelanggan mungkin memiliki banyak pesanan, tetapi pesanan dilakukan oleh satu pelanggan. Jadi, kita mencantumkan referensi ke pelanggan dalam pesanan.

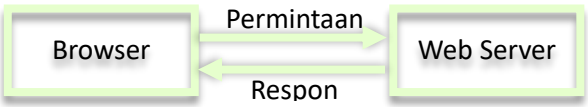
- Tabel penghubung yang menggambarkan hubungan banyak-ke-banyak antara dua objek nyata seperti hubungan antara Pesanan dan Buku. Tabel ini sering dikaitkan dengan beberapa jenis transaksi dunia nyata.

1.3 ARSITEKTUR BASIS DATA WEB

Sekarang setelah kita membahas arsitektur internal basis data Anda, kita akan melihat arsitektur eksternal sistem basis data Web, dan membahas metodologi untuk mengembangkan sistem basis data Web.

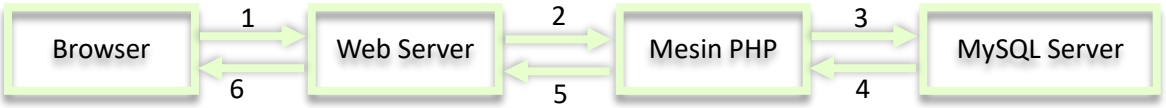
Arsitektur

Operasi dasar server Web ditunjukkan pada Gambar 1.8. Sistem ini terdiri dari dua objek: peramban web dan server web. Diperlukan tautan komunikasi di antara keduanya. Peramban web mengajukan permintaan ke server. Server mengirimkan respons. Arsitektur ini cocok untuk server yang menyediakan halaman statis. Arsitektur yang menyediakan situs web yang didukung basis data sedikit lebih rumit.



Gambar 1.8 Hubungan klien/server antara peramban web dan server web memerlukan komunikasi.

Aplikasi basis data web yang akan kita bangun dalam buku ini mengikuti struktur basis data web umum yang ditunjukkan pada Gambar 1.9. Sebagian besar struktur ini seharusnya sudah familier bagi Anda.



Gambar 1.9 Arsitektur basis data web dasar terdiri dari peramban web, server web, mesin skrip, dan server basis data.

Transaksi basis data Web yang umum terdiri dari tahap-tahap berikut, yang diberi nomor pada Gambar 1.9. Kita akan memeriksa tahap-tahap tersebut dalam konteks contoh Book-O-Rama.

1. Peramban Web pengguna mengeluarkan permintaan HTTP untuk halaman Web tertentu. Misalnya, ia mungkin telah meminta pencarian untuk semua buku di Book-O-Rama yang ditulis oleh Laura Thomson, menggunakan formulir HTML. Halaman hasil pencarian disebut `results.php`.
2. Server Web menerima permintaan untuk results.php, mengambil berkas, dan meneruskannya ke mesin PHP untuk diproses.

3. Mesin PHP mulai mengurai skrip. Di dalam skrip tersebut terdapat perintah untuk terhubung ke basis data dan menjalankan kueri (melakukan pencarian buku). PHP membuka koneksi ke server MySQL dan mengirimkan kueri yang sesuai.
4. Server MySQL menerima kueri basis data dan memprosesnya, dan mengirimkan hasilnya—daftar buku—kembali ke mesin PHP.
5. Mesin PHP selesai menjalankan skrip, yang biasanya melibatkan pemformatan hasil kueri dengan baik dalam HTML. Kemudian, mesin tersebut mengembalikan HTML yang dihasilkan ke server Web.
6. Server Web meneruskan HTML kembali ke browser, tempat pengguna dapat melihat daftar buku yang diminta.

Prosesnya pada dasarnya sama, apa pun mesin skrip atau server basis data yang Anda gunakan. Sering kali perangkat lunak server Web, mesin PHP, dan server basis data semuanya berjalan pada mesin yang sama. Namun, server basis data juga cukup umum berjalan pada mesin yang berbeda. Anda mungkin melakukan ini karena alasan keamanan, peningkatan kapasitas, atau penyebaran beban. Dari perspektif pengembangan, cara kerjanya akan sama saja, tetapi mungkin menawarkan beberapa keuntungan signifikan dalam hal kinerja.

BAB 2
MEMBUAT BASIS DATA WEB

Dalam bab ini, kita akan membahas cara menyiapkan basis data MySQL untuk digunakan di situs Web. Kita akan membahas

- Membuat basis data
- Pengguna dan hak istimewa
- Pengenalan sistem hak istimewa
- Membuat tabel basis data
- Jenis kolom dalam MySQL

Dalam bab ini, kita akan membahas aplikasi toko buku daring Book-O-Rama yang dibahas di bab sebelumnya. Sebagai pengingat, berikut adalah skema untuk aplikasi Book-O-Rama:

```
Customers(CustomerID, Name, Address, City)
Orders(OrderID, CustomerID, Amount, Date)
Books(ISBN, Author, Title, Price)
Order_Items(OrderID, ISBN, Quantity)
Book_Reviews(ISBN, Reviews)
```

Ingat bahwa kunci utama digarisbawahi dan kunci asing memiliki garis bawah putus-putus. Untuk menggunakan materi di bagian ini, Anda harus memiliki akses ke MySQL. Ini biasanya berarti bahwa Anda

1. Telah menyelesaikan instalasi dasar MySQL di server Web Anda. Ini termasuk
   - Menginstal file
   - Menyiapkan pengguna untuk menjalankan MySQL sebagai
   - Menyiapkan jalur Anda
   - Menjalankan mysql_install_db, jika diperlukan
   - Mengatur kata sandi root
   - Menghapus pengguna anonim
   - Memulai server MySQL dan mengaturnya agar berjalan secara otomatis

   Jika Anda mengalami masalah di titik mana pun dalam bab ini, mungkin karena sistem MySQL Anda tidak diatur dengan benar. Jika itu terjadi, rujuk kembali daftar ini dan Lampiran A untuk memastikan bahwa pengaturan Anda sudah benar.
2. Memiliki akses ke MySQL pada mesin yang tidak Anda kelola seperti layanan hosting web, mesin di tempat kerja Anda, dan sebagainya.

   Jika demikian halnya, untuk mengerjakan contoh-contoh atau membuat basis data Anda sendiri, Anda perlu meminta administrator Anda menyiapkan pengguna dan basis data untuk Anda gunakan dan memberi tahu Anda nama pengguna, kata sandi, dan nama basis data yang telah mereka tetapkan untuk Anda.

Anda dapat melewati bagian-bagian bab ini yang menjelaskan cara menyiapkan pengguna dan basis data atau membacanya untuk menjelaskan dengan lebih baik apa yang Anda perlukan kepada administrator sistem Anda. Sebagai pengguna biasa, Anda tidak akan dapat menjalankan perintah untuk membuat pengguna dan basis data.

Semua contoh dalam bab ini dibuat dan diuji dengan MySQL versi 3.22.27. Beberapa versi MySQL sebelumnya memiliki fungsionalitas yang lebih sedikit. Anda harus menginstal atau memutakhirkan ke rilis stabil terkini pada saat membaca. Anda dapat mengunduh rilis terkini dari situs MySQL di http://mysql.com.

Catatan tentang Penggunaan MySQL Monitor

akan melihat bahwa contoh MySQL dalam bab ini dan bab berikutnya mengakhiri setiap perintah dengan titik koma (;). Ini memberi tahu MySQL untuk menjalankan perintah. Jika Anda menghilangkan titik koma, tidak akan terjadi apa-apa. Ini adalah masalah umum bagi pengguna baru.

Ini juga berarti Anda dapat memiliki baris baru di tengah perintah. Kami telah menggunakan ini untuk membuat contoh lebih mudah dibaca. Anda akan melihat di mana kami telah melakukan ini karena MySQL menyediakan simbol kelanjutan. Ini adalah anak panah yang terlihat seperti ini:

```
mysql> grant select
->
```

Ini berarti MySQL mengharapkan lebih banyak masukan. Sampai Anda mengetik titik koma, Anda akan mendapatkan karakter ini setiap kali Anda menekan Enter.

Hal lain yang perlu diperhatikan adalah bahwa pernyataan SQL tidak peka huruf besar/kecil, tetapi nama database dan tabel dapat peka huruf besar/kecil—lebih lanjut tentang ini nanti.

2.1 CARA MASUK KE MYSQL

Untuk melakukannya, buka antarmuka baris perintah di komputer Anda dan ketik yang berikut:

```
> mysql -h hostname -u username -p
```

Prompt perintah Anda mungkin terlihat berbeda tergantung pada sistem operasi dan shell yang Anda gunakan.

Perintah mysql memanggil monitor MySQL. Ini adalah klien baris perintah yang menghubungkan Anda ke server MySQL.

Tombol -h digunakan untuk menentukan host yang ingin Anda hubungkan; yaitu, mesin tempat server MySQL berjalan. Jika Anda menjalankan perintah ini pada mesin yang sama dengan server MySQL, Anda dapat mengabaikan tombol ini dan parameter nama host. Jika tidak, Anda harus mengganti parameter nama host dengan nama mesin tempat server MySQL berjalan.

Tombol -u digunakan untuk menentukan nama pengguna yang ingin Anda hubungkan. Jika Anda tidak menentukan, defaultnya adalah nama pengguna yang Anda gunakan untuk masuk ke sistem operasi.

Jika Anda telah menginstal MySQL pada mesin atau server Anda sendiri, Anda harus masuk sebagai root dan membuat basis data yang akan kita gunakan di bagian ini. Dengan asumsi bahwa Anda telah melakukan instalasi bersih, root adalah satu-satunya pengguna yang harus Anda gunakan untuk memulai.

Jika Anda menggunakan MySQL pada mesin yang dikelola oleh orang lain, gunakan nama pengguna yang mereka berikan kepada Anda. Sakelar -p memberi tahu server bahwa Anda ingin terhubung menggunakan kata sandi. Anda dapat mengabaikannya jika kata sandi belum ditetapkan untuk pengguna yang Anda gunakan untuk masuk.

Jika Anda masuk sebagai root dan belum menetapkan kata sandi untuk root, saya sangat menyarankan Anda untuk mengunjungi Lampiran A dan melakukannya sekarang juga. Tanpa kata sandi root, sistem Anda tidak aman.

Anda tidak perlu menyertakan kata sandi pada baris ini. Server MySQL akan menanyakannya kepada Anda. Bahkan, lebih baik jika Anda tidak melakukannya. Jika Anda memasukkan kata sandi pada baris perintah, kata sandi tersebut akan muncul sebagai teks biasa di layar, dan akan cukup mudah ditemukan oleh pengguna lain.

Setelah Anda memasukkan perintah sebelumnya, Anda akan mendapatkan respons seperti ini:

```
Masukkan kata sandi: ****
```

(Jika ini tidak berhasil, verifikasi bahwa server MySQL sedang berjalan, dan perintah mysql ada di suatu tempat di jalur Anda.)

Anda harus memasukkan kata sandi Anda. Jika semuanya berjalan dengan baik, Anda akan melihat respons seperti ini:

```
Welcome to the MySQL monitor. Commands end with ; or \g.
Your MySQL connection id is 9 to server version: 3.22.34-shareware-debug
Type ‘help’ for help.
mysql>
```

Di komputer Anda sendiri: Jika Anda tidak mendapatkan respons seperti ini, pastikan Anda telah menjalankan `mysql_install_db` jika diperlukan, Anda telah menetapkan kata sandi root, dan Anda telah mengetiknya dengan benar.

Jika itu bukan komputer Anda, pastikan Anda mengetikkan kata sandi dengan benar. Anda sekarang akan berada di prompt perintah MySQL, siap untuk membuat basis data. Jika Anda menggunakan komputer Anda sendiri, ikuti panduan di bagian berikutnya.

Jika Anda menggunakan komputer orang lain, ini seharusnya sudah dilakukan untuk Anda. Anda dapat langsung ke bagian "Menggunakan Basis Data yang Tepat". Anda mungkin ingin membaca bagian-bagian di antaranya untuk latar belakang umum, tetapi Anda tidak akan

dapat menjalankan perintah yang ditentukan di sana. (Atau setidaknya Anda seharusnya tidak dapat melakukannya!)

2.2 MEMBUAT BASIS DATA DAN PENGGUNA

Sistem basis data MySQL dapat mendukung banyak basis data yang berbeda. Anda biasanya akan memiliki satu basis data per aplikasi. Dalam contoh Book-o-Rama kami, basis data akan disebut buku.

Membuat Basis Data

Ini adalah bagian yang termudah. Pada prompt perintah MySQL, ketik

```
mysql> create database dbname;
```

Anda harus mengganti nama basis data yang ingin Anda buat dengan dbname. Untuk mulai membuat contoh Book-O-Rama, Anda dapat membuat basis data bernama books.

Itu saja. Anda akan melihat respons seperti

```
Query OK, 1 row affecting (0.06 sec)
```

Ini berarti semuanya telah berhasil. Jika Anda tidak mendapatkan respons ini, pastikan Anda mengetik titik koma di akhir baris. Titik koma memberi tahu MySQL bahwa Anda telah selesai, dan MySQL seharusnya benar-benar menjalankan perintah tersebut.

Pengguna dan Hak Istimewa

Sistem MySQL dapat memiliki banyak pengguna. Pengguna root umumnya hanya boleh digunakan untuk tujuan administrasi, demi alasan keamanan. Untuk setiap pengguna yang perlu menggunakan sistem, Anda perlu menyiapkan akun dan kata sandi.

Akun dan kata sandi ini tidak harus sama dengan nama pengguna dan kata sandi di luar MySQL (misalnya, nama pengguna dan kata sandi UNIX atau NT). Prinsip yang sama berlaku untuk root. Sebaiknya Anda memiliki kata sandi yang berbeda untuk sistem dan MySQL, terutama untuk kata sandi root.

Tidak wajib untuk menyiapkan kata sandi bagi pengguna, tetapi kami sangat menyarankan agar Anda menyiapkan kata sandi untuk semua pengguna yang Anda buat. Untuk keperluan menyiapkan basis data Web, sebaiknya Anda menyiapkan setidaknya satu pengguna per aplikasi Web.

Anda mungkin bertanya, "Mengapa saya ingin melakukan ini?"—jawabannya terletak pada hak istimewa.

Pengantar Sistem Hak Istimewa MySQL

Salah satu fitur terbaik MySQL adalah dukungannya terhadap sistem hak istimewa yang canggih. Hak istimewa adalah hak untuk melakukan tindakan tertentu pada objek tertentu, dan dikaitkan dengan pengguna tertentu. Konsepnya sangat mirip dengan izin berkas.

Saat Anda membuat pengguna dalam MySQL, Anda memberinya serangkaian hak istimewa untuk menentukan apa yang dapat dan tidak dapat dilakukannya dalam sistem.

Prinsip Hak Istimewa Terkecil

Prinsip hak istimewa terkecil dapat digunakan untuk meningkatkan keamanan sistem komputer mana pun. Ini adalah prinsip dasar, tetapi sangat penting yang sering kali diabaikan. Prinsipnya adalah sebagai berikut:

> *Seorang pengguna (atau proses) harus memiliki tingkat hak istimewa terendah yang diperlukan untuk melakukan tugas yang diberikan kepadanya.*

Prinsip ini berlaku di MySQL seperti di tempat lain. Misalnya, untuk menjalankan kueri dari Web, pengguna tidak memerlukan semua hak istimewa yang dapat diakses oleh pengguna root. Oleh karena itu, kita harus membuat pengguna lain yang hanya memiliki hak istimewa yang diperlukan untuk mengakses basis data yang baru saja kita buat.

Menyiapkan Pengguna: Perintah GRANT

Perintah GRANT dan REVOKE digunakan untuk memberikan dan mencabut hak kepada dan dari pengguna MySQL pada empat tingkat hak istimewa. Tingkat-tingkat ini adalah

- Global
- Basis Data
- Tabel
- Kolom

Kita akan melihat sebentar lagi bagaimana masing-masing tingkat ini dapat diterapkan.
Perintah GRANT digunakan untuk membuat pengguna dan memberi mereka hak istimewa. Bentuk umum dari perintah GRANT adalah

```
GRANT privileges [columns]
ON item
TO user_name [IDENTIFIED BY 'password']
[WITH GRANT OPTION]
```

Klausa dalam tanda kurung siku bersifat opsional. Ada sejumlah placeholder dalam sintaksis ini.

Yang pertama, hak istimewa, harus berupa daftar hak istimewa yang dipisahkan dengan koma. MySQL memiliki serangkaian hak istimewa yang telah ditetapkan. Hak istimewa tersebut dijelaskan di bagian berikutnya.

Placeholder kolom bersifat opsional. Anda dapat menggunakannya untuk menentukan hak istimewa berdasarkan kolom per kolom. Anda dapat menggunakan nama kolom tunggal atau daftar nama kolom yang dipisahkan dengan koma.

Placeholder item adalah basis data atau tabel tempat hak istimewa baru diterapkan. Anda dapat memberikan hak istimewa pada semua basis data dengan menentukan *.* sebagai item. Ini disebut pemberian hak istimewa global. Anda juga dapat melakukannya dengan menentukan * saja jika Anda tidak menggunakan basis data tertentu.

Lebih umum, Anda akan menentukan semua tabel dalam basis data sebagai dbname.*, pada tabel tunggal sebagai `dbname.tablename`, atau pada kolom tertentu dengan

menentukan dbname.tablename dan beberapa kolom tertentu di placeholder kolom. Ini mewakili tiga tingkat hak istimewa lain yang tersedia: basis data, tabel, dan kolom. Jika Anda menggunakan basis data tertentu saat Anda mengeluarkan perintah ini, namatabel itu sendiri akan ditafsirkan sebagai tabel dalam basis data saat ini.

User_name harus berupa nama yang Anda inginkan agar pengguna masuk seperti di MySQL. Ingatlah bahwa nama tersebut tidak harus sama dengan nama login sistem. User_name di MySQL juga dapat berisi nama host. Anda dapat menggunakan ini untuk membedakan antara, misalnya, laura (diartikan sebagai laura@localhost) dan laura@somewhere.com. Ini cukup berguna karena pengguna dari domain yang berbeda sering kali memiliki nama yang sama. Ini juga meningkatkan keamanan karena Anda dapat menentukan dari mana pengguna dapat terhubung, dan bahkan tabel atau basis data mana yang dapat mereka akses dari lokasi tertentu.

Kata sandi harus berupa kata sandi yang Anda inginkan agar pengguna masuk. Aturan umum untuk memilih kata sandi berlaku. Kita akan membahas lebih lanjut tentang keamanan nanti, tetapi kata sandi tidak boleh mudah ditebak. Ini berarti bahwa kata sandi tidak boleh berupa kata kamus atau sama dengan nama pengguna.

Idealnya, kata sandi akan berisi campuran huruf besar dan kecil serta karakter nonalfabetik. Opsi WITH GRANT OPTION, jika ditentukan, memungkinkan pengguna yang ditentukan untuk memberikan hak istimewanya sendiri kepada orang lain.

Hak istimewa disimpan dalam empat tabel sistem, dalam database yang disebut mysql. Keempat tabel ini disebut mysql.user, mysql.db, mysql.tables_priv, dan mysql.columns_priv; mereka berhubungan langsung dengan empat tingkat hak istimewa yang disebutkan sebelumnya. Sebagai alternatif untuk GRANT, Anda dapat mengubah tabel ini secara langsung.

2.3 JENIS DAN TINGKATAN HAK ISTIMEWA

Ada tiga jenis hak istimewa dasar di MySQL: hak istimewa yang sesuai untuk diberikan kepada pengguna biasa, hak istimewa yang sesuai untuk administrator, dan beberapa hak istimewa khusus. Setiap pengguna dapat diberikan salah satu dari hak istimewa ini, tetapi biasanya masuk akal untuk membatasi hak istimewa jenis administrator hanya untuk administrator, sesuai dengan prinsip hak istimewa paling sedikit.

Anda harus memberikan hak istimewa kepada pengguna hanya untuk basis data dan tabel yang perlu mereka gunakan. Anda tidak boleh memberikan akses ke basis data mysql kepada siapa pun kecuali administrator. Di sinilah semua pengguna, kata sandi, dan sebagainya disimpan.

Hak istimewa untuk pengguna biasa secara langsung berhubungan dengan jenis perintah SQL tertentu dan apakah pengguna diizinkan untuk menjalankannya. Kita akan membahas perintah SQL ini secara terperinci di bab berikutnya. Untuk saat ini, kami telah memberikan deskripsi konseptual tentang apa yang mereka lakukan. Hak istimewa ini ditunjukkan pada Tabel 2.1. Item di bawah kolom Berlaku Untuk mencantumkan objek yang dapat diberikan hak istimewa jenis ini.

Tabel 2.1 Hak Istimewa bagi Pengguna

Hak Istimewa	*Berlaku untuk*	*Keterangan*
SELECT	Tabel, kolom	Memungkinkan pengguna untuk memilih baris (catatan) dari tabel.
INSERT	Tabel, kolom	Memungkinkan pengguna untuk menyisipkan baris baru ke dalam tabel.
UPDATE	Tabel, kolom	Memungkinkan pengguna untuk mengubah nilai dalam baris tabel yang ada.
DELETE	Tabel	Memungkinkan pengguna untuk menghapus baris tabel yang ada.
INDEX	Tabel	Memungkinkan pengguna untuk membuat dan menghapus indeks pada tabel tertentu.
ALTER	Tabel	Memungkinkan pengguna untuk mengubah struktur tabel yang ada, misalnya, menambahkan kolom, mengganti nama kolom atau tabel, dan mengubah tipe data kolom.
CREATE	Tabel, Database	Memungkinkan pengguna membuat basis data atau tabel baru. Jika basis data atau tabel tertentu ditetapkan dalam GRANT, mereka hanya dapat CREATE basis data atau tabel tersebut, yang berarti mereka harus DROP nya terlebih dahulu.
DROP	Tabel, Database	Memungkinkan pengguna untuk menghapus (dropping) basis data atau tabel.

Sebagian besar hak istimewa untuk pengguna biasa relatif tidak berbahaya dalam hal keamanan sistem. Hak istimewa ALTER dapat digunakan untuk mengakali sistem hak istimewa dengan mengganti nama tabel, tetapi hak istimewa ini sangat dibutuhkan oleh pengguna. Keamanan selalu menjadi pilihan antara kegunaan dan keselamatan. Anda harus membuat keputusan sendiri terkait ALTER, tetapi hak istimewa ini sering diberikan kepada pengguna.

Selain hak istimewa yang tercantum dalam Tabel 2.1, ada hak istimewa REFERENCES yang saat ini tidak digunakan, dan ada hak istimewa GRANT yang diberikan dengan WITH GRANT OPTION dan bukan dalam daftar hak istimewa. Tabel 2.2 menunjukkan hak istimewa yang sesuai untuk digunakan oleh pengguna administratif.

Tabel 2.2 Hak Istimewa untuk Administrator

Previlege	*Deskripsi*
RELOAD	Memungkinkan administrator memuat ulang tabel hibah dan membersihkan hak istimewa, host, log, dan tabel.
SHUTDOWN	Memungkinkan administrator untuk mematikan server MySQL.

PROCESS	Memungkinkan administrator untuk melihat proses server dan menghentikannya
FILE	Memungkinkan data dibaca ke dalam tabel dari file, dan sebaliknya.

Hak istimewa ini dapat diberikan kepada nonadministrator, tetapi kehati-hatian yang ekstrem harus digunakan jika Anda mempertimbangkan untuk melakukannya. Pengguna rata-rata seharusnya tidak perlu menggunakan hak istimewa RELOAD, SHUTDOWN, dan PROCESS.

Hak istimewa FILE sedikit berbeda. Hak istimewa ini berguna bagi pengguna karena memuat data dari file dapat menghemat banyak waktu untuk memasukkan kembali data setiap kali memasukkannya ke dalam basis data. Namun, pemuatan file dapat digunakan untuk memuat file apa pun yang dapat dilihat oleh server MySQL, termasuk basis data milik pengguna lain dan, mungkin, file kata sandi. Berikan hak istimewa ini dengan hati-hati, atau tawarkan untuk memuat data bagi pengguna. Ada dua hak istimewa khusus, dan hak istimewa ini ditunjukkan pada Tabel 2.3.

Tabel 2.3 Hak Istimewa Khusus

Hak Istimewa	***Deskripsi***
All	Memberikan semua hak istimewa yang tercantum dalam Tabel 3.1 dan 3.2. Anda juga dapat menulis ALL PRIVILEGES sebagai ganti ALL.
USAGE	Tidak memberikan hak istimewa. Ini akan membuat pengguna dan mengizinkannya untuk masuk, tetapi tidak akan mengizinkannya untuk melakukan apa pun. Biasanya Anda akan menambahkan lebih banyak hak istimewa nanti.

Perintah REVOKE

Kebalikan dari GRANT adalah REVOKE. Perintah ini digunakan untuk mencabut hak istimewa dari pengguna. Sintaks perintah ini sangat mirip dengan GRANT:

```
REVOKE privileges [(columns)]
ON item
FROM user_name
```

Jika Anda telah memberikan klausul WITH GRANT OPTION, Anda dapat mencabutnya dengan melakukan:

```
REVOKE GRANT OPTION
ON item
FROM user_name
```

Contoh Penggunaan GRANT dan REVOKE

Untuk mengatur administrator, Anda dapat mengetik

```
mysql> grant all
    -> on *
    -> to fred identified by ‘mnb123’
    -> with grant option;
```

Ini memberikan semua hak istimewa pada semua basis data kepada pengguna bernama Fred dengan kata sandi mnb123, dan memungkinkannya untuk meneruskan hak istimewa tersebut.

Kemungkinannya Anda tidak menginginkan pengguna ini di sistem Anda, jadi lanjutkan dan cabut hak istimewanya:

```
mysql> revoke all
     -> on *
     -> from fred;
```

Sekarang mari kita atur pengguna biasa tanpa hak istimewa:

```
mysql> grant usage
     -> on books.*
     -> to sally identified by ‘magic123’;
```

Setelah berbicara dengan Sally, kami mengetahui lebih banyak tentang apa yang ingin dia lakukan, sehingga kami dapat memberinya hak istimewa yang sesuai:

```
mysql> grant select, insert, update, delete, index, alter, create, drop
     -> on books.*
     -> to sally;
```

Perhatikan bahwa kita tidak perlu menentukan kata sandi Sally untuk melakukan hal ini.
Jika kita memutuskan bahwa Sally telah melakukan sesuatu di basis data, kita mungkin memutuskan untuk mengurangi hak istimewanya:

```
mysql> revoke alter, create, drop
     -> on books.*
     -> from sally;
```

Dan nanti, ketika dia tidak perlu lagi menggunakan database tersebut, kita dapat mencabut hak istimewanya secara keseluruhan:

```
mysql> revoke all
     -> on books.*
     -> from sally;
```

Menyiapkan Pengguna Untuk Web

Anda perlu menyiapkan pengguna untuk skrip PHP Anda agar dapat terhubung ke MySQL. Sekali lagi, kita dapat menerapkan hak istimewa prinsip paling rendah: Apa yang seharusnya dapat dilakukan skrip?

Dalam kebanyakan kasus, skrip hanya perlu `SELECT`, `INSERT`, `DELETE`, dan `UPDATE` baris dari tabel. Anda dapat menyiapkannya sebagai berikut:

```
mysql> grant select, insert, delete, update
    -> on books.*
    -> to bookorama identified by 'bookorama123';
```

Tentu saja, demi alasan keamanan, Anda harus memilih kata sandi yang lebih baik daripada ini. Jika Anda menggunakan layanan hosting web, Anda biasanya akan mendapatkan akses ke hak istimewa tipe pengguna lain pada basis data yang mereka buat untuk Anda. Mereka biasanya akan memberi Anda nama pengguna dan kata sandi yang sama untuk penggunaan baris perintah (menyiapkan tabel dan sebagainya) dan untuk koneksi skrip web (menanyakan basis data). Ini sedikit kurang aman. Anda dapat menyiapkan pengguna dengan tingkat hak istimewa ini sebagai berikut:

```
mysql> grant select, insert, update, delete, index, alter, create, drop
    -> on books.*
    -> to bookorama identified by 'bookorama123';
```

Lanjutkan dan atur pengguna kedua ini.

Keluar Sebagai root

Anda dapat keluar dari monitor MySQL dengan mengetik quit. Anda harus masuk kembali sebagai pengguna Web untuk menguji apakah semuanya berfungsi dengan benar.

2.4 MENGGUNAKAN BASIS DATA YANG TEPAT

Jika Anda telah mencapai tahap ini, Anda seharusnya masuk ke akun MySQL tingkat pengguna yang siap menguji kode contoh, baik karena Anda baru saja mengaturnya, atau karena administrator server Web telah mengaturnya untuk Anda.

Hal pertama yang perlu Anda lakukan saat masuk adalah menentukan basis data mana yang ingin Anda gunakan. Anda dapat melakukannya dengan mengetik

```
mysql> use dbname;
```

di mana dbname adalah nama basis data Anda.

Sebagai alternatif, Anda dapat menghindari perintah use dengan menentukan basis data saat masuk, sebagai berikut:

```
mysql dbname -h hostname -u username -p
```

Dalam contoh ini, kita akan menggunakan basis data buku:

```
mysql> use books;
```

Saat Anda mengetik perintah ini, MySQL akan memberi Anda respons seperti

```
Database changed
```

Jika Anda tidak memilih basis data sebelum mulai bekerja, MySQL akan memberi Anda pesan kesalahan seperti

```
ERROR 1046: Tidak Ada Basis Data yang Dipilih
```

Membuat Tabel Basis Data

Langkah berikutnya dalam menyiapkan basis data adalah membuat tabel. Anda dapat melakukannya menggunakan perintah SQL `CREATE TABLE`. Bentuk umum pernyataan `CREATE TABLE` adalah

```
CREATE TABLE nama_tabel(kolom)
```

Anda harus mengganti placeholder `nama_tabel` dengan nama tabel yang ingin Anda buat, dan placeholder kolom dengan daftar kolom dalam tabel yang dipisahkan koma.
Setiap kolom akan memiliki nama yang diikuti oleh tipe data. Berikut skema Book-O-Rama:

```
Customers(CustomerID, Name, Address, City)
Orders(OrderID, CustomerID, Amount, Date)
Books(ISBN, Author, Title, Price)
Order_Items(OrderID, ISBN, Quantity)
Book_Reviews(ISBN, Review)
```

Listing 2.1 menunjukkan SQL untuk membuat tabel ini, dengan asumsi Anda telah membuat basis data yang disebut buku.

Anda dapat menjalankan file SQL yang sudah ada, seperti yang dimuat dari CD-ROM, melalui MySQL dengan mengetik:

```
> mysql -h host -u bookorama books -p < bookorama.sql
```

Menggunakan pengalihan file cukup praktis untuk ini karena ini berarti Anda dapat mengedit SQL Anda di editor teks pilihan Anda sebelum menjalankannya.

Listing 2.1 bookorama.sql—SQL untuk Membuat Tabel untuk Book-O-Rama

```
create table customers
(  customerid int unsigned not null auto_increment primary key,
   name char(30) not null,
   address char(40) not null,
   city char(20) not null
);

create table orders
(  orderid int unsigned not null auto_increment primary key,
   customerid int unsigned not null,
   amount float(6,2),
   date date not null
);

create table books
(  isbn char(13) not null primary key,
   author char(30),
   title char(60),
   price float(4,2)
);

create table order_items
(  orderid int unsigned not null,
   isbn char(13) not null,
   quantity tinyint unsigned,
   primary key (orderid, isbn)
);

create table book_reviews
(  isbn char(13) not null primary key,
   review text
);
```

Setiap tabel dibuat dengan pernyataan `CREATE TABLE` yang terpisah. Anda melihat bahwa kami telah membuat setiap tabel dalam skema dengan kolom yang kami desain di bab terakhir. Anda akan melihat bahwa setiap kolom memiliki tipe data yang tercantum setelah namanya. Beberapa kolom juga memiliki penentu lainnya.

Apa Arti Kata Kunci Lainnya

`NOT NULL` berarti bahwa semua baris dalam tabel harus memiliki nilai dalam atribut ini. Jika tidak ditentukan, kolom dapat kosong (NULL).

`AUTO_INCREMENT` adalah fitur MySQL khusus yang dapat Anda gunakan pada kolom integer. Artinya jika kita membiarkan kolom tersebut kosong saat memasukkan baris ke dalam tabel, MySQL akan secara otomatis menghasilkan nilai pengenal unik. Nilai tersebut akan menjadi satu lebih besar dari nilai maksimum di kolom yang sudah ada. Anda hanya dapat

memiliki satu dari ini di setiap tabel. Kolom yang menentukan AUTO_INCREMENT harus diindeks.

PRIMARY KEY setelah nama kolom menentukan bahwa kolom ini adalah kunci utama untuk tabel tersebut. Entri dalam kolom ini harus unik. MySQL akan mengindeks kolom ini secara otomatis. Perhatikan bahwa saat kita menggunakannya di atas dengan customerid di tabel customers, kita menggunakannya dengan AUTO_INCREMENT. Indeks otomatis pada kunci utama menangani indeks yang dibutuhkan oleh AUTO_INCREMENT.

Menentukan PRIMARY KEY setelah nama kolom hanya dapat digunakan untuk kunci utama kolom tunggal. Klausa PRIMARY KEY di akhir pernyataan order_items adalah bentuk alternatif. Kita menggunakannya di sini karena kunci utama untuk tabel ini terdiri dari dua kolom bersama-sama.

UNSIGNED setelah tipe integer berarti hanya dapat memiliki nilai nol atau positif.

2.5 MEMAHAMI TIPE KOLOM

Mari kita ambil tabel pertama sebagai contoh:

```
create table customers
(  customerid int unsigned not null auto_increment primary key,
   name char(30) not null,
   address char(40) not null,
   city char(20) not null
);
```

Saat membuat tabel apa pun, Anda perlu membuat keputusan tentang jenis kolom.

Dengan tabel pelanggan, kita memiliki empat kolom sebagaimana ditentukan dalam skema kita. Yang pertama, customerid, adalah kunci utama, yang telah kita tentukan secara langsung. Kita telah memutuskan bahwa ini akan menjadi bilangan bulat (tipe data int) dan ID ini harus tidak bertanda. Kita juga telah memanfaatkan fasilitas auto_increment sehingga MySQL dapat mengelolanya untuk kita—satu hal yang tidak perlu dikhawatirkan berkurang.

Kolom lainnya akan menampung data tipe string. Kita telah memilih tipe char untuk ini. Ini menentukan bidang dengan lebar tetap. Lebar ditentukan dalam tanda kurung, jadi, misalnya, nama dapat memiliki hingga 30 karakter.

Tipe data ini akan selalu mengalokasikan 30 karakter penyimpanan untuk nama, meskipun tidak semuanya digunakan. MySQL akan mengisi data dengan spasi untuk membuatnya berukuran tepat. Alternatifnya adalah varchar, yang hanya menggunakan jumlah penyimpanan yang diperlukan (ditambah satu byte). Ini adalah pengorbanan kecil—varchars akan menggunakan lebih sedikit ruang tetapi chars lebih cepat.

Untuk pelanggan sungguhan dengan nama dan alamat sungguhan, lebar kolom ini akan terlalu sempit. Perhatikan bahwa kami telah mendeklarasikan semua kolom sebagai NOT NULL. Ini adalah pengoptimalan kecil yang dapat Anda lakukan sedapat mungkin yang juga akan membuat semuanya berjalan sedikit lebih cepat.

Beberapa pernyataan CREATE lainnya memiliki variasi sintaksis. Mari kita lihat tabel pesanan:

```
create table orders
(  orderid int unsigned not null auto_increment primary key,
   customerid int unsigned not null,
   amount float(6,2),
   date date not null
);
```

Kolom jumlah ditetapkan sebagai angka floating point bertipe float. Dengan sebagian besar tipe data floating point, Anda dapat menetapkan lebar tampilan dan jumlah tempat desimal. Dalam kasus ini, jumlah pesanan akan dalam dolar, jadi kami telah mengizinkan total pesanan yang cukup besar (lebar 6) dan dua tempat desimal untuk sen.

Kolom tanggal memiliki tipe data date.

Dalam tabel khusus ini, kami telah menetapkan bahwa semua kolom kecuali jumlah sebagai NOT NULL. Mengapa? Ketika pesanan dimasukkan ke dalam basis data, kami harus membuatnya dalam orders, menambahkan items ke order_items, lalu menghitung jumlahnya. Kami mungkin tidak mengetahui jumlahnya ketika pesanan dibuat, jadi kami mengizinkannya menjadi NULL.

Tabel books memiliki beberapa karakteristik serupa:

```
create table books
   ( isbn char(13) not null primary key,
   author char(30),
   title char(60),
   price float(4,2)
);
```

Dalam kasus ini, kita tidak perlu membuat kunci utama karena ISBN dibuat di tempat lain. Kita membiarkan kolom lainnya NULL karena toko buku mungkin mengetahui ISBN sebuah buku sebelum mereka mengetahui judul, penulis, atau harganya. Tabel order_items menunjukkan cara membuat kunci utama multikolom:

```
create table order_items
(  orderid int unsigned not null,
   isbn char(13) not null,
   quantity tinyint unsigned,
   primary key (orderid, isbn)
);
```

Kami telah menetapkan kuantitas buku tertentu sebagai `TINYINT UNSIGNED`, yang berisi bilangan bulat antara 0 dan 255.

Seperti yang telah kami sebutkan sebelumnya, kunci utama multikolom perlu ditetapkan dengan klausa kunci utama khusus. Klausa ini digunakan di sini.

Terakhir, jika Anda mempertimbangkan tabel `book_reviews`:

```
create table book_reviews
(
  isbn char(13) not null primary key,
  review text
);
```

Ini menggunakan tipe data baru, teks, yang belum kita bahas. Ini digunakan untuk teks yang lebih panjang, seperti artikel. Ada beberapa varian pada ini, yang akan kita bahas nanti di bab ini.

Untuk memahami pembuatan tabel secara lebih rinci, mari kita bahas nama kolom dan pengenal secara umum, lalu tipe data yang dapat kita pilih untuk kolom. Namun, pertama-tama, mari kita lihat basis data yang telah kita buat.

Melihat Basis Data Dengan SHOW Dan DESCRIBE

Masuk ke monitor MySQL dan gunakan basis data buku. Anda dapat melihat tabel dalam basis data dengan mengetik:

```
mysql> show tables;
```

MySQL akan menampilkan daftar semua tabel dalam basis data:

```
+-----------------+
| Tables in books |
+-----------------+
| book_reviews    |
| books           |
| customers       |
| order_items     |
| orders          |
+-----------------+
5 rows in set (0.06 sec)
```

Anda juga dapat menggunakan show untuk melihat daftar basis data dengan mengetik

```
mysql> show databases;
```

Anda dapat melihat informasi lebih lanjut tentang tabel tertentu, misalnya, buku, menggunakan DESCRIBE:

```
mysql> describe books;
```

MySQL akan menampilkan informasi yang Anda berikan saat membuat basis data:

```
+--------+------------+------+-----+---------+-------+
| Field  | Type       | Null | Key | Default | Extra |
+--------+------------+------+-----+---------+-------+
| isbn   | char(13)   |      | PRI |         |       |
| author | char(30)   | YES  |     | NULL    |       |
| title  | char(60)   | YES  |     | NULL    |       |
| price  | float(4,2) | YES  |     | NULL    |       |
+--------+------------+------+-----+---------+-------+
4 rows in set (0.05 sec)
```

Perintah-perintah ini berguna untuk mengingatkan Anda tentang jenis kolom, atau untuk menavigasi basis data yang tidak Anda buat.

Pengenal pada MySQL

Ada empat jenis pengenal di MySQL—basis data, tabel, dan kolom, yang sudah kita kenal, dan alias, yang akan kita bahas di bab berikutnya.

Basis data di MySQL dipetakan ke direktori dalam struktur file yang mendasarinya, dan tabel dipetakan ke file. Ini berdampak langsung pada nama yang dapat Anda berikan. Ini juga memengaruhi kepekaan huruf besar-kecil nama-nama ini—jika direktori dan nama file peka huruf besar-kecil di sistem operasi Anda, nama basis data dan tabel akan peka huruf besar-kecil (misalnya, di UNIX), jika tidak, tidak akan (misalnya, di Windows). Nama kolom dan nama alias tidak peka huruf besar-kecil, tetapi Anda tidak dapat menggunakan versi dengan huruf besar-kecil yang berbeda dalam pernyataan SQL yang sama.

Sebagai catatan tambahan, lokasi direktori dan file yang berisi data akan berada di tempat yang ditetapkan dalam konfigurasi. Anda dapat memeriksa lokasi di sistem Anda dengan menggunakan fasilitas mysqladmin sebagai berikut:

```
mysqladmin variables
```

Ringkasan kemungkinan pengenal ditunjukkan pada Tabel 2.4. Satu-satunya pengecualian tambahan adalah Anda tidak dapat menggunakan ASCII(0) atau ASCII(255) dalam pengenal (dan sejujurnya, saya tidak yakin mengapa Anda ingin melakukannya).

Tabel 2.4 Pengenal MySQL

Tipe	Panjang Max	Kasus Sensitif	Karakter yang diperbolehkan
Database	64	Sama seperti O/S	Apa pun yang diizinkan dalam nama direktori di O/S Anda kecuali karakter /

Table	64	Sama seperti O/S	Apa pun yang diizinkan dalam nama file di O/S Anda kecuali karakter / dan .
Kolom	64	Tidak	Apa pun
Alias	255	Tidak	Apa pun

Aturan-aturan ini sangat terbuka.

Sejak MySQL 3.23.6, Anda bahkan dapat memiliki kata-kata khusus dan karakter khusus dari semua jenis dalam pengenal, satu-satunya batasan adalah bahwa jika Anda menggunakan sesuatu yang aneh seperti ini, Anda harus meletakkannya dalam tanda kutip terbalik (terletak di bawah tombol tilde di kiri atas sebagian besar papan ketik). Misalnya

```
create database `create database`;
```

Aturan dalam versi MySQL (sebelum 3.23.6) lebih ketat, dan tidak memungkinkan Anda melakukan ini.

Tentu saja, Anda harus menerapkan akal sehat untuk semua kebebasan ini. Hanya karena Anda dapat memanggil basis data `` `create database` ``, bukan berarti Anda harus melakukannya. Prinsip yang sama berlaku seperti dalam jenis pemrograman lainnya—gunakan pengenal yang bermakna.

2.6 TIPE DATA KOLOM

Tiga tipe kolom dasar dalam MySQL adalah: numerik, tanggal dan waktu, dan string. Dalam setiap kategori ini terdapat sejumlah besar tipe. Masing-masing dari ketiga jenis tersebut hadir dalam berbagai ukuran penyimpanan. Saat memilih jenis kolom, prinsip umumnya adalah memilih jenis terkecil yang sesuai dengan data Anda.

Untuk banyak jenis data, saat Anda membuat kolom jenis tersebut, Anda dapat menentukan panjang tampilan maksimum. Hal ini ditunjukkan dalam tabel jenis data berikut sebagai M. Jika bersifat opsional untuk jenis tersebut, hal ini ditunjukkan dalam tanda kurung siku. Nilai maksimum yang dapat Anda tentukan untuk M adalah 255. Nilai opsional di seluruh deskripsi ini ditunjukkan dalam tanda kurung siku.

Tipe Numerik

Tipe numerik berupa bilangan bulat atau bilangan floating point. Untuk bilangan floating point, Anda dapat menentukan jumlah digit setelah tempat desimal. Ini ditunjukkan dalam buku ini sebagai D. Nilai maksimum yang dapat Anda tentukan untuk D adalah 30 atau M-2 (yaitu, panjang tampilan maksimum dikurangi dua—satu karakter untuk titik desimal dan satu untuk bagian integral dari bilangan), mana yang lebih rendah.

Tabel 2.5 Tipe Data Integral

Tipe	Jangkauan	Penyimpanan (Byte)	Deskripsi
`TINYINT[(M)]`	− 127 … 128 atau 0 … 255	1	Bilangan bulat sangat kecil

SMALLINT[(M)]	$-32768 \dots 32767$ atau $0 \dots 65535$	2	Bilangan bulat kecil
MEDIUMINT[(M)]	$-8388608 \dots 8388607$ atau $0 \dots 16777215$	3	Bilangan bulat berukuran sedang
INT[(M)]	$-2^{31} \dots 2^{31} - 1$ atau $0 \dots 2^{31} - 1$	4	Bilangan bulat biasa
INTEGER[(M)]			Sinonim INT
BIGINT[(M)]	$-2^{63} \dots 2^{63} - 1$ atau $0 \dots 2^{64} - 1$	8	Bilangan bulat besar

Untuk tipe integer, Anda juga dapat menentukan apakah Anda menginginkannya menjadi UNSIGNED, seperti yang ditunjukkan dalam Listing 2.1. Untuk semua tipe numerik, Anda juga dapat menentukan atribut ZEROFILL. Saat nilai dari kolom ZERO-FILL ditampilkan, nilai tersebut akan diberi awalan nol.

Tipe integral ditunjukkan dalam Tabel 2.5. Perhatikan bahwa rentang yang ditunjukkan dalam tabel ini menunjukkan rentang bertanda pada satu baris dan rentang tak bertanda pada baris berikutnya. Jenis titik mengambang ditunjukkan pada Tabel 2.6.

Tabel 2.6 Tipe Data Titik Float

Tipe	Jangkauan	Penyimpanan (Byte)	Deskripsi
FLOAT (Precision)		Bervariasi	Dapat digunakan untuk menentukan angka floating point presisi tunggal atau ganda.
FLOAT[(M,D)]	±1.175494351E-38 ±3.402823466E+38	4	Angka floating point presisi tunggal. Ini setara dengan FLOAT(4), tetapi dengan lebar tampilan dan jumlah tempat desimal yang ditentukan.
DOUBLE[(M,D)]	±1.7976931348623157E +308 ±2.2250738585072014E +308	8	Angka floating point presisi ganda. Ini setara dengan FLOAT(8) tetapi dengan lebar tampilan dan jumlah tempat desimal yang ditentukan.
DOUBLE PRECISION[(M,D)]	Seperti diatas		Sinonim Untuk DOUBLE[(M,D)]
REAL[(M,D)]	Seperti diatas		Sinonim Untuk DOUBLE[(M,D)]
DECIMAL[(M,[,D])]	Bervariasi	M+2	Angka floating point disimpan sebagai char. Rentangnya bergantung pada M, lebar tampilan.
NUMERIC[(M,D)]	Seperti diatas		Sinonim untuk Desimal

Tipe Tanggal dan Waktu

MySQL mendukung sejumlah tipe tanggal dan waktu. Tipe-tipe ini ditunjukkan pada Tabel 2.7. Dengan semua tipe ini, Anda dapat memasukkan data dalam format string atau

numerik. Perlu dicatat bahwa kolom TIMESTAMP pada baris tertentu akan ditetapkan ke tanggal dan waktu operasi terbaru pada baris tersebut jika Anda tidak menetapkannya secara manual. Ini berguna untuk pencatatan transaksi.

Tabel 2.7 Tipe Data Tanggal dan Waktu

Tipe	Jangkauan	Keterangan
DATE	1000-01-01 999-12-21	Tanggal. Akan ditampilkan sebagai YYYY-MM-DD.
TIME	-838:59:59 838:59:59	Waktu. Akan ditampilkan sebagai HH:MM:SS. Perhatikan bahwa rentangnya jauh lebih luas daripada yang mungkin ingin Anda gunakan.
DATETIME	1000-01-01 00:00:00 9999-12-31	Tanggal dan waktu. Akan ditampilkan sebagai YYYY-MM-DDHH:MM:SS.
TIMESTAMP[(M)]	1970-01-01 00:00:00 Kadang dalam Timestamps 2037	Stempel waktu, berguna untuk transaksi pelaporan. Format tampilan tergantung pada nilai M (lihat Tabel 2.8 berikut). Puncak rentang bergantung pada batasan pada UNIX.
YEAR[(2|4)]	70-69 (1970-2069) 1901-2155	Setahun. Anda dapat menentukan format 2 atau 4 digit. Masing-masing memiliki rentang yang berbeda, seperti yang ditunjukkan.

Tabel 2.8 menunjukkan kemungkinan jenis tampilan yang berbeda untuk TIMESTAMP.

Tabel 2.8 Jenis Tampilan TIMESTAMP

Tipe lebih Spesifik	Tampilan
TIMESTAMP	YYYYMMDDHHMMSS
TIMESTAMP(14)	YYYYMMDDHHMMSS
TIMESTAMP(12)	YYMMDDHHMMSS
TIMESTAMP(10)	YYMMDDHHMM
TIMESTAMP(8)	YYYYMMDD
TIMESTAMP(6)	YYMMDD
TIMESTAMP(4)	YYMM
TIMESTAMP(2)	YY

Tipe String

Tipe string terbagi menjadi tiga kelompok. Pertama, ada string biasa, yaitu potongan teks pendek. Tipe ini adalah CHAR (karakter dengan panjang tetap) dan VARCHAR (karakter dengan panjang variabel). Anda dapat menentukan lebar masing-masing. Kolom tipe CHAR

akan diberi spasi hingga lebar maksimum tanpa mempedulikan ukuran data, sedangkan kolom VARCHAR lebarnya bervariasi sesuai dengan data. (Perhatikan bahwa MySQL akan menghapus spasi di akhir dari CHAR saat diambil, dan dari VARCHAR saat disimpan.)

Kedua, ada tipe TEXT dan BLOB. Tipe ini tersedia dalam berbagai ukuran. Tipe ini masing-masing untuk teks yang lebih panjang atau data biner. BLOB adalah objek biner besar. Tipe ini dapat menampung apa pun yang Anda suka, misalnya, data gambar atau suara.

Tabel 2.9 Tipe String Biasa

Tipe	Jangkauan	Keterangan
[NATIONAL] CHAR (M) [BINARY]	1 – 255 Karakter	String dengan panjang tetap dengan panjang M, di mana M berada di antara 1 dan 255. Kata kunci NATIONAL menetapkan bahwa set karakter default harus digunakan. Ini adalah default di MySQL, tetapi disertakan karena merupakan bagian dari standar ANSI SQL. Kata kunci BINARY menetapkan bahwa data harus diperlakukan sebagai tidak peka huruf besar/kecil. (Defaultnya peka huruf besar/kecil.)
[NATIONAL] VARCHAR (M) [BINARY]	1 – 255 Karakter	Sama seperti di atas, kecuali panjangnya bervariasi.

Tabel 2.10 menunjukkan jenis TEXT dan BLOB. Panjang maksimum kolom TEXT dalam karakter adalah ukuran maksimum file dalam byte yang dapat disimpan dalam kolom tersebut.

Tabel 2.10 Jenis TEXT dan BLOB

Type	Panjang Maksimal (Karakter)	Keterangan
TINYBLOB	$2^8 – 1$ (255)	Bidang objek besar biner kecil (BLOB)
TINYTEXT	$2^8 – 1$ (255)	Bidang TEKS kecil
BLOB	$2^{16} – 1$ (65.535)	Bidang BLOB berukuran normal
TEXT	$2^{16} – 1$ (65.535)	Bidang TEKS berukuran normal
MEDIUMBLOB	$2^{24} – 1$ (16.777.215)	Bidang BLOB berukuran sedang
MEDIUMTEXT	$2^{24} – 1$ (16.777.215)	Bidang TEKS berukuran sedang
LONGBLOB	$2^{32} – 1$ (4.294.967.295)	Bidang BLOB yang panjang
LONGTEXT	$2^{32} – 1$ (4.294.967.295)	Bidang TEKS yang panjang

Dalam praktiknya, kolom BLOB dan TEXT sama saja, kecuali TEXT peka huruf besar/kecil dan BLOB tidak. Karena tipe kolom ini dapat menampung data dalam jumlah besar, tipe ini memerlukan beberapa pertimbangan khusus.

Kelompok ketiga memiliki dua tipe khusus, SET dan ENUM. Tipe SET digunakan untuk menentukan bahwa nilai dalam kolom ini harus berasal dari sekumpulan nilai tertentu yang ditentukan. Nilai kolom dapat berisi lebih dari satu nilai dari sekumpulan tersebut. Anda dapat memiliki maksimal 64 hal dalam sekumpulan yang ditentukan.

ENUM adalah enumerasi. Tipe ini sangat mirip dengan SET, kecuali bahwa kolom tipe ini hanya dapat memiliki satu dari nilai yang ditentukan atau NULL, dan Anda dapat memiliki maksimal 65535 hal dalam enumerasi.

Kami telah meringkas tipe data string dalam Tabel 2.9, 2.10, dan 2.11. Tabel 2.9 menunjukkan tipe string biasa. Tabel 2.11 menunjukkan tipe ENUM dan SET.

Tabel 2.11 Tipe SET dan ENUM

Tipe	Nilai Maksimum di Set	Keterangan
ENUM('value1','value2',...)	65535	Kolom jenis ini hanya dapat menampung satu nilai yang terdaftar atau NULL.
SET('value1','value2',...)	64	Kolom jenis ini dapat menampung sekumpulan nilai yang ditentukan atau NULL.

BAB 3
BEKERJA DENGAN BASIS DATA MYSQL

Dalam bab ini, kita akan membahas Bahasa Kueri Terstruktur (SQL) dan penggunaannya dalam kueri basis data. Kita akan terus mengembangkan basis data Book-O-Rama dengan melihat cara memasukkan, menghapus, dan memperbarui data, serta cara mengajukan pertanyaan ke basis data.

Topik yang akan kita bahas meliputi

- Apa itu SQL?
- Memasukkan data ke dalam basis data
- Mengambil data dari basis data
- Menggabungkan tabel
- Memperbarui catatan dari basis data
- Mengubah tabel setelah pembuatan
- Menghapus catatan dari basis data
- Menghapus tabel
- Kita akan mulai dengan membahas tentang apa itu SQL dan mengapa itu merupakan hal yang berguna untuk dipahami.

Jika Anda belum menyiapkan basis data Book-O-Rama, Anda harus melakukannya sebelum Anda dapat menjalankan kueri SQL dalam bab ini. Petunjuk untuk melakukan ini ada di Bab 2, “Membuat Basis Data Web.”

3.1 APA ITU SQL?

SQL adalah singkatan dari *Structured Query Language*. Bahasa ini adalah bahasa paling standar untuk mengakses sistem manajemen basis data relasional (RDBMS). SQL digunakan untuk menyimpan dan mengambil data ke dan dari basis data. SQL digunakan dalam sistem basis data seperti MySQL, Oracle, PostgreSQL, Sybase, dan Microsoft SQL Server.

Ada standar ANSI untuk SQL, dan sistem basis data seperti MySQL menerapkan standar ini. Mereka juga biasanya menambahkan beberapa fitur tambahan. Sistem hak istimewa di MySQL adalah salah satunya.

Anda mungkin pernah mendengar frasa *Data Definition Languages* (DDL), yang digunakan untuk mendefinisikan basis data, dan *Data Manipulation Languages (*DML), yang digunakan untuk mengkueri basis data. SQL mencakup kedua basis ini. Di Bab 8, kita melihat definisi data (DDL) dalam SQL, jadi kita sudah sedikit menggunakannya. Anda menggunakan DDL saat pertama kali menyiapkan basis data. Anda akan menggunakan aspek DML dari SQL lebih sering karena ini adalah bagian yang kami gunakan untuk menyimpan dan mengambil data nyata dalam basis data.

Memasukkan Data ke dalam Basis Data

Sebelum Anda dapat melakukan banyak hal dengan basis data, Anda perlu menyimpan beberapa data di dalamnya. Cara yang paling umum untuk melakukannya adalah dengan pernyataan SQL `INSERT`.

Ingat kembali bahwa RDBMS berisi tabel, yang pada gilirannya berisi baris data yang disusun menjadi kolom. Setiap baris dalam tabel biasanya menggambarkan beberapa objek atau hubungan dunia nyata, dan nilai kolom untuk baris tersebut menyimpan informasi tentang objek dunia nyata. Kita dapat menggunakan pernyataan `INSERT` untuk memasukkan baris data ke dalam basis data.

Bentuk umum pernyataan INSERT adalah

```
INSERT [INTO] table [(column1, column2, column3,...)] VALUES (value1,
value2, value3,...);
```

Misalnya, untuk memasukkan catatan ke dalam tabel Pelanggan Book-O-Rama, Anda bisa mengetik

```
insert into customers values
  (NULL, “Julie Smith”, “25 Oak Street”, “Airport West”);
```

Anda dapat melihat bahwa kami telah mengganti tabel dengan nama tabel sebenarnya tempat kami ingin meletakkan data, dan nilai dengan nilai tertentu. Semua nilai dalam contoh ini diapit tanda kutip ganda. String harus selalu diapit tanda kutip tunggal atau ganda di MySQL. (Kami akan menggunakan keduanya dalam buku ini.) Angka dan tanggal tidak memerlukan tanda kutip.

Ada beberapa hal menarik yang perlu diperhatikan tentang pernyataan `INSERT`.

Nilai yang kami tentukan akan digunakan untuk mengisi kolom tabel secara berurutan. Jika Anda ingin mengisi hanya beberapa kolom, atau jika Anda ingin menentukannya dalam urutan yang berbeda, Anda dapat mencantumkan kolom tertentu di bagian kolom pernyataan. Misalnya,

```
insert into customers (name, city) values
(“Melissa Jones”, “Nar Nar Goon North”);
```

Pendekatan ini berguna jika Anda hanya memiliki sebagian data tentang rekaman tertentu, atau jika beberapa kolom dalam rekaman bersifat opsional. Anda juga dapat memperoleh efek yang sama dengan sintaks berikut:

```
insert into customers
```

```
set name=”Michael Archer”,
  address=”12 Adderley Avenue”,
  city=”Leeton”;
```

Anda juga akan melihat bahwa kami menetapkan nilai NULL untuk kolom customerid saat menambahkan Julie Smith dan mengabaikan kolom tersebut saat menambahkan pelanggan lainnya. Anda mungkin ingat bahwa saat kami menyiapkan basis data, kami membuat customerid sebagai kunci utama untuk tabel Pelanggan, jadi ini mungkin tampak aneh. Namun, kami menetapkan kolom tersebut sebagai AUTOINCREMENT. Ini berarti bahwa, jika kami memasukkan baris dengan nilai NULL atau tanpa nilai di kolom ini, MySQL akan membuat nomor berikutnya dalam urutan autoincrement dan memasukkannya untuk kami secara otomatis. Ini cukup berguna.

Listing 3.1 book_insert.sql —SQL untuk Mengisi Tabel untuk Book-O-Rama

```
use books;

insert into customers values
  (NULL, “Julie Smith”, “25 Oak Street”, “Airport West”),
  (NULL, “Alan Wong”, “1/47 Haines Avenue”, “Box Hill”),
  (NULL, “Michelle Arthur”, “357 North Road”, “Yarraville”);

insert into orders values
  (NULL, 3, 69.98, “02-Apr-2000”),
  (NULL, 1, 49.99, “15-Apr-2000”),
  (NULL, 2, 74.98, “19-Apr-2000”),
  (NULL, 3, 24.99, “01-May-2000”);

insert into books values
  (“0-672-31697-8”, “Michael Morgan”, “Java 2 for Professional
    Developers”, 34.99),
  (“0-672-31745-1”, “Thomas Down”, “Installing Debian GNU/Linux”, 24.99),
  (“0-672-31509-2”, “Pruitt, et al.”, “Sams Teach Yourself GIMP in 24
    Hours”, 24.99),
  (“0-672-31769-9”, “Thomas Schenk”, “Caldera OpenLinux System
    Administration Unleashed”, 49.99);

insert into order_items values
  (1, “0-672-31697-8”, 2),
  (2, “0-672-31769-9”, 1),
  (3, “0-672-31769-9”, 1),
  (3, “0-672-31509-2”, 1),
  (4, “0-672-31745-1”, 3);

insert into book_reviews values
```

```
("0-672-31697-8", "Morgan's book is clearly written and goes well beyond
  most of the basic Java books out there.");
```

Anda juga dapat memasukkan beberapa baris ke dalam tabel sekaligus. Setiap baris harus berada dalam set tanda kurungnya sendiri, dan setiap set tanda kurung harus dipisahkan dengan koma.

Kami telah mengumpulkan beberapa contoh data sederhana untuk mengisi basis data. Ini hanyalah serangkaian pernyataan `INSERT` sederhana yang menggunakan pendekatan penyisipan multibaris ini. Hal ini ditunjukkan pada listing 3.1.
Anda dapat menjalankan skrip ini dengan menyalurkannya melalui MySQL sebagai berikut:

```
>mysql -h host -u bookorama -p < book_insert.sql
```

Mengambil Data dari Basis Data

Pekerja utama SQL adalah pernyataan `SELECT`. Pernyataan ini digunakan untuk mengambil data dari basis data dengan memilih baris yang sesuai dengan kriteria tertentu dari tabel. Ada banyak opsi dan cara berbeda untuk menggunakan pernyataan `SELECT`.

Bentuk dasar `SELECT` adalah

```
SELECT items
FROM tables
[ WHERE condition ]
[ GROUP BY group_type ]
[ HAVING where_definition ]
[ ORDER BY order_type ]
[ LIMIT limit_criteria ] ;
```

Kita akan membahas setiap klausa pernyataan tersebut. Namun, pertama-tama, mari kita lihat kueri tanpa klausa opsional, yang memilih beberapa item dari tabel tertentu.

Biasanya, item-item ini adalah kolom dari tabel. (Item-item ini juga bisa berupa hasil dari ekspresi MySQL apa pun. Kita akan membahas beberapa yang lebih berguna nanti di bagian ini.) Kueri ini mencantumkan konten kolom nama dan kota dari tabel Pelanggan:

```
select name, city
from customers;
```

Kueri ini memiliki keluaran berikut, dengan asumsi Anda telah memasukkan data sampel dari Listing 3.1:

```
+-----------------+--------------------+
| name            | city               |
+-----------------+--------------------+
| Julie Smith     | Airport West       |
| Alan Wong       | Box Hill           |
| Michelle Arthur | Yarraville         |
| Melissa Jones   | Nar Nar Goon North |
| Michael Archer  | Leeton             |
+-----------------+--------------------+
```

Seperti yang Anda lihat, kita memiliki tabel yang berisi item yang kita pilih—nama dan kota—dari tabel yang kita tentukan, Pelanggan. Data ini ditampilkan untuk semua baris dalam tabel Pelanggan.

Anda dapat menentukan kolom sebanyak yang Anda inginkan dari tabel dengan mencantumkannya setelah kata kunci select. Anda juga dapat menentukan beberapa item lainnya. Salah satu yang berguna adalah operator wildcard, *, yang cocok dengan semua kolom dalam tabel atau tabel yang ditentukan. Misalnya, untuk mengambil semua kolom dan semua baris dari tabel `order_items`, kita akan menggunakan

```
select *
from order_items;
```

yang akan memberikan output berikut:

```
+---------+---------------+----------+
| orderid | isbn          | quantity |
+---------+---------------+----------+
|       1 | 0-672-31697-8 |        2 |
|       2 | 0-672-31769-9 |        1 |
|       3 | 0-672-31769-9 |        1 |
|       3 | 0-672-31509-2 |        1 |
|       4 | 0-672-31745-1 |        3 |
+---------+---------------+----------+
```

Mengambil Data dengan Kriteria Tertentu

Untuk mengakses sebagian baris dalam tabel, kita perlu menentukan beberapa kriteria pemilihan. Anda dapat melakukannya dengan klausa `WHERE`. Misalnya,

```
select *
from orders
where customerid = 3;
```

akan memilih semua kolom dari tabel pesanan, tetapi hanya baris dengan `customerid` 3. Berikut outputnya:

```
+---------+------------+--------+------------+
| orderid | customerid | amount | date       |
+---------+------------+--------+------------+
|       1 |          3 |  69.98 | 0000-00-00 |
|       4 |          3 |  24.99 | 0000-00-00 |
+---------+------------+--------+------------+
```

Klausa `WHERE` menentukan kriteria yang digunakan untuk memilih baris tertentu. Dalam kasus ini, kami telah memilih baris dengan `customerid 3.` Tanda sama dengan tunggal digunakan untuk menguji kesetaraan—perhatikan bahwa ini berbeda dari PHP, dan mudah membingungkan saat Anda menggunakannya bersama-sama.

Selain kesetaraan, MySQL mendukung serangkaian operator dan ekspresi reguler yang lengkap. Yang paling sering Anda gunakan dalam klausa `WHERE` tercantum dalam Tabel 3.1. Perhatikan bahwa ini bukan daftar lengkap—jika Anda memerlukan sesuatu yang tidak tercantum di sini, periksa manual MySQL.

Tabel 3.1 Operator Perbandingan yang Berguna untuk Klausa WHERE

Operator	Nama (Jika diaplikasikan)	Contoh	Keterangan
=	Persamaan	customerid = 3	Menguji apakah dua nilai sama
>	Lebih Besar Dari	amount > 60.00	Menguji apakah satu nilai lebih besar dari nilai lainnya
<	Kurang dari	amount < 60.00	Menguji apakah satu nilai lebih kecil dari nilai lainnya
>=	Lebih besar atau sama dengan	amount >= 60.00	Menguji apakah satu nilai lebih besar atau sama dengan nilai lainnya
<=	Kurang dari atau sama dengan	amount <= 60.00	Menguji apakah satu nilai kurang dari atau sama dengan nilai lainnya
!= atau <>	Tidak sama	quality !=0	Menguji apakah dua nilai tidak sama
IS NOT NULL	Alamat tidak NOL		Menguji apakah bidang benar-benar berisi nilai
IS NULL	Alamatnya NOL		Menguji apakah bidang tidak berisi nilai
BETWEEN	Jumlah Antara 0 dan 60.00		Menguji apakah suatu nilai lebih besar atau sama dengan nilai minimum dan kurang dari atau sama dengan nilai maksimum
IN	Kota di “Semarang”, “Ungaran”		Menguji apakah suatu nilai berada dalam set tertentu
NOT IN	Kota bukan di “Semarang”, “Ungaran”		Menguji apakah suatu nilai tidak ada dalam satu set

LIKE	kecocokan pola	Name like (“Fred %”)	Memeriksa apakah suatu nilai cocok dengan suatu pola menggunakan pencocokan pola SQL sederhana
NOT LIKE	kecocokan pola	Name not like (“Fred %”)	Memeriksa apakah suatu nilai tidak cocok dengan suatu pola
REGEXP	Ekspresi Regular	name regexp	Memeriksa apakah suatu nilai cocok dengan ekspresi reguler

Tiga baris terakhir dalam tabel mengacu pada LIKE dan REGEXP. Keduanya merupakan bentuk pencocokan pola.

LIKE menggunakan pencocokan pola SQL sederhana. Pola dapat terdiri dari teks biasa ditambah karakter % (persen) untuk menunjukkan pencocokan karakter pengganti ke sejumlah karakter dan karakter _ (garis bawah) untuk pencocokan karakter pengganti ke satu karakter. Di MySQL, pencocokan ini tidak peka huruf besar-kecil. Misalnya, ‘Fred %’ akan cocok dengan nilai apa pun yang dimulai dengan ‘fred‘.

Kata kunci REGEXP digunakan untuk pencocokan ekspresi reguler. MySQL menggunakan ekspresi reguler POSIX. Alih-alih REGEXP, Anda juga dapat menggunakan RLIKE, yang merupakan sinonim. Ekspresi reguler POSIX juga digunakan dalam PHP.

Anda dapat menguji beberapa kriteria dengan cara ini dan menggabungkannya dengan AND dan OR. Misalnya,

```
select *
from orders
where customerid = 3 or customerid = 4;
```

Mengambil Data dari Beberapa Tabel

Sering kali, untuk menjawab pertanyaan dari basis data, Anda perlu menggunakan data dari lebih dari satu tabel. Misalnya, jika Anda ingin tahu pelanggan mana yang memesan bulan ini, Anda perlu melihat tabel Pelanggan dan tabel Pesanan. Jika Anda juga ingin tahu apa yang mereka pesan, Anda juga perlu melihat tabel item_Pesanan.

Item-item ini berada dalam tabel terpisah karena berhubungan dengan objek dunia nyata yang terpisah. Ini adalah salah satu prinsip desain basis data yang baik yang telah kita bahas di Bab 1, “Merancang Basis Data Web.”

Untuk menyatukan informasi ini dalam SQL, Anda harus melakukan operasi yang disebut gabungan. Ini berarti menggabungkan dua tabel atau lebih untuk mengikuti hubungan antara data. Misalnya, jika kita ingin melihat pesanan yang dilakukan pelanggan Julie Smith, kita perlu melihat tabel Pelanggan untuk menemukan ID Pelanggan Julie, lalu melihat tabel Pesanan untuk pesanan dengan ID Pelanggan tersebut.

Meskipun gabungan secara konseptual sederhana, gabungan merupakan salah satu bagian SQL yang lebih rumit dan rumit. Beberapa jenis join yang berbeda diimplementasikan dalam MySQL, dan masing-masing digunakan untuk tujuan yang berbeda.

3.2 TWO-TABLE JOINS SEDERHANA

Mari kita mulai dengan melihat beberapa SQL untuk query tentang Julie Smith yang baru saja kita bicarakan:

```
select orders.orderid, orders.amount, orders.date
from customers, orders
where customers.name = 'Julie Smith'
and customers.customerid = orders.customerid;
```

Output dari query ini adalah

```
+---------+--------+------------+
| orderid | amount | date       |
+---------+--------+------------+
|       2 |  49.99 | 0000-00-00 |
+---------+--------+------------+
```

Ada beberapa hal yang perlu diperhatikan di sini.

Pertama-tama, karena informasi dari dua tabel diperlukan untuk menjawab kueri ini, kami telah mencantumkan kedua tabel tersebut.

Kami juga telah menentukan jenis gabungan, mungkin tanpa mengetahuinya. Tanda koma di antara nama tabel setara dengan mengetik `INNER JOIN` atau `CROSS JOIN`. Ini adalah jenis gabungan yang terkadang juga disebut sebagai gabungan penuh, atau produk Cartesian dari tabel. Artinya, "Ambil tabel yang tercantum, dan buat satu tabel besar. Tabel besar tersebut harus memiliki baris untuk setiap kemungkinan kombinasi baris dari setiap tabel yang tercantum, entah itu masuk akal atau tidak." Dengan kata lain, kami memperoleh tabel, yang memiliki setiap baris dari tabel Pelanggan yang cocok dengan setiap baris dari tabel Pesanan, terlepas dari apakah pelanggan tertentu melakukan pemesanan tertentu.

Itu tidak masuk akal dalam banyak kasus. Sering kali yang kami inginkan adalah melihat baris yang benar-benar cocok, yaitu, pesanan yang dilakukan oleh pelanggan tertentu yang cocok dengan pelanggan tersebut. Kita mencapainya dengan menempatkan kondisi gabungan dalam klausa `WHERE`. Ini adalah jenis pernyataan kondisional khusus yang menjelaskan atribut mana yang menunjukkan hubungan antara dua tabel. Dalam kasus ini, kondisi gabungan kita adalah

```
customers.customerid = orders.customerid
```

yang memberi tahu MySQL untuk hanya meletakkan baris dalam tabel hasil jika `CustomerId` dari tabel `Customers` cocok dengan `CustomerID` dari tabel `Orders`.

Dengan menambahkan kondisi gabungan ini ke kueri, kita sebenarnya telah mengubah gabungan ke jenis yang berbeda, yang disebut `equi-join`.

Anda juga akan melihat notasi titik yang kita gunakan untuk memperjelas tabel mana kolom tertentu berasal, yaitu, `customers.customerid` merujuk ke kolom customerid dari tabel `Customers`, dan `orders.customerid` merujuk ke kolom `customerid` dari tabel `Orders`. Notasi titik ini diperlukan jika nama kolom ambigu, yaitu, jika muncul di lebih dari satu tabel.

Sebagai ekstensi, notasi ini juga dapat digunakan untuk menghilangkan ambiguitas nama kolom dari basis data yang berbeda. Dalam contoh ini, kami telah menggunakan notasi `table.column`. Anda dapat menentukan basis data dengan notasi `database.table.column`, misalnya, untuk menguji kondisi seperti

```
books.orders.customerid = other_db.orders.customerid
```

Namun, Anda dapat menggunakan notasi titik untuk semua referensi kolom dalam kueri. Ini bisa menjadi ide yang bagus, terutama setelah kueri Anda mulai menjadi rumit. MySQL tidak memerlukannya, tetapi notasi ini membuat kueri Anda jauh lebih mudah dibaca dan dipelihara. Anda akan melihat bahwa kami telah mengikuti konvensi ini di sisa kueri sebelumnya, misalnya, dengan penggunaan kondisi

```
customers.name = 'Julie Smith'
```

Nama kolom hanya muncul di tabel `customers`, jadi kami tidak perlu menentukannya, tetapi notasi ini membuatnya lebih jelas.

Menggabungkan Lebih dari Dua Tabel

Menggabungkan lebih dari dua tabel tidak lebih sulit daripada penggabungan dua tabel. Sebagai aturan umum, Anda perlu menggabungkan tabel secara berpasangan dengan kondisi penggabungan. Anggap saja seperti mengikuti hubungan antara data dari tabel ke tabel ke tabel.

Misalnya, jika kita ingin mengetahui pelanggan mana yang telah memesan buku di Java (mungkin agar kita dapat mengirimkan informasi tentang buku Java baru), kita perlu melacak hubungan ini melalui beberapa tabel.

Kita perlu menemukan pelanggan yang telah memesan setidaknya satu buku yang menyertakan `order_item` yaitu buku tentang Java. Untuk berpindah dari tabel Pelanggan ke tabel Pesanan, kita dapat menggunakan `customerid` seperti yang kita lakukan sebelumnya. Untuk berpindah dari tabel Pesanan ke tabel `Item_Pesanan`, kita dapat menggunakan `orderid`. Untuk berpindah dari tabel `Item_Pesanan` ke buku tertentu di tabel Buku, kita dapat menggunakan ISBN. Setelah membuat semua tautan tersebut, kita dapat menguji buku dengan Java di judulnya, dan mengembalikan nama pelanggan yang membeli salah satu buku tersebut.

Mari kita lihat kueri yang melakukan semua hal tersebut:

```
select customers.name
from customers, orders, order_items, books
```

```
where customers.customerid = orders.customerid
and orders.orderid = order_items.orderid
and order_items.isbn = books.isbn
and books.title like '%Java%';
```

Kueri ini akan mengembalikan keluaran berikut:

```
+-----------------+
| name            |
+-----------------+
| Michelle Arthur |
+-----------------+
```

Perhatikan bahwa kami menelusuri data melalui empat tabel yang berbeda, dan untuk melakukannya dengan equi-join, kami memerlukan tiga kondisi join yang berbeda. Secara umum, Anda memerlukan satu kondisi join untuk setiap pasangan tabel yang ingin Anda gabungkan, dan oleh karena itu jumlah total kondisi join kurang satu dari jumlah total tabel yang ingin Anda gabungkan. Aturan praktis ini dapat berguna untuk men-debug kueri yang tidak berfungsi dengan baik. Centang kondisi join Anda dan pastikan Anda telah mengikuti alur dari apa yang Anda ketahui hingga apa yang ingin Anda ketahui.

3.3 MENEMUKAN BARIS YANG TIDAK COCOK

Jenis join utama lainnya yang akan Anda gunakan di MySQL adalah left join. Pada contoh sebelumnya, Anda akan melihat bahwa hanya baris yang terdapat kecocokan antara tabel yang disertakan. Terkadang, kami secara khusus menginginkan baris yang tidak terdapat kecocokan—misalnya, pelanggan yang belum pernah memesan, atau buku yang belum pernah dipesan.

Cara termudah untuk menjawab pertanyaan jenis ini di MySQL adalah dengan menggunakan `left join`. Gabungan kiri akan mencocokkan baris pada kondisi gabungan tertentu antara dua tabel. Jika tidak ada baris yang cocok di tabel kanan, baris yang berisi nilai `NULL` di kolom kanan akan ditambahkan ke hasil.
Mari kita lihat contohnya:

```
select customers.customerid, customers.name, orders.orderid
from customers left join orders
on customers.customerid = orders.customerid;
```

Kueri SQL ini menggunakan gabungan kiri untuk menggabungkan Pelanggan dengan Pesanan. Anda akan melihat bahwa gabungan kiri menggunakan sintaksis yang sedikit berbeda untuk kondisi gabungan—dalam kasus ini, kondisi gabungan masuk ke klausa ON khusus dari pernyataan SQL.

Hasil dari kueri ini adalah

```
+------------+-----------------+---------+
| customerid | name            | orderid |
+------------+-----------------+---------+
|          1 | Julie Smith     |       2 |
|          2 | Alan Wong       |       3 |
|          3 | Michelle Arthur |       1 |
|          3 | Michelle Arthur |       4 |
|          4 | Melissa Jones   |    NULL |
|          5 | Michael Archer  |    NULL |
+------------+-----------------+---------+
```

Output ini menunjukkan kepada kita bahwa tidak ada `orderid` yang cocok untuk pelanggan Melissa Jones dan Michael Archer karena orderid untuk pelanggan tersebut adalah `NULL`. Jika kita ingin melihat hanya pelanggan yang belum memesan apa pun, kita dapat melakukannya dengan memeriksa `NULL` tersebut di kolom kunci utama tabel kanan (dalam kasus ini orderid) karena itu tidak boleh `NULL` di baris mana pun:

```
select customers.customerid, customers.name
from customers left join orders
using (customerid)
where orders.orderid is null;
```

Hasilnya adalah

```
+------------+----------------+
| customerid | name           |
+------------+----------------+
|          4 | Melissa Jones  |
|          5 | Michael Archer |
+------------+----------------+
```

Anda juga akan melihat bahwa kami menggunakan sintaksis yang berbeda untuk kondisi gabungan dalam contoh ini. Gabungan kiri mendukung sintaksis ON yang kami gunakan dalam contoh pertama, atau sintaksis USING dalam contoh kedua. Perhatikan bahwa sintaksis USING tidak menentukan tabel tempat atribut gabungan berasal—untuk alasan ini, kolom dalam dua tabel harus memiliki nama yang sama jika Anda ingin menggunakan USING.

Menggunakan Nama Lain untuk Tabel: Alias

Sering kali berguna dan terkadang penting untuk dapat merujuk ke tabel dengan nama lain. Nama lain untuk tabel disebut alias. Anda dapat membuatnya di awal kueri lalu menggunakannya di seluruh bagian. Alias sering kali berguna sebagai singkatan. Pertimbangkan kueri besar yang kita lihat sebelumnya, yang ditulis ulang dengan alias:

```
select c.name
from customers as c, orders as o, order_items as oi, books as b
where c.customerid = o.customerid
and o.orderid = oi.orderid
and oi.isbn = b.isbn
and b.title like '%Java%';
```

Saat kita mendeklarasikan tabel yang akan kita gunakan, kita menambahkan klausa AS untuk mendeklarasikan alias untuk tabel tersebut. Kita juga dapat menggunakan alias untuk kolom, tetapi kita akan membahasnya lagi saat kita membahas fungsi agregat sebentar lagi.

Kita perlu menggunakan alias tabel saat kita ingin menggabungkan tabel dengan dirinya sendiri. Ini terdengar lebih sulit dan lebih sulit daripada yang sebenarnya. Ini berguna, misalnya, jika kita ingin menemukan baris dalam tabel yang sama yang memiliki nilai yang sama. Jika kita ingin menemukan pelanggan yang tinggal di kota yang sama—mungkin untuk membuat kelompok baca—kita dapat memberikan dua alias yang berbeda untuk tabel yang sama (Pelanggan):

```
select c1.name, c2.name, c1.city
from customers as c1, customers as c2
where c1.city = c2.city
and c1.name != c2.name;
```

Pada dasarnya, yang kita lakukan adalah berpura-pura bahwa tabel Pelanggan adalah dua tabel yang berbeda, `c1` dan `c2`, dan melakukan penggabungan pada kolom Kota. Anda akan melihat bahwa kita juga memerlukan kondisi kedua, `c1.name != c2.name`–ini untuk menghindari setiap pelanggan muncul sebagai pasangannya sendiri.

Ringkasan Gabungan

Berbagai jenis gabungan yang telah kita lihat dirangkum dalam Tabel 3.2. Ada beberapa yang lain, tetapi ini adalah yang utama yang akan Anda gunakan.

Tabel 3.2 Jenis Gabungan dalam MySQL

Nama	***Keterangan***
Cartesian Product	Semua kombinasi semua baris di semua tabel dalam gabungan. Digunakan dengan menentukan koma di antara nama tabel, dan tidak menentukan klausa `WHERE`.
Full Join	Sama seperti sebelumnya.
Cross Join	Sama seperti sebelumnya. Dapat juga digunakan dengan menentukan kata kunci `CROSS JOIN` di antara nama tabel yang akan digabungkan.
Inner Join	Secara semantik setara dengan koma. Dapat juga ditentukan menggunakan kata kunci `INNER JOIN`. Tanpa kondisi `WHERE`, setara dengan gabungan penuh. Biasanya, Anda akan menentukan kondisi `WHERE` untuk menjadikan ini gabungan dalam yang sebenarnya

Equi-join	Menggunakan ekspresi kondisional dengan tanda = untuk mencocokkan baris dari tabel yang berbeda dalam gabungan. Dalam SQL, ini adalah gabungan dengan klausa `WHERE`.
Left Joim	Mencoba mencocokkan baris di seluruh tabel dan mengisi baris yang tidak cocok dengan `NULL`. Digunakan dalam SQL dengan kata kunci `LEFT JOIN`. Digunakan untuk menemukan nilai yang hilang. Anda dapat menggunakan `RIGHT JOIN` dengan cara yang sama.

Mengambil Data Dalam Urutan Tertentu

Jika Anda ingin menampilkan baris yang diambil oleh kueri dalam urutan tertentu, Anda dapat menggunakan klausa `ORDER BY` dari pernyataan `SELECT`. Fitur ini berguna untuk menyajikan output dalam format yang dapat dibaca manusia.

Klausa `ORDER BY` digunakan untuk mengurutkan baris pada satu atau beberapa kolom yang tercantum dalam klausa `SELECT`. Misalnya,

```
select name, address
from customers
order by name;
```

Kueri ini akan mengembalikan nama dan alamat pelanggan dalam urutan abjad berdasarkan nama, seperti ini:

```
+-----------------+--------------------+
| name            | address            |
+-----------------+--------------------+
| Alan Wong       | 1/47 Haines Avenue |
| Julie Smith     | 25 Oak Street      |
| Melissa Jones   |                    |
| Michael Archer  | 12 Adderley Avenue |
| Michelle Arthur | 357 North Road     |
+-----------------+--------------------+
```

(Perhatikan bahwa dalam kasus ini, karena nama-nama tersebut dalam format nama depan, nama belakang, maka nama-nama tersebut diurutkan berdasarkan abjad berdasarkan nama depan. Jika Anda ingin mengurutkan berdasarkan nama belakang, Anda harus memilikinya sebagai dua kolom yang berbeda.)

Urutan default adalah menaik (a hingga z atau secara numerik ke atas). Anda dapat menentukan ini jika Anda suka menggunakan kata kunci `ASC`:

```
select name, address
from customers
order by name asc;
```

Anda juga dapat melakukannya dalam urutan sebaliknya dengan menggunakan kata kunci DESC (descending):

```
select name, address
from customers
order by name desc;
```

Anda dapat mengurutkan berdasarkan lebih dari satu kolom. Anda juga dapat menggunakan alias kolom atau bahkan nomor posisinya (misalnya, 3 adalah kolom ketiga dalam tabel) sebagai ganti nama.

3.4 PENGELOMPOKAN DAN PENGGABUNGAN DATA

Kita sering ingin mengetahui berapa banyak baris yang termasuk dalam satu set tertentu, atau nilai rata-rata beberapa kolom—misalnya, nilai dolar rata-rata per pesanan. MySQL memiliki serangkaian fungsi agregat yang berguna untuk menjawab jenis kueri ini. Fungsi agregat ini dapat diterapkan ke tabel secara keseluruhan, atau ke kelompok data dalam tabel. Yang paling umum digunakan tercantum dalam Tabel 3.3.

Tabel 3.3 Fungsi Agregat dalam MySQL

Nama	***Keterangan***
AVG(column)	Rata-rata nilai dalam kolom yang ditentukan.
COUNT(items)	Jika Anda menentukan kolom, ini akan memberi Anda jumlah nilai bukan NULL di kolom tersebut. Jika Anda menambahkan kata DISTINCT di depan nama kolom, Anda akan mendapatkan jumlah nilai berbeda di kolom tersebut saja. Jika Anda menentukan COUNT(*), Anda akan mendapatkan jumlah baris tanpa memperhitungkan nilai NULL.
MIN(column)	Nilai minimum pada kolom yang ditentukan.
MAX(column)	Nilai maksimum pada kolom yang ditentukan.
STD(column)	Simpangan baku nilai dalam kolom yang ditentukan.
STDDEV(column)	Sama dengan STD (kolom).
SUM(column)	Jumlah nilai dalam kolom yang ditentukan.

Mari kita lihat beberapa contoh, dimulai dengan contoh yang disebutkan sebelumnya. Kita dapat menghitung total rata-rata pesanan seperti ini:

```
select avg(amount)
from orders;
```

Outputnya akan seperti ini:

```
+-------------+
| avg(amount) |
+-------------+
|   54.985002 |
+-------------+
```

Untuk mendapatkan informasi yang lebih rinci, kita dapat menggunakan klausa GROUP BY. Hal ini memungkinkan kita untuk melihat total pesanan rata-rata berdasarkan kelompok—misalnya, berdasarkan nomor pelanggan. Hal ini akan memberi tahu kita pelanggan mana yang melakukan pemesanan terbesar:

```
select customerid, avg(amount)
from orders
group by customerid;
```

Bila Anda menggunakan klausa GROUP BY dengan fungsi agregat, hal itu sebenarnya mengubah perilaku fungsi tersebut. Daripada memberikan rata-rata jumlah pesanan di seluruh tabel, kueri ini akan memberikan jumlah pesanan rata-rata untuk setiap pelanggan (atau, lebih khusus lagi, untuk setiap customerid

```
+------------+-------------+
| customerid | avg(amount) |
+------------+-------------+
|          1 |   49.990002 |
|          2 |   74.980003 |
|          3 |   47.485002 |
+------------+-------------+
```

Satu hal yang perlu diperhatikan saat menggunakan fungsi pengelompokan dan agregat: Dalam ANSI SQL, jika Anda menggunakan fungsi agregat atau klausa GROUP BY, satu-satunya hal yang dapat muncul dalam klausa SELECT Anda adalah fungsi agregat dan kolom yang disebutkan dalam klausa GROUP BY. Selain itu, jika Anda ingin menggunakan kolom dalam klausa GROUP BY, kolom tersebut harus dicantumkan dalam klausa SELECT.

MySQL sebenarnya memberi Anda sedikit keleluasaan di sini. MySQL mendukung sintaksis yang diperluas, yang memungkinkan Anda untuk tidak menyertakan item dalam klausa SELECT jika Anda tidak benar-benar menginginkannya.

Selain mengelompokkan dan mengagregasi data, kita sebenarnya dapat menguji hasil agregat menggunakan klausa HAVING. Klausa ini muncul langsung setelah klausa GROUP BY dan seperti WHERE yang hanya berlaku untuk pengelompokan dan agregat.

Untuk memperluas contoh kita sebelumnya, jika kita ingin mengetahui pelanggan mana yang memiliki total pesanan rata-rata lebih dari Rp. 500.000, kita dapat menggunakan kueri berikut:

```
select customerid, avg(amount)
from orders
group by customerid
having avg(amount) > 50;
```

Perhatikan bahwa klausa HAVING berlaku untuk grup. Kueri ini akan mengembalikan output berikut:

```
+------------+-------------+
| customerid | avg(amount) |
+------------+-------------+
|          2 |   74.980003 |
+------------+-------------+
```

Memilih Baris yang Akan Dikembalikan

Salah satu klausa pernyataan SELECT yang dapat sangat berguna dalam aplikasi Web adalah klausa LIMIT. Klausa ini digunakan untuk menentukan baris mana dari output yang harus dikembalikan. Klausa ini memerlukan dua parameter: nomor baris tempat memulai dan jumlah baris yang akan dikembalikan. Kueri ini menggambarkan penggunaan LIMIT:

```
select name
from customers
limit 2, 3;
```

Kueri ini dapat dibaca sebagai, “Pilih nama dari pelanggan, lalu kembalikan 3 baris, dimulai dari baris 2 dalam output.” Perhatikan bahwa nomor baris diindeks nol—yaitu, baris pertama dalam output adalah baris nomor nol.

Hal ini sangat berguna untuk aplikasi Web, seperti saat pelanggan menelusuri produk dalam katalog, dan kami ingin menampilkan 10 item pada setiap halaman.

Memperbarui Catatan dalam Basis Data

Selain mengambil data dari basis data, kami sering ingin mengubahnya. Misalnya, kami mungkin ingin menaikkan harga buku dalam basis data. Kami dapat melakukannya menggunakan pernyataan UPDATE. Bentuk umum pernyataan UPDATE adalah

```
UPDATE tablename
SET column1=expression1,column2=expression2,...
[WHERE condition]
[LIMIT number]
```

Ide dasarnya adalah memperbarui tabel yang disebut tablename, dengan menetapkan setiap kolom yang diberi nama ke ekspresi yang sesuai. Anda dapat membatasi UPDATE ke baris tertentu dengan klausa WHERE, dan membatasi jumlah total baris yang akan terpengaruh dengan klausa LIMIT.

Mari kita lihat beberapa contoh.

Jika kita ingin menaikkan semua harga buku sebesar 10%, kita dapat menggunakan pernyataan UPDATE tanpa klausa WHERE:

```
update books
set price=price*1.1;
```

Sebaliknya, jika kita ingin mengubah satu baris—misalnya, untuk memperbarui alamat pelanggan—kita dapat melakukannya seperti ini:

```
update customers
set address = '250 Olsens Road'
where customerid = 4;
```

Mengubah Tabel Setelah Pembuatan

Selain memperbarui baris, Anda mungkin ingin mengubah struktur tabel dalam basis data Anda. Untuk tujuan ini, Anda dapat menggunakan pernyataan ALTER TABLE yang fleksibel. Bentuk dasar pernyataan ini adalah

```
ALTER TABLE tablename alteration [, alteration ...]
```

Perhatikan bahwa dalam ANSI SQL Anda hanya dapat membuat satu perubahan per pernyataan ALTER TABLE, tetapi MySQL memungkinkan Anda untuk membuat sebanyak yang Anda suka. Setiap klausa perubahan dapat digunakan untuk mengubah berbagai aspek tabel. Berbagai jenis perubahan yang dapat Anda buat dengan pernyataan ini ditunjukkan pada Tabel 3.4.

Tabel 3.4 Kemungkinan Perubahan dengan Pernyataan ALTER TABLE

Syntax	Keterangan
ADD [COLUMN] *column_description [FIRST | AFTER column]*	Tambahkan kolom baru di lokasi yang ditentukan (jika tidak ditentukan, kolom akan diletakkan di bagian akhir). Perhatikan bahwa deskripsi column_ memerlukan nama dan tipe, seperti dalam pernyataan CREATE.
ADD [COLUMN] (*column_description, column_description,...*)	Tambahkan satu atau lebih kolom baru di akhir tabel.
ADD INDEX [index] (column, ...)	Tambahkan indeks ke tabel pada kolom atau kolom-kolom yang ditentukan.
ADD PRIMARY KEY (column, ...)	Jadikan kolom atau kolom-kolom yang ditentukan sebagai kunci utama tabel.
ADD UNIQUE [index] (column, ...)	Tambahkan indeks unik ke tabel pada kolom atau kolom-kolom yang ditentukan.
ALTER [COLUMN] column {SET DEFAULT value | DROP DEFAULT}	Tambahkan atau hapus nilai default untuk kolom tertentu.

CHANGE [COLUMN] column new_column_description	Ubah kolom yang disebut kolom sehingga memiliki deskripsi yang tercantum.
MODIFY [COLUMN] column_description	Perhatikan bahwa ini dapat digunakan untuk mengubah nama kolom karena column_description menyertakan nama. Mirip dengan CHANGE. Dapat digunakan untuk mengubah jenis kolom, bukan nama.
DROP [COLUMN] column	Hapus kolom yang diberi nama.
DROP PRIMARY KEY	Hapus indeks utama (tetapi bukan kolomnya).
DROP INDEX index	Hapus indeks bernama.
RENAME[AS] new_table_name	Ganti nama tabel

Mari kita lihat beberapa penggunaan ALTER TABLE yang lebih umum.

Satu hal yang sering muncul adalah kesadaran bahwa Anda belum membuat kolom tertentu "cukup besar" untuk menampung data yang harus ditampungnya. Misalnya, dalam tabel Pelanggan, kami telah mengizinkan nama dengan panjang 30 karakter. Setelah kami mulai mendapatkan beberapa data, kami mungkin memperhatikan bahwa beberapa nama terlalu panjang dan terpotong. Kami dapat memperbaikinya dengan mengubah tipe data kolom sehingga panjangnya menjadi 45 karakter:

```
alter table customers
modify name char(45) not null;
```

Kejadian umum lainnya adalah perlunya menambahkan kolom. Bayangkan pajak penjualan atas buku diberlakukan secara lokal, dan Book-O-Rama perlu menambahkan jumlah pajak ke total pesanan, tetapi mencatatnya secara terpisah. Kita dapat menambahkan kolom pajak ke tabel Pesanan sebagai berikut:

```
alter table orders
add tax float(6,2) after amount;
```

Menghapus kolom adalah kasus lain yang sering terjadi. Kita dapat menghapus kolom yang baru saja kita tambahkan sebagai berikut:

```
alter table orders
drop tax;
```

Menghapus Catatan dari Basis Data

Menghapus baris dari basis data sangatlah mudah. Anda dapat melakukannya dengan menggunakan pernyataan DELETE, yang secara umum terlihat seperti ini:

```
DELETE FROM table
[WHERE condition] [LIMIT number]
```

Jika Anda menulis

```
DELETE FROM table;
```

dengan sendirinya, semua baris dalam tabel akan dihapus, jadi berhati-hatilah! Biasanya, Anda ingin menghapus baris tertentu, dan Anda dapat menentukan baris yang ingin dihapus dengan klausa `WHERE`. Anda dapat melakukan ini, misalnya, jika buku tertentu tidak lagi tersedia, atau jika pelanggan tertentu sudah lama tidak memesan, dan Anda ingin melakukan pembersihan:

```
delete from customers
where customerid=5;
```

Klausa `LIMIT` dapat digunakan untuk membatasi jumlah maksimum baris yang benar-benar dihapus.

Menghapus Tabel

Terkadang Anda mungkin ingin menghapus seluruh tabel. Anda dapat melakukannya dengan pernyataan `DROP TABLE`. Ini sangat sederhana, dan tampak seperti ini:

```
DROP TABLE table;
```

Ini akan menghapus semua baris dalam tabel dan tabel itu sendiri, jadi berhati-hatilah saat menggunakannya.

Menghapus Seluruh Basis Data

Anda dapat melangkah lebih jauh dan menghapus seluruh basis data dengan pernyataan `DROP DATABASE`, yang tampak seperti ini:

```
DROP DATABASE database;
```

Ini akan menghapus semua baris, semua tabel, semua indeks, dan basis data itu sendiri, jadi tidak perlu dikatakan lagi bahwa Anda harus agak berhati-hati saat menggunakan pernyataan ini.

BAB 4
MENGAKSES BASIS DATA MYSQL DARI WEB DENGAN PHP

Sebelumnya, dalam pekerjaan kita dengan PHP, kita menggunakan berkas datar untuk menyimpan dan mengambil data. Sekarang, setelah bekerja dengan MySQL untuk membuat basis data, kita dapat mulai menghubungkan basis data ini ke antarmuka berbasis Web.

Dalam bab ini, kita akan menjelaskan cara mengakses basis data Book-O-Rama dari Web menggunakan PHP. Anda akan mempelajari cara membaca dari dan menulis ke basis data, dan cara memfilter data masukan yang berpotensi bermasalah. Secara keseluruhan, kita akan membahas:

- Cara kerja arsitektur basis data Web
- Langkah-langkah dasar dalam melakukan kueri basis data dari Web
- Menyiapkan koneksi
- Mendapatkan informasi tentang basis data yang tersedia
- Memilih basis data untuk digunakan
- Melakukan kueri basis data
- Mengambil hasil kueri
- Memutuskan sambungan dari basis data
- Menempatkan informasi baru dalam basis data
- Mengamankan basis data Anda
- Fungsi PHP—MySQL lain yang bermanfaat
- Antarmuka basis data PHP lainnya

4.1 CARA KERJA ARSITEKTUR BASIS DATA WEB

Pada Bab 1, “Merancang Basis Data Web,” kami menguraikan cara kerja arsitektur basis data Web. Sekadar mengingatkan Anda, berikut ini langkah-langkahnya lagi:

1. Peramban Web pengguna mengeluarkan permintaan HTTP untuk halaman Web tertentu. Misalnya, pengguna mungkin telah meminta pencarian untuk semua buku yang ditulis oleh Michael Morgan di Book-O-Rama, menggunakan formulir HTML. Halaman hasil pencarian disebut `results.php`.
2. Server web menerima permintaan untuk `results.php`, mengambil file, dan meneruskannya ke mesin PHP untuk diproses.
3. Mesin PHP mulai mengurai skrip. Di dalam skrip terdapat perintah untuk terhubung ke basis data dan menjalankan kueri (melakukan pencarian buku). PHP membuka koneksi ke server MySQL dan mengirimkan kueri yang sesuai.
4. Server MySQL menerima kueri basis data, memprosesnya, dan mengirimkan hasilnya—daftar buku—kembali ke mesin PHP.
5. Mesin PHP menyelesaikan menjalankan skrip yang biasanya melibatkan pemformatan hasil kueri dengan baik dalam HTML. Kemudian, skrip tersebut mengembalikan HTML yang dihasilkan ke server Web.

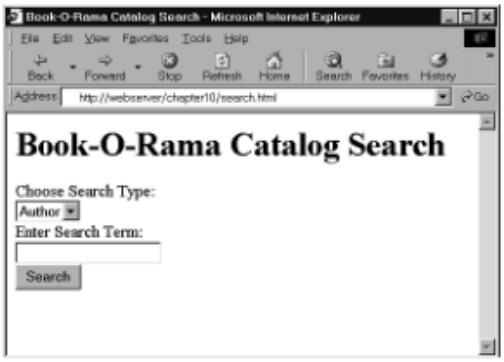
6. Server Web meneruskan HTML kembali ke browser, tempat pengguna dapat melihat daftar buku yang dimintanya.

   Sekarang, kita memiliki basis data MySQL, sehingga kita dapat menulis kode PHP untuk melakukan langkah-langkah sebelumnya. Kita akan mulai dengan formulir pencarian. Ini adalah formulir HTML biasa. Kode untuk formulir tersebut ditampilkan dalam Daftar 4.1.

Daftar 4.1 search.html—Halaman Pencarian Basis Data Book-O-Rama

```
<html>
<head>
  <title>Book-O-Rama Catalog Search</title>
</head>

<body>
  <h1>Book-O-Rama Catalog Search</h1>
  <form action=”results.php” method=”post”>
    Choose Search Type:<br>
    <select name=”searchtype”>
      <option value=”author”>Author
      <option value=”title”>Title
      <option value=”isbn”>ISBN
    </select>
    <br>
    Enter Search Term:<br>
    <input name=”searchterm” type=text>
    <br>
    <input type=submit value=”Search”>
  </form>

</body>
</html>
```

Ini adalah formulir HTML yang cukup mudah. Output dari HTML ini ditunjukkan pada Gambar 4.1.



Gambar 4.1 Formulir pencarian cukup umum, jadi Anda dapat mencari buku berdasarkan judul, penulis, atau ISBN.

Skrip yang akan dipanggil saat tombol Cari ditekan adalah `results.php`. Skrip ini tercantum secara lengkap di Listing 4.2. Selama bab ini, kita akan membahas apa yang dilakukan skrip ini dan cara kerjanya.

Listing 4.2 `results.php`—Mengambil Hasil Pencarian dari Basis Data MySQL dan Memformatnya untuk Ditampilkan

```
<html>
<head>
  <title>Book-O-Rama Search Results</title>
</head>
<body>
<h1>Book-O-Rama Search Results</h1>
<?
  trim($searchterm);
  if (!$searchtype || !$searchterm)
  {
    echo “You have not entered search details. Please go back and try
      again.”;
    exit;
  }

  $searchtype = addslashes($searchtype);
  $searchterm = addslashes($searchterm);

  @ $db = mysql_pconnect(“localhost”, “bookorama”, “bookorama”);

  if (!$db)
  {
    echo “Error: Could not connect to database. Please try again later.”;
  exit;
  }

  mysql_select_db(“books”);
  $query = “select * from books where “.$searchtype.” Like
    ‘%”.$searchterm.”%’”;
  $result = mysql_query($query);
  $num_results = mysql_num_rows($result);

  echo “<p>Number of books found: “.$num_results.”</p>”;

  for ($i=0; $i <$num_results; $i++)
  {
    $row = mysql_fetch_array($result);
    echo “<p><strong>”.($i+1).”. Title: “;
    echo htmlspecialchars( stripslashes($row[“title”]));
    echo “</strong><br>Author: “;
```

```
        echo htmlspecialchars (stripslashes($row[“author”]));
        echo “<br>ISBN: “;
        echo htmlspecialchars (stripslashes($row[“isbn”]));
        echo “<br>Price: “;
        echo htmlspecialchars (stripslashes($row[“price”]));
        echo “</p>”;
    }

?>

</body>
</html>
```

Gambar 4.2 mengilustrasikan hasil penggunaan skrip ini untuk melakukan pencarian.

Gambar 4.2 Hasil pencarian buku tentang Java di basis data disajikan dalam halaman Web menggunakan skrip `results.php`.

Langkah-Langkah Dasar dalam Menanyakan Basis Data dari Web

Dalam skrip apa pun yang digunakan untuk mengakses basis data dari Web, Anda akan mengikuti beberapa langkah dasar:

1. Memeriksa dan memfilter data yang datang dari pengguna.
2. Menyiapkan koneksi ke basis data yang sesuai.
3. Menanyakan basis data.
4. Mengambil hasilnya.
5. Menyajikan kembali hasilnya kepada pengguna.

Ini adalah langkah-langkah yang telah kami ikuti dalam skrip results.php, dan kami akan membahas masing-masing langkah secara bergantian.

4.2 MEMERIKSA DAN MEMFILTER DATA INPUT

Kami memulai skrip kami dengan menghapus spasi kosong yang mungkin dimasukkan pengguna secara tidak sengaja di awal atau akhir istilah pencariannya. Kami melakukannya dengan menerapkan fungsi `trim()` ke

```
$searchterm. trim($searchterm);
```

Langkah kita selanjutnya adalah memverifikasi bahwa pengguna telah memasukkan istilah pencarian dan jenis pencarian. Perhatikan bahwa kita memeriksa apakah ia memasukkan istilah pencarian setelah memangkas spasi dari ujung `$searchterm.` Jika kita menyusun baris-baris ini dalam urutan yang berlawanan, kita bisa mendapatkan situasi di mana istilah pencarian pengguna tidak kosong, jadi tidak membuat pesan kesalahan, tetapi semuanya berupa spasi, jadi dihapus oleh

```
trim():

if (!$searchtype || !$searchterm)
{
  echo “You have not entered search details. Please go back and try
  again.”;
  exit;
}
```

Anda akan melihat bahwa kami telah memeriksa variabel `$searchtype` meskipun dalam kasus ini berasal dari `HTML SELECT.` Anda mungkin bertanya mengapa kami repot-repot memeriksa data yang harus diisi. Penting untuk diingat bahwa mungkin ada lebih dari satu antarmuka ke basis data Anda.

Misalnya, Amazon memiliki banyak afiliasi yang menggunakan antarmuka pencarian mereka. Selain itu, penting untuk menyaring data jika terjadi masalah keamanan yang dapat muncul karena pengguna datang dari titik masuk yang berbeda.

Selain itu, ketika Anda akan menggunakan input data apa pun oleh pengguna, penting untuk memfilternya dengan tepat untuk setiap karakter kontrol. Berbicara tentang fungsi `addslashes()` dan `stripslashes().` Anda perlu menggunakan `addslashes()` saat mengirimkan input pengguna apa pun ke basis data seperti MySQL dan `stripslashes()` saat mengembalikan output ke pengguna yang karakter kontrolnya telah dicoret. Dalam kasus ini, kami telah menggunakan `addSlashes()` pada istilah pencarian:

```
$searchterm = addslashes($searchterm);
```

Kami juga telah menggunakan `stripslashes()` pada data yang kembali dari basis data. Tidak ada data yang kami masukkan secara manual ke dalam basis data yang memiliki garis miring—namun, data tersebut juga tidak memiliki karakter kontrol apa pun di dalamnya. Panggilan ke `stripslashes()` tidak akan berpengaruh. Saat kami membangun antarmuka Web untuk basis data, kemungkinan besar kami ingin memasukkan buku baru di dalamnya, dan beberapa detail yang dimasukkan oleh pengguna mungkin berisi karakter ini. Saat kami memasukkannya ke dalam basis data, kami akan memanggil `addslashes(),` yang berarti bahwa kami harus memanggil `stripslashes` saat mengambil kembali data. Ini adalah kebiasaan yang masuk akal untuk dilakukan.

Kami menggunakan fungsi `htmlspecialchars()` untuk mengodekan karakter yang memiliki arti khusus dalam HTML. Data pengujian kami saat ini tidak menyertakan simbol ampersand (&), kurang dari (<), lebih dari (>), atau tanda kutip ganda (“), tetapi banyak judul buku bagus yang mengandung ampersand. Dengan menggunakan fungsi ini, kami dapat menghilangkan kesalahan di masa mendatang.

Menyiapkan Koneksi

Kami menggunakan baris ini dalam skrip kami untuk terhubung ke server MySQL:

```
@ $db = mysql_pconnect(“localhost”, “bookorama”, “bookorama”);
```

Kami telah menggunakan fungsi `mysql_pconnect()` untuk terhubung ke database. Fungsi ini memiliki prototipe berikut:

```
int mysql_pconnect(  [string host [:port] [:/socketpath] ] ,
                     [string user] , [string password] );
```

Secara umum, Anda akan memberikan nama host tempat server MySQL berjalan, nama pengguna untuk login, dan kata sandi pengguna tersebut. Semua ini bersifat opsional, dan jika Anda tidak menentukannya, fungsi tersebut menggunakan beberapa default yang masuk akal—localhost untuk host, nama pengguna tempat proses PHP berjalan, dan kata sandi kosong.

Fungsi tersebut mengembalikan pengenal tautan ke basis data MySQL Anda jika berhasil (yang harus Anda simpan untuk penggunaan lebih lanjut) atau false jika gagal. Hasilnya layak untuk diperiksa karena tidak ada kode lainnya yang akan berfungsi tanpa koneksi basis data yang valid. Kami telah melakukannya menggunakan kode berikut:

```
if (!$db)
{
  echo “Error: Could not connect to database. Please try again later.”;
  exit;
}
```

Fungsi alternatif yang melakukan hal yang hampir sama dengan `mysql_pconnect()` adalah `mysql_connect()`. Perbedaannya adalah `mysql_pconnect()` mengembalikan koneksi persisten ke basis data.

Koneksi normal ke basis data akan ditutup saat skrip selesai dieksekusi, atau saat skrip memanggil fungsi `mysql_close()`. Koneksi persisten tetap terbuka setelah skrip selesai dieksekusi dan tidak dapat ditutup dengan fungsi `mysql_close()`.

Anda mungkin bertanya-tanya mengapa kita ingin melakukan ini. Jawabannya adalah membuat koneksi ke basis data melibatkan sejumlah overhead tertentu dan karenanya memerlukan waktu. Saat `mysql_pconnect()` dipanggil, sebelum mencoba terhubung ke basis data, ia akan secara otomatis memeriksa apakah ada koneksi persisten yang sudah terbuka.

Jika demikian, ia akan menggunakan koneksi ini daripada membuka yang baru. Ini menghemat waktu dan overhead server.

Perlu dicatat juga bahwa koneksi persisten tidak bertahan jika Anda menjalankan PHP sebagai CGI. (Setiap panggilan ke skrip PHP memulai instance PHP baru dan menutupnya saat skrip selesai dieksekusi. Ini juga menutup semua koneksi persisten.)

Perlu diingat bahwa ada batasan jumlah koneksi MySQL yang dapat ada pada saat yang sama. Parameter MySQL `max_connections` menentukan batasan ini. Tujuan dari parameter ini dan parameter Apache terkait MaxClients adalah untuk memberi tahu server agar menolak permintaan koneksi baru daripada mengizinkan semua sumber daya mesin digunakan pada waktu sibuk atau saat perangkat lunak mogok.

Anda dapat mengubah kedua parameter ini dari nilai default dengan mengedit file konfigurasi. Untuk menyetel MaxClients di Apache, edit `file httpd.conf` di sistem Anda. Untuk menyetel `max_connections` untuk MySQL, edit `file my.conf.`

Jika Anda menggunakan koneksi persisten dan hampir setiap halaman di situs Anda melibatkan akses basis data, kemungkinan besar Anda akan memiliki koneksi persisten yang terbuka untuk setiap proses Apache. Ini dapat menyebabkan masalah jika Anda membiarkan parameter ini disetel ke nilai default. Secara default, Apache mengizinkan 150 koneksi, tetapi MySQL hanya mengizinkan 100. Pada waktu sibuk, mungkin tidak ada cukup koneksi untuk semua orang. Bergantung pada kemampuan perangkat keras Anda, Anda harus menyesuaikannya sehingga setiap proses server Web dapat memiliki koneksi.

4.3 MEMILIH BASIS DATA UNTUK DIGUNAKAN

Anda akan ingat bahwa ketika kita menggunakan MySQL dari antarmuka baris perintah, kita perlu memberitahukannya basis data mana yang akan kita gunakan dengan perintah seperti use books;

Kita juga perlu melakukan ini ketika menghubungkan dari Web. Kita melakukannya dari PHP dengan memanggil fungsi `mysql_select_db(),` yang telah kita lakukan dalam kasus ini sebagai berikut:

```
mysql_select_db(“books”);
```

Fungsi `mysql_select_db()` memiliki prototipe berikut:

```
int mysql_select_db(string database, [int database_connection] );
```

Fungsi ini akan mencoba menggunakan basis data yang disebut database. Anda juga dapat secara opsional menyertakan tautan basis data tempat Anda ingin melakukan operasi ini (dalam kasus ini $db), tetapi jika Anda tidak menentukannya, tautan terakhir yang dibuka akan digunakan. Jika Anda tidak membuka tautan, tautan default akan dibuka seolah-olah Anda telah memanggil `mysql_connect().`

Melakukan Query pada Basis Data

Untuk benar-benar melakukan query, kita dapat menggunakan fungsi `mysql_query()`. Namun, sebelum melakukannya, sebaiknya Anda menyiapkan query yang ingin Anda jalankan:

```
$query = “select * from books where “.$searchtype.” like ‘%”.$searchterm.”%’”;
```

Dalam kasus ini, kami mencari nilai input pengguna (`$searchterm`) di bidang yang ditentukan pengguna (`$searchtype`). Anda akan melihat bahwa kami menggunakan like untuk mencocokkan daripada equal—biasanya merupakan ide yang baik untuk bersikap lebih toleran dalam pencarian basis data.
Sekarang kita dapat menjalankan kueri:

```
$result = mysql_query($query);
```

Fungsi `mysql_query()` memiliki prototipe berikut:

```
int mysql_query(string query, [int database_connection] );
```

Anda meneruskan kueri yang ingin Anda jalankan, dan secara opsional, tautan basis data (sekali lagi, dalam kasus ini $db). Jika tidak ditentukan, fungsi akan menggunakan tautan terakhir yang dibuka. Jika tidak ada, fungsi akan membuka tautan default seolah-olah Anda telah memanggil `mysql_connect()`.

Anda mungkin ingin menggunakan fungsi `mysql_db_query()` sebagai gantinya. Fungsi ini memiliki prototipe berikut:

```
int mysql_db_query(string database, string query, [int database_connection] );
```

Fungsi ini sangat mirip tetapi memungkinkan Anda menentukan basis data tempat Anda ingin menjalankan kueri. Fungsi ini seperti gabungan fungsi `mysql_select_db()` dan `mysql_query()`.

Kedua fungsi ini mengembalikan pengidentifikasi hasil (yang memungkinkan Anda mengambil hasil kueri) jika berhasil dan salah jika gagal. Anda harus menyimpan ini (seperti yang kami miliki dalam kasus ini di `$result`) sehingga Anda dapat melakukan sesuatu yang berguna dengannya.

Mengambil Hasil Kueri

Berbagai fungsi tersedia untuk memisahkan hasil dari pengidentifikasi hasil dengan berbagai cara. Pengidentifikasi hasil adalah kunci untuk mengakses nol, satu, atau lebih baris yang dikembalikan oleh kueri.

Dalam contoh kami, kami telah menggunakan dua di antaranya: `mysql_numrows()` dan `mysql_fetch_array()`. Fungsi `mysql_numrows()` memberi Anda jumlah baris yang dikembalikan oleh kueri. Anda harus meneruskannya dengan pengidentifikasi hasil, seperti ini:

```
$num_results = mysql_num_rows($result);
```

Penting untuk mengetahui hal ini—jika kita berencana untuk memproses atau menampilkan hasilnya, kita tahu berapa banyak hasilnya dan sekarang dapat mengulanginya:

```
for ($i=0; $i <$num_results; $i++)
{
  // process results
}
```

Dalam setiap iterasi loop ini, kita memanggil `mysql_fetch_array()`. Loop tidak akan dijalankan jika tidak ada baris yang dikembalikan. Ini adalah fungsi yang mengambil setiap baris dari `resultset` dan mengembalikan baris tersebut sebagai array asosiatif, dengan setiap kunci merupakan nama atribut dan setiap nilai merupakan nilai yang sesuai dalam array:

```
$row = mysql_fetch_array($result);
```

Mengingat array asosiatif `$row`, kita dapat menelusuri setiap bidang dan menampilkannya dengan tepat, misalnya:

```
echo “<br>ISBN: “;
echo stripslashes($row[“isbn”]);
```

Seperti yang disebutkan sebelumnya, kita telah memanggil `stripslashes()` untuk merapikan nilai sebelum menampilkannya. Ada beberapa variasi dalam mendapatkan hasil dari pengidentifikasi hasil. Alih-alih array asosiatif, kita dapat mengambil hasil dalam array yang dijumlahkan dengan `mysql_fetch_row()`, sebagai berikut:

```
$row = mysql_fetch_row($result);
```

Nilai atribut akan dicantumkan di setiap nilai array `$row[0], $row[1],` dan seterusnya. Anda juga dapat mengambil baris ke dalam objek dengan fungsi `mysql_fetch_object()`:

```
$row = mysql_fetch_object($result);
```

Anda kemudian dapat mengakses setiap atribut melalui `$row->title, $row->author`, dan seterusnya. Masing-masing pendekatan ini mengambil baris pada satu waktu. Pendekatan lainnya adalah mengakses bidang pada satu waktu menggunakan `mysql_result()`. Untuk ini, Anda harus menentukan nomor baris (dari nol hingga jumlah baris—1) serta nama bidang. Misalnya

```
$row = mysql_result($result, $i, “title”);
```

Anda dapat menentukan nama bidang sebagai string (baik dalam bentuk "`title`" atau "`books.title`") atau sebagai angka (seperti dalam `mysql_fetch_row()`). Anda tidak boleh mencampur penggunaan `mysql_result()` dengan fungsi pengambilan lainnya. Fungsi pengambilan berorientasi baris jauh lebih efisien daripada `mysql_result()`, jadi secara umum Anda harus menggunakan salah satunya.

Memutuskan Sambungan dari Basis Data

Anda dapat menggunakan

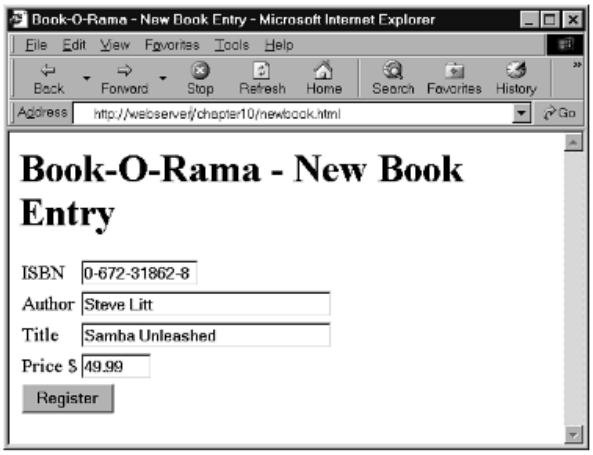
```
mysql_close(database_connection);
```

untuk menutup sambungan basis data nonpersisten. Hal ini tidak sepenuhnya diperlukan karena sambungan tersebut akan ditutup saat skrip selesai dieksekusi.

Memasukkan Informasi Baru ke dalam Basis Data

Memasukkan item baru ke dalam basis data sangat mirip dengan mengeluarkan item dari basis data. Anda mengikuti langkah dasar yang sama—membuat sambungan, mengirim kueri, dan memeriksa hasilnya. Dalam kasus ini, kueri yang Anda kirim akan berupa `INSERT`, bukan `SELECT`.

Meskipun semuanya sangat mirip, terkadang ada baiknya untuk melihat contoh. Pada Gambar 4.3, Anda dapat melihat formulir HTML dasar untuk memasukkan buku baru ke dalam basis data.



Gambar 4.3 Antarmuka untuk memasukkan buku baru ke dalam basis data ini dapat digunakan oleh staf Book-O-Rama.

HTML untuk halaman ini ditunjukkan pada Listing 4.3.

Listing 10.3 newbook.html—HTML untuk Halaman Entri Buku

```
<html>
<head>
  <title>Book-O-Rama - New Book Entry</title>
</head>
```

```
<body>
  <h1>Book-O-Rama - New Book Entry</h1>

  <form action=”insert_book.php” method=”post”>
    <table border=0>
      <tr>
        <td>ISBN</td>
          <td><input type=text name=isbn maxlength=13 size=13><br></td>
      </tr>
      <tr>
    <td>Author</td>
    <td> <input type=text name=author maxlength=30 size=30><br></td>
  </tr>
  <tr>
    <td>Title</td>
    <td> <input type=text name=title maxlength=60 size=30><br></td>
  </tr>
  <tr>
    <td>Price $</td>
    <td><input type=text name=price maxlength=7 size=7><br></td>
  </tr>
  <tr>
    <td colspan=2><input type=submit value=”Register”></td>
  </tr>
  </table>
</form>
</body>
</html>
```

Hasil formulir ini diteruskan ke `insert_book.php`, skrip yang mengambil detail, melakukan beberapa validasi minor, dan mencoba menulis data ke dalam basis data. Kode untuk skrip ini ditunjukkan pada Listing 4.4.

Listing 4.4 `insert_book.php`—Skrip Ini Menulis Buku Baru ke dalam Basis Data

```
<html>
<head>
  <title>Book-O-Rama Book Entry Results</title>
</head>
<body>
<h1>Book-O-Rama Book Entry Results</h1>
<?
  if (!$isbn || !$author || !$title || !$price)
    {
      echo “You have not entered all the required details.<br>”
        .”Please go back and try again.”;
```

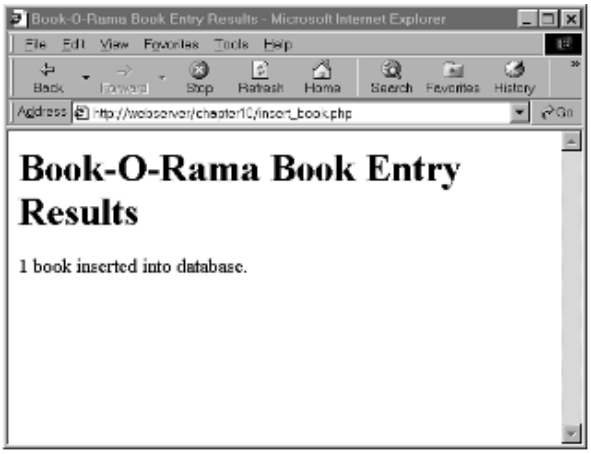
```
        exit;
      }

    $isbn = addslashes($isbn);
    $author = addslashes($author);
    $title = addslashes($title);
    $price = doubleval($price);

    @ $db = mysql_pconnect("localhost", "bookorama", "bookorama");

    if (!$db)
      {
        echo "Error: Could not connect to database. Please try again
        later.";
      exit;
    }

    mysql_select_db("books");
    $query = "insert into books values
        ('".$isbn."', '".$author."', '".$title."', '".$price."')";
    $result = mysql_query($query);
    if ($result)
        echo mysql_affected_rows()." book inserted into database.";
?>

</body>
</html>
```

Hasil dari penyisipan buku yang berhasil ditunjukkan pada Gambar 4.4.

Jika Anda melihat kode untuk `insert_book.php`, Anda akan melihat bahwa sebagian besar kode tersebut mirip dengan skrip yang kami tulis untuk mengambil data dari basis data. Kami telah memeriksa bahwa semua kolom formulir telah diisi, dan memformatnya dengan benar untuk penyisipan ke dalam basis data dengan `addslashes()`:

```
$isbn = addslashes($isbn);
$author = addslashes($author);
$title = addslashes($title);
$price = doubleval($price);
```

Karena harga disimpan dalam basis data sebagai float, kita tidak ingin memasukkan garis miring ke dalamnya. Kita dapat memperoleh efek yang sama dengan memfilter karakter ganjil apa pun pada bidang numerik ini dengan memanggil `doubleval()`, ini juga akan menangani simbol mata uang apa pun yang mungkin diketik pengguna dalam formulir.



Gambar 4.4 Skrip berhasil diselesaikan dan melaporkan bahwa buku telah ditambahkan ke basis data.

Sekali lagi, kita telah terhubung ke basis data menggunakan mysql_pconnect(), dan menyiapkan kueri untuk dikirim ke basis data. Dalam kasus ini, kueri tersebut adalah SQL INSERT:

```
$query = “insert into books values
        (‘“.$isbn.”’, ‘“.$author.”’, ‘“.$title.”’, ‘“.$price.”’)”;
$result = mysql_query($query);
```

Hal ini dijalankan pada basis data dengan cara biasa dengan memanggil `mysql_query()`. Satu perbedaan signifikan antara penggunaan INSERT dan SELECT adalah pada penggunaan `mysql_affected_rows()`:

```
echo mysql_affected_rows().” book included into database.”;
```

Pada skrip sebelumnya, kita menggunakan `mysql_num_rows()` untuk menentukan berapa banyak baris yang dikembalikan oleh SELECT. Saat Anda menulis kueri yang mengubah basis data seperti `INSERT`, `DELETE`, dan `UPDATE`, Anda harus menggunakan `mysql_affected_rows()` sebagai gantinya.

Ini mencakup dasar-dasar penggunaan basis data MySQL dari PHP. Kita akan melihat sekilas beberapa fungsi berguna lainnya yang belum kita bahas.

4.5 FUNGSI PHP-MYSQL LAINNYA

Ada beberapa fungsi PHP-MySQL berguna lainnya, yang akan kita bahas secara singkat.

Membebaskan Sumber Daya

Jika Anda mengalami masalah memori saat skrip sedang berjalan, Anda mungkin ingin menggunakan `mysql_free_result()`. Ini memiliki prototipe berikut:

```
int mysql_free_result(int result);
```

Anda memanggilnya dengan pengenal hasil, seperti ini:

```
mysql_free_result($result);
```

Ini memiliki efek membebaskan memori yang digunakan untuk menyimpan hasil. Jelas Anda tidak akan memanggil ini sampai Anda selesai bekerja dengan kumpulan hasil.

Membuat dan Menghapus Basis Data

Untuk membuat basis data MySQL baru dari skrip PHP, Anda dapat menggunakan `mysql_create_db()`, dan untuk menghapusnya, Anda dapat menggunakan `mysql_drop_db()`.

Fungsi-fungsi ini memiliki prototipe berikut:

```
int mysql_create_db(string database, [int database_connection] );
int mysql_drop_db(string database, [int database_connection] );
```

Kedua fungsi ini mengambil nama basis data dan koneksi opsional. Jika tidak ada koneksi yang diberikan, koneksi terakhir yang terbuka akan digunakan. Fungsi-fungsi ini akan mencoba membuat atau menghapus basis data yang diberi nama. Kedua fungsi akan mengembalikan true jika berhasil dan false jika gagal.

Antarmuka Basis Data PHP Lainnya

PHP mendukung pustaka untuk menghubungkan ke sejumlah besar basis data termasuk Oracle, Microsoft SQL Server, mSQL, dan PostgreSQL. Secara umum, prinsip-prinsip untuk menghubungkan dan meminta kueri ke salah satu basis data ini hampir sama. Nama-nama fungsi individual bervariasi, dan basis data yang berbeda memiliki fungsionalitas yang sedikit berbeda, tetapi jika Anda dapat terhubung ke MySQL, Anda seharusnya dapat dengan mudah menyesuaikan pengetahuan Anda dengan yang lain.

Jika Anda ingin menggunakan basis data yang tidak memiliki pustaka khusus yang tersedia di PHP, Anda dapat menggunakan fungsi ODBC generik. ODBC adalah singkatan dari *Open Database Connectivity* dan merupakan standar untuk koneksi ke basis data. Fungsi ini memiliki fungsionalitas yang paling terbatas dari semua set fungsi, karena alasan yang cukup jelas. Jika Anda harus kompatibel dengan semuanya, Anda tidak dapat memanfaatkan fitur-fitur khusus apa pun.

Selain pustaka yang disertakan dalam PHP, tersedia kelas abstraksi basis data seperti Metabase yang memungkinkan Anda menggunakan nama fungsi yang sama untuk setiap jenis basis data yang berbeda.

BAB 5
MYSQL TINGKAT LANJUT

Dalam bab ini, kita akan membahas beberapa topik MySQL tingkat lanjut termasuk hak istimewa tingkat lanjut, keamanan, dan pengoptimalan.
Topik yang akan kita bahas adalah

- Memahami sistem hak istimewa secara mendetail
- Mengamankan basis data MySQL Anda
- Mendapatkan informasi lebih lanjut tentang basis data
- Mempercepat dengan indeks
- Kiat pengoptimalan
- Berbagai jenis tabel

5.1 MEMAHAMI SISTEM HAK ISTIMEWA SECARA DETAIL

Sebelumnya (dalam Bab 2, “Membuat Basis Data Web”) kita membahas pengaturan pengguna dan pemberian hak istimewa kepada mereka. Kita melakukannya dengan perintah `GRANT`. Jika Anda akan mengelola basis data MySQL, akan berguna untuk memahami dengan tepat apa yang dilakukan `GRANT` dan cara kerjanya.

Saat Anda mengeluarkan pernyataan `GRANT`, hal itu memengaruhi tabel dalam basis data khusus yang disebut mysql. Informasi hak istimewa disimpan dalam lima tabel dalam basis data ini. Mengingat hal ini, saat memberikan hak istimewa pada basis data, Anda harus berhati-hati dalam memberikan akses ke basis data mysql. Satu catatan tambahan adalah bahwa perintah `GRANT` hanya tersedia mulai dari MySQL versi 3.22.11 dan seterusnya.

Kita dapat melihat apa yang ada di database mysql dengan masuk sebagai administrator dan mengetik:

```
use mysql;
```

Jika Anda melakukan ini, Anda kemudian dapat melihat tabel dalam database ini dengan mengetik:

```
show tables;
```

Hasil yang Anda dapatkan akan terlihat seperti ini:

```
+-----------------+
| Tables in mysql |
+-----------------+
| columns_priv    |
| db              |
| host            |
| tables_priv     |
| user            |
+-----------------+
```

Masing-masing tabel ini menyimpan informasi tentang hak istimewa. Tabel-tabel ini terkadang disebut tabel pemberian izin. Tabel-tabel ini bervariasi dalam fungsi spesifiknya, tetapi semuanya memiliki fungsi umum yang sama, yaitu menentukan apa yang boleh dan tidak boleh dilakukan oleh pengguna. Masing-masing tabel berisi dua jenis bidang: bidang cakupan, yang mengidentifikasi pengguna, host, dan bagian dari basis data; dan bidang hak istimewa, yang mengidentifikasi tindakan mana yang dapat dilakukan oleh pengguna tersebut dalam cakupan tersebut.

Tabel pengguna digunakan untuk memutuskan apakah pengguna dapat terhubung ke server MySQL dan apakah ia memiliki hak istimewa administrator. Tabel db dan host menentukan basis data mana yang dapat diakses pengguna. Tabel `tables_priv` menentukan tabel mana dalam basis data yang dapat digunakan pengguna, dan tabel `columns_priv` menentukan kolom mana dalam tabel yang dapat mereka akses.

Tabel pengguna

Tabel ini berisi detail hak istimewa pengguna global. Tabel ini menentukan apakah pengguna diizinkan untuk terhubung ke server MySQL sama sekali, dan apakah ia memiliki hak istimewa tingkat global; yaitu, hak istimewa yang berlaku untuk setiap basis data dalam sistem.

Kita dapat melihat struktur tabel ini dengan mengeluarkan pernyataan describe user; Skema tabel pengguna ditunjukkan pada Tabel 5.1.

Tabel 5.1 Skema Tabel Pengguna dalam Database MySQL

Field	***Type***
`Host`	char (60)
`User`	char (16)
`Password`	char (16)
`Select_priv`	enum ('N' ,'Y')
`Insert_priv`	enum ('N' , 'Y')
`Update_priv`	enum ('N' ,'Y')
`Delete_priv`	enum ('N' ,'Y')
`Create_priv`	enum ('N' ,'Y')
`Drop_priv`	enum ('N' ,'Y')
`Reload_priv`	enum ('N' ,'Y')
`Shutdown_priv`	enum ('N' ,'Y')
`Process_priv`	enum ('N' ,'Y')
`File_priv`	enum ('N' ,'Y')

Grant_priv	enum ('N','Y')
References_priv	enum ('N','Y')
Index_priv	enum ('N','Y')
Alter priv	enum ('N', 'Y')

Setiap baris dalam tabel ini sesuai dengan serangkaian hak istimewa untuk pengguna yang datang dari host dan masuk dengan kata sandi Password. Ini adalah bidang cakupan untuk tabel ini, karena bidang ini menjelaskan cakupan bidang lainnya, yang disebut bidang hak istimewa.

Hak istimewa yang tercantum dalam tabel ini (dan yang lainnya setelahnya) sesuai dengan hak istimewa yang kami berikan menggunakan GRANT di Bab 4. Misalnya, Select_priv sesuai dengan hak istimewa untuk menjalankan perintah SELECT.

Jika pengguna memiliki hak istimewa tertentu, nilai dalam kolom tersebut akan menjadi Y. Sebaliknya, jika pengguna belum diberikan hak istimewa tersebut, nilainya akan menjadi N.

Semua hak istimewa yang tercantum dalam tabel pengguna bersifat global, yaitu, berlaku untuk semua basis data dalam sistem (termasuk basis data mysql). Oleh karena itu, administrator akan memiliki beberapa Y di sana, tetapi sebagian besar pengguna harus memiliki semua N. Pengguna normal harus memiliki hak untuk basis data yang sesuai, bukan semua tabel.

Tabel db dan host

Sebagian besar hak istimewa pengguna rata-rata disimpan dalam tabel db dan host. Tabel db menentukan pengguna mana yang dapat mengakses basis data mana dari host mana. Hak istimewa yang tercantum dalam tabel ini berlaku untuk basis data mana pun yang diberi nama dalam baris tertentu.

Tabel host melengkapi tabel db. Jika pengguna akan terhubung ke basis data dari beberapa host, tidak ada host yang akan dicantumkan untuk pengguna tersebut dalam tabel db. Sebaliknya, ia akan memiliki serangkaian entri dalam tabel host, satu untuk menentukan hak istimewa untuk setiap kombinasi pengguna-host. Skema dari kedua tabel ini ditunjukkan pada Tabel 5.2 dan 5.3, masing-masing.

Tabel 5.2 Skema Tabel db dalam Basis Data mysql

Field	***Type***
Host	char (60)
db	char (64)
User	char (16)
Select_priv	enum ('N','Y')
Insert_priv	enum ('N','Y')
Update_priv	enum ('N','Y')
Delete_priv	enum ('N','Y')
Create_priv	enum ('N','Y')
Drop_priv	enum ('N','Y')
Grant_priv	enum ('N','Y')

References_priv	enum ('N' ,'Y')
Index_priv	enum ('N' ,'Y')
Alter_priv	enum ('N' ,'Y')

Tabel 5.3 Skema Tabel Host pada database mysql

Field	***Type***
Host	char (60)
Db	char (64)
Select_priv	enum ('N' ,'Y')
Insert_priv	enum ('N' ,'Y')
Update_priv	enum ('N' ,'Y')
Delete_priv	enum ('N' ,'Y')
Create_priv	enum ('N' ,'Y')
Drop_priv	enum ('N' , 'Y')
Grant_priv	enum ('N' ,'Y')
References_priv	enum ('N' , 'Y')
Index_priv	enum ('N' ,'Y')
Alter_priv	enum ('N', 'Y')

Tabel `tables_priv` dan `columns_priv`

Kedua tabel ini masing-masing digunakan untuk menyimpan hak istimewa tingkat tabel dan hak istimewa tingkat kolom. Keduanya berfungsi seperti tabel db, kecuali bahwa keduanya menyediakan hak istimewa untuk tabel dalam database tertentu dan kolom dalam tabel tertentu.

Tabel 5.4 Skema Tabel `tables_priv` di Database mysql

Field	***Type***
Host	char (60)
Db	char (64)
User	char (16)
Table_name	char (64)
Grantor	char (77)
Timestamp	timestamp(14)
Table_priv	set ('Select', 'Insert', 'Update' , 'Delete', 'Create', 'Drop', 'Grant', 'References', 'Index', 'Alter')
Column_priv	set('Select', 'Insert', 'Update', 'References')

Tabel ini memiliki struktur yang sedikit berbeda dengan tabel pengguna, db, dan host. Skema untuk tabel `tables_priv` dan tabel `columns_priv` masing-masing ditunjukkan pada Tabel 5.4 dan 5.5.

TABEL 5.5 Skema Tabel `columns_priv` di Database mysql

Field	***Type***
Host	char(60)

Db	char(60)
User	char(16)
Table_name	char(60)
Column_name	char (59)
Timestamp	timestamp(14)
Column_priv	set('Select', 'Insert', 'Update', 'References')

Kolom Grantor dalam tabel tables_priv menyimpan pengguna yang memberikan hak istimewa ini kepada pengguna ini. Kolom Timestamp dalam kedua tabel ini menyimpan tanggal dan waktu saat hak istimewa diberikan.

5.2 CARA MYSQL MENGGUNAKAN TABEL GRANT

MySQL menggunakan tabel grant untuk menentukan apa yang diizinkan untuk dilakukan pengguna dalam proses dua tahap:

1. Verifikasi koneksi. Di sini, MySQL memeriksa apakah Anda diizinkan untuk terhubung sama sekali, berdasarkan informasi dari tabel pengguna, seperti yang ditunjukkan sebelumnya. Ini berdasarkan nama pengguna, nama host, dan kata sandi Anda. Jika nama pengguna kosong, maka nama pengguna tersebut cocok dengan semua pengguna. Nama host dapat ditentukan dengan karakter wildcard (%). Ini dapat digunakan sebagai seluruh bidang—yaitu, % cocok dengan semua host—atau sebagai bagian dari nama host, misalnya, %.tangledweb.com.au cocok dengan semua host yang diakhiri dengan .tangledweb.com.au. Jika bidang kata sandi kosong, maka tidak diperlukan kata sandi. Lebih aman untuk menghindari pengguna kosong, wildcard di host, dan pengguna tanpa kata sandi.
2. Verifikasi permintaan. Setiap kali Anda memasukkan permintaan, setelah Anda membuat koneksi, MySQL akan memeriksa apakah Anda memiliki tingkat hak istimewa yang sesuai untuk melakukan permintaan tersebut. Sistem akan mulai dengan memeriksa hak istimewa global Anda (di tabel pengguna) dan jika hak istimewa tersebut tidak mencukupi, akan memeriksa tabel db dan host. Jika hak istimewa Anda masih belum mencukupi, MySQL akan memeriksa tabel tables_priv, dan, jika ini tidak mencukupi, akan memeriksa tabel columns_priv.

Memperbarui Hak Istimewa: Kapan Perubahan Berlaku?

Server MySQL secara otomatis membaca tabel hibah saat dimulai, dan saat Anda mengeluarkan pernyataan GRANT dan REVOKE. Namun, sekarang setelah kita mengetahui di mana dan bagaimana hak istimewa tersebut disimpan, kita dapat mengubahnya secara manual. Saat Anda memperbaruinya secara manual, server MySQL tidak akan menyadari bahwa hak istimewa tersebut telah berubah.

Anda perlu memberi tahu server bahwa telah terjadi perubahan, dan ada tiga cara untuk melakukannya. Anda dapat mengetik FLUSH PRIVILEGES; pada prompt MySQL (Anda harus login sebagai administrator untuk melakukan ini). Ini adalah cara yang paling umum digunakan untuk memperbarui hak istimewa. Atau Anda dapat menjalankan mysqladmin flush-privileges atau mysqladmin reload dari sistem operasi Anda.

Setelah ini, hak istimewa tingkat global akan diperiksa saat pengguna terhubung lagi; hak istimewa basis data akan diperiksa saat pernyataan penggunaan berikutnya dikeluarkan; dan hak istimewa tingkat tabel dan kolom akan diperiksa pada permintaan pengguna berikutnya.

5.3 MENGAMANKAN BASIS DATA MYSQL

Keamanan itu penting, terutama saat Anda mulai menghubungkan basis data MySQL ke situs Web Anda. Di bagian ini, kita akan melihat tindakan pencegahan yang harus Anda ambil untuk melindungi basis data Anda.

MySQL dari Sudut Pandang Sistem Operasi

Menjalankan server MySQL (mysqld) sebagai root adalah ide yang buruk jika Anda menjalankan sistem operasi mirip UNIX. Hal ini memberikan hak penuh kepada pengguna MySQL untuk membaca dan menulis file di mana saja dalam sistem operasi. Ini adalah poin penting yang mudah diabaikan, yang terkenal digunakan untuk meretas situs web Apache. (Untungnya, para peretas itu adalah "topi putih" [orang baik], dan satu-satunya tindakan yang mereka lakukan adalah memperketat keamanan.)

Sebaiknya Anda menyiapkan pengguna MySQL khusus untuk tujuan ini. Selain itu, Anda kemudian dapat membuat direktori (tempat data fisik disimpan) hanya dapat diakses oleh pengguna MySQL. Dalam banyak instalasi, server disiapkan untuk berjalan sebagai userid mysql, dalam grup mysql.

Anda juga sebaiknya menyiapkan server MySQL di balik firewall. Dengan cara ini Anda dapat menghentikan koneksi dari mesin yang tidak sah—periksa dan lihat apakah Anda dapat terhubung dari luar ke server Anda pada port nomor 3306. Ini adalah port default tempat MySQL berjalan, dan harus ditutup pada firewall Anda.

Kata Sandi

Pastikan semua pengguna Anda memiliki kata sandi (terutama root!) dan kata sandi tersebut dipilih dengan baik dan diubah secara berkala, seperti kata sandi sistem operasi. Aturan dasar yang perlu diingat di sini adalah bahwa kata sandi yang merupakan atau berisi kata-kata dari kamus adalah ide yang buruk. Kombinasi huruf dan angka adalah yang terbaik.

Jika Anda akan menyimpan kata sandi dalam file skrip, pastikan hanya pengguna yang kata sandinya disimpan yang dapat melihat skrip tersebut. Dua tempat utama terjadinya hal ini adalah

1. Dalam skrip `mysql.server`, Anda mungkin perlu menggunakan kata sandi root `UNIX`. Jika demikian halnya, pastikan hanya root yang dapat membaca skrip ini.
2. Dalam skrip PHP yang digunakan untuk terhubung ke basis data, Anda perlu menyimpan kata sandi untuk pengguna tersebut. Hal ini dapat dilakukan dengan aman dengan meletakkan login dan kata sandi dalam sebuah berkas yang disebut, misalnya, `dbconnect.php`, yang kemudian Anda sertakan saat diperlukan. Skrip ini dapat disimpan di luar pohon dokumen Web dan hanya dapat diakses oleh pengguna yang sesuai. Ingatlah bahwa jika Anda meletakkan detail ini dalam berkas .inc atau beberapa berkas ekstensi lain di pohon Web, Anda harus berhati-hati untuk memeriksa apakah

server Web Anda mengetahui berkas ini harus ditafsirkan sebagai PHP sehingga detailnya tidak dapat dilihat di peramban Web.

Jangan simpan kata sandi dalam teks biasa di basis data Anda. Kata sandi MySQL tidak disimpan dengan cara itu, tetapi umumnya dalam aplikasi Web Anda juga ingin menyimpan nama login dan kata sandi anggota situs Web. Anda dapat mengenkripsi kata sandi (satu arah) menggunakan fungsi `PASSWORD()` atau `MD5()` MySQL. Ingatlah bahwa jika Anda `INSERT` kata sandi dalam salah satu format ini saat Anda menjalankan `SELECT` (untuk mencoba dan memasukkan pengguna), Anda perlu menggunakan fungsi yang sama lagi untuk memeriksa kata sandi yang diketik pengguna.

Hak Istimewa Pengguna

Pengetahuan adalah kekuatan. Pastikan Anda memahami sistem hak istimewa MySQL, dan konsekuensi pemberian hak istimewa tertentu. Jangan berikan hak istimewa lebih kepada pengguna mana pun daripada yang dibutuhkannya. Anda harus memeriksanya dengan melihat tabel pemberian hak istimewa.

Secara khusus, jangan berikan hak istimewa `PROCESS`, `FILE`, `SHUTDOWN`, dan `RELOAD` kepada pengguna mana pun selain administrator kecuali benar-benar diperlukan. Hak istimewa `PROCESS` dapat digunakan untuk melihat apa yang dilakukan dan diketik pengguna lain, termasuk kata sandi mereka. Hak istimewa `FILE` dapat digunakan untuk membaca dan menulis file ke dan dari sistem operasi (termasuk, misalnya, /etc/password pada sistem UNIX).

Hak istimewa `GRANT` juga harus diberikan dengan hati-hati karena ini memungkinkan pengguna untuk berbagi hak istimewa mereka dengan orang lain. Pastikan bahwa saat Anda menyiapkan pengguna, Anda hanya memberi mereka akses dari host tempat mereka akan terhubung. Jika Anda memiliki `jane@localhost` sebagai pengguna, itu tidak masalah, tetapi plain jane cukup umum dan dapat masuk dari mana saja—dan dia mungkin bukan jane yang Anda kira. Hindari penggunaan karakter pengganti dalam nama host karena alasan yang sama.

Anda dapat lebih meningkatkan keamanan dengan menggunakan IP daripada nama domain dalam tabel host Anda. Ini menghindari masalah dengan kesalahan atau peretas di DNS Anda. Anda dapat menerapkannya dengan memulai daemon MySQL dengan opsi --skip-name-resolve, yang berarti bahwa semua nilai kolom host harus berupa alamat IP atau localhost.

Alternatif lain adalah memulai mysqld dengan opsi --secure. Ini memeriksa IP yang telah diselesaikan untuk melihat apakah mereka menyelesaikan kembali ke nama host yang diberikan. (Ini aktif secara default dari versi 3.22 dan seterusnya.)

Anda juga harus mencegah pengguna non-administratif memiliki akses ke program mysqladmin di server Web Anda. Karena ini berjalan dari baris perintah, ini adalah masalah hak istimewa sistem operasi.

Masalah Web

Saat Anda menghubungkan basis data MySQL ke Web, hal itu menimbulkan beberapa masalah keamanan khusus. Bukan ide yang buruk untuk memulai dengan menyiapkan pengguna khusus hanya untuk tujuan koneksi Web. Dengan cara ini Anda dapat memberi mereka hak istimewa minimum yang diperlukan dan tidak memberikan, misalnya, hak

istimewa DROP, ALTER, atau CREATE kepada pengguna tersebut. Anda dapat memberikan SELECT hanya pada tabel katalog, dan INSERT hanya pada tabel pesanan. Sekali lagi, ini adalah ilustrasi tentang cara menggunakan prinsip hak istimewa paling rendah.

Anda harus selalu memeriksa semua data yang masuk dari pengguna. Bahkan jika formulir HTML Anda terdiri dari kotak pilihan dan tombol radio, seseorang mungkin mengubah URL untuk mencoba memecahkan skrip Anda. Ada baiknya juga memeriksa ukuran data yang masuk.

Jika pengguna mengetikkan kata sandi atau data rahasia untuk disimpan dalam basis data Anda, ingatlah bahwa data tersebut akan dikirimkan dari browser ke server dalam bentuk teks biasa kecuali Anda menggunakan SSL (Secure Sockets Layer). Kita akan membahas penggunaan SSL secara lebih rinci nanti.

5.4 INFORMASI LEBIH LANJUT TENTANG BASIS DATA

Sejauh ini, kita telah menggunakan SHOW dan DESCRIBE untuk mengetahui tabel apa saja yang ada dalam basis data dan kolom apa saja yang ada di dalamnya. Kita akan melihat sekilas bagaimana cara lain keduanya dapat digunakan, dan penggunaan pernyataan EXPLAIN untuk mendapatkan informasi lebih lanjut tentang cara SELECT dilakukan.

Mendapatkan Informasi dengan SHOW

Sebelumnya kita telah menggunakan

```
SHOW TABLES;
```

untuk mendapatkan daftar tabel dalam basis data.

Pernyataan

```
show databases;
```

akan menampilkan daftar basis data yang tersedia. Anda kemudian dapat menggunakan pernyataan SHOW TABLES untuk melihat daftar tabel dalam salah satu basis data tersebut:

```
show tables from books;
```

Ketika Anda menggunakan SHOW TABLES tanpa menentukan basis data, secara default akan menggunakan basis data yang sedang digunakan. Ketika Anda mengetahui tabel apa saja yang ada, Anda dapat memperoleh daftar kolom:

```
show columns from orders from books;
```

Jika Anda tidak mengaktifkan parameter basis data, pernyataan SHOW COLUMNS akan secara default menggunakan basis data yang sedang digunakan. Anda juga dapat menggunakan notasi table.column:

```
show columns from books.orders;
```

Satu variasi lain yang sangat berguna dari pernyataan SHOW dapat digunakan untuk melihat hak istimewa yang dimiliki pengguna. Misalnya, jika kita menjalankan yang berikut ini, kita akan memperoleh output yang ditunjukkan pada Gambar 5.1:
show grants for bookorama;

```
+-----------------------------------------------------------------------------------------------------+
| Grants for bookorama@%                                                                              |
+-----------------------------------------------------------------------------------------------------+
|GRANT USAGE ON *.* TO 'bookorama'@'%' IDENTIFIED BY PASSWORD '6a87b6810cb073de'                      |
|GRANT SELECT, INSERT, UPDATE, DELETE, CREATE, DROP, INDEX, ALTER ON books.* TO 'bookorama'@'%' |
+-----------------------------------------------------------------------------------------------------+
```

Gambar 5.1 Output dari pernyataan SHOW GRANTS.

Pernyataan GRANT yang ditampilkan belum tentu merupakan pernyataan yang dijalankan untuk memberikan hak istimewa kepada pengguna tertentu, tetapi merupakan ringkasan pernyataan setara yang akan menghasilkan tingkat hak istimewa pengguna saat ini.

Ada banyak variasi lain dari pernyataan SHOW. Ringkasan semua variasi ditunjukkan pada Tabel 5.6.

Tabel 5.6 Sintaks Pernyataan SHOW

Variasi	*Deskripsi*
SHOW DATABASES [LIKE *database*]	Mencantumkan basis data yang tersedia, secara opsional dengan nama seperti basis data.
SHOW TABLES [FROM *database*] [LIKE *table*]	Mencantumkan tabel dari basis data yang sedang digunakan, atau dari basis data yang disebut database jika ditentukan, secara opsional dengan nama tabel seperti tabel.
SHOW COLUMNS FROM *table* [FROM *database*] [LIKE *column*]	Mencantumkan semua kolom dalam tabel tertentu dari database yang sedang digunakan, atau dari database yang ditentukan, secara opsional dengan nama kolom seperti column. Anda dapat menggunakan SHOW FIELDS alih-alih SHOW COLUMNS.
SHOW INDEX FROM *table* [FROM *database*]	Menampilkan detail semua indeks pada tabel tertentu dari database yang sedang digunakan, atau dari database yang disebut database jika ditentukan. Anda dapat menggunakan SHOW KEYS sebagai gantinya.
SHOW STATUS [LIKE *status_item*]	Memberikan informasi tentang sejumlah item sistem, seperti jumlah thread yang berjalan. Klausa LIKE digunakan untuk mencocokkan dengan nama item ini, jadi, misalnya, 'Thread%' cocok dengan item 'Threads_cached', 'Threads_connected', dan 'Threads_running'.

SHOW VARIABLES [LIKE *variable_name*]	Menampilkan nama dan nilai variabel sistem MySQL, seperti nomor versi. Klausa LIKE dapat digunakan untuk mencocokkannya dengan cara yang mirip dengan SHOW STATUS.
SHOW [FULL] PROCESSLIST	Menampilkan semua proses yang berjalan dalam sistem, yaitu kueri yang sedang dijalankan. Sebagian besar pengguna akan melihat utas mereka sendiri, tetapi jika mereka memiliki hak istimewa PROCESS, mereka dapat melihat proses semua orang, termasuk kata sandi jika ada dalam kueri. Kueri dipotong menjadi 100 karakter secara default. Menggunakan kata kunci opsional FULL akan menampilkan kueri lengkap.
SHOW TABLE STATUS [FROM *database*] [LIKE *database*]	Menampilkan informasi tentang setiap tabel dalam basis data yang sedang digunakan, atau basis data yang disebut basis data jika ditentukan, secara opsional dengan pencocokan karakter pengganti. Informasi ini mencakup jenis tabel dan kapan setiap tabel terakhir diperbarui.
SHOW GRANTS FOR *user*	Menunjukkan pernyataan GRANT yang diperlukan untuk memberikan pengguna yang ditentukan dalam user tingkat hak istimewanya saat ini.

Mendapatkan Informasi Tentang Kolom dengan DESCRIBE

Sebagai alternatif untuk pernyataan SHOW COLUMNS, Anda dapat menggunakan pernyataan DESCRIBE, mirip dengan pernyataan DESCRIBE di Oracle (RDBMS lain). Sintaks dasarnya adalah DESCRIBE table [kolom];

Ini akan memberikan informasi tentang semua kolom dalam tabel atau kolom tertentu jika kolom ditentukan. Anda dapat menggunakan karakter pengganti dalam nama kolom jika Anda suka.

Memahami Cara Kerja Kueri dengan EXPLAIN

Pernyataan EXPLAIN dapat digunakan dengan dua cara. Pertama, Anda dapat menggunakan

```
EXPLAIN tabel;
```

Ini memberikan output yang sangat mirip dengan tabel DESCRIBE atau SHOW COLUMNS FROM tabel.

Cara kedua dan lebih menarik yang dapat Anda gunakan untuk menggunakan EXPLAIN memungkinkan Anda untuk melihat dengan tepat bagaimana MySQL mengevaluasi kueri SELECT. Untuk menggunakannya dengan cara ini, cukup letakkan kata explain di depan pernyataan SELECT. Anda dapat menggunakan pernyataan EXPLAIN saat Anda mencoba menjalankan kueri yang rumit dan jelas belum melakukannya dengan benar, atau saat kueri

membutuhkan waktu lebih lama untuk diproses daripada yang seharusnya. Jika Anda menulis kueri yang rumit, Anda dapat memeriksanya terlebih dahulu dengan menjalankan perintah EXPLAIN sebelum Anda benar-benar menjalankan kueri. Dengan output dari pernyataan ini, Anda dapat mengerjakan ulang SQL Anda untuk mengoptimalkannya jika perlu. Ini juga merupakan alat pembelajaran yang praktis.

Misalnya, coba jalankan kueri berikut pada basis data Book-O-Rama. Ini menghasilkan output yang ditunjukkan pada Gambar 5.2.

```
explain
select customers.name
from customers, orders, order_items, books
where customers.customerid = orders.customerid
and orders.orderid = order_items.orderid
and order_items.isbn = books.isbn
and books.title like '%Java%';
```

```
+-------------+--------+---------------+---------+---------+------------------+------+-------------+
| table       | type   | possible_keys | key     | key_len | ref              |rows | Extra       |
+-------------+--------+---------------+---------+---------+------------------+------+-------------+
| orders      | ALL    | PRIMARY       | NULL    |    NULL | NULL             |   4 |             |
| order_items | ref    | PRIMARY       | PRIMARY |       4 | orders.orderid   |   1 | Using index |
| customers   | ALL    | PRIMARY       | NULL    |    NULL | NULL             |   3 | where used  |
| books       | eq_ref | PRIMARY       | PRIMARY |      13 | order_items.isbn |   1 | where used  |
+-------------+--------+---------------+---------+---------+------------------+------+-------------+
```

Gambar 5.2 Output dari pernyataan EXPLAIN.

Hal ini mungkin terlihat membingungkan pada awalnya, tetapi bisa sangat berguna. Mari kita lihat kolom-kolom dalam tabel ini satu per satu. Kolom pertama, tabel, hanya mencantumkan tabel yang digunakan untuk menjawab kueri. Setiap baris dalam hasil memberikan informasi lebih lanjut tentang bagaimana tabel tertentu digunakan dalam kueri ini. Dalam kasus ini, Anda dapat melihat bahwa tabel yang digunakan adalah `orders`, `order_items`, `customers`, dan `books`. (Kita sudah mengetahuinya dengan melihat kueri.)

Kolom type menjelaskan bagaimana tabel digunakan dalam join dalam kueri. Kumpulan nilai yang dapat dimiliki kolom ini ditunjukkan dalam Tabel 5.7. Nilai-nilai ini dicantumkan dalam urutan dari yang tercepat hingga yang paling lambat dalam hal eksekusi kueri. Hal ini memberi Anda gambaran tentang berapa banyak baris yang perlu dibaca dari setiap tabel untuk mengeksekusi kueri.

Tabel 5.7 Kemungkinan Jenis Join seperti yang Ditunjukkan dalam Output dari `EXPLAIN`

Tipe	***Deskripsi***
`const or system`	Tabel hanya dibaca satu kali. Hal ini terjadi jika tabel memiliki tepat satu baris. Tipe sistem digunakan jika tabel tersebut merupakan tabel sistem, dan tipe const jika tidak.
`eq_ref`	Untuk setiap set baris dari tabel lain dalam gabungan, kita membaca satu baris dari tabel ini. Ini digunakan saat gabungan menggunakan semua bagian indeks pada tabel, dan indeksnya UNIK atau merupakan kunci utama.

ref	Untuk setiap set baris dari tabel lain dalam gabungan, kita membaca satu set baris dari tabel ini yang semuanya cocok. Ini digunakan saat gabungan tidak dapat memilih satu baris berdasarkan kondisi gabungan, yaitu, saat hanya sebagian kunci yang digunakan dalam gabungan, atau jika bukan UNIQUE atau kunci utama.
range	Untuk setiap set baris dari tabel lain dalam gabungan, kita membaca satu set baris dari tabel ini yang termasuk dalam rentang tertentu.
indeks	Seluruh indeks dipindai.
ALL	Setiap baris dalam tabel dipindai

Pada contoh sebelumnya, Anda dapat melihat bahwa salah satu tabel digabungkan menggunakan eq_ref (books), dan satu digabungkan menggunakan ref (order_items), tetapi dua lainnya (orders dan customers) digabungkan menggunakan ALL; yaitu, dengan melihat setiap baris dalam tabel.

Kolom rows mendukung hal ini—kolom ini mencantumkan (secara kasar) jumlah baris setiap tabel yang harus dipindai untuk melakukan penggabungan. Anda dapat mengalikan keduanya untuk mendapatkan jumlah total baris yang diperiksa saat kueri dilakukan. Kami mengalikan angka-angka ini karena penggabungan seperti produk dari baris-baris dalam tabel yang berbeda—lihat Bab 3, "Bekerja dengan Basis Data MySQL," untuk detailnya. Ingatlah bahwa ini adalah jumlah baris yang diperiksa, bukan jumlah baris yang dikembalikan, dan ini hanya perkiraan—MySQL tidak dapat mengetahui jumlah pastinya tanpa melakukan kueri.

Tentu saja, semakin kecil angka ini, semakin baik. Saat ini kami memiliki jumlah data yang cukup kecil dalam basis data, tetapi ketika basis data mulai bertambah besar, kueri ini akan berhenti saat dijalankan. Kami akan membahasnya lagi sebentar lagi.

Kolom possible_keys mencantumkan, seperti yang Anda duga, kunci yang mungkin digunakan MySQL untuk menggabungkan tabel. Dalam kasus ini, Anda dapat melihat bahwa semua kunci yang mungkin adalah PRIMARY KEY.

Kolom kunci adalah kunci dari tabel yang sebenarnya digunakan MySQL, atau NULL jika tidak ada kunci yang digunakan. Anda akan melihat bahwa, meskipun ada kemungkinan PRIMARY KEY untuk tabel orders dan customers, kunci tersebut tidak digunakan dalam kueri ini. Kami akan melihat cara memperbaikinya sebentar lagi.

Kolom key_len menunjukkan panjang kunci yang digunakan. Anda dapat menggunakan ini untuk mengetahui apakah hanya sebagian dari kunci yang digunakan. Ini relevan ketika Anda memiliki kunci yang terdiri dari lebih dari satu kolom. Dalam kasus ini, di mana kunci digunakan (order_items dan books), kunci lengkap digunakan. Kolom ref menunjukkan kolom yang digunakan dengan kunci untuk memilih baris dari tabel.

Terakhir, kolom Extra memberi tahu Anda informasi lain tentang bagaimana penggabungan dilakukan. Nilai yang mungkin Anda lihat di kolom ini ditunjukkan pada Tabel 5.8.

Tabel 5.8 Nilai yang Mungkin untuk Kolom Extra seperti yang Ditunjukkan dalam Output dari `EXPLAIN`

Nilai	***Katerangan***
Not exist	Kueri telah dioptimalkan untuk menggunakan `LEFT JOIN`.
Range checked for each record	Untuk setiap baris dalam kumpulan baris dari tabel lain dalam gabungan, cobalah temukan indeks terbaik untuk digunakan, jika ada.
Using filesort	Diperlukan dua kali lintasan untuk mengurutkan data. (Ini jelas memerlukan waktu dua kali lebih lama.)
Using Indeks	Semua informasi dari tabel berasal dari indeks-dengan kata lain, baris-barisnya tidak benar-benar dicari.
Using temporary	Tabel sementara perlu dibuat untuk menjalankan kueri ini.
WHERE used	Klausa `WHERE` digunakan untuk memilih baris.

Ada beberapa cara untuk memperbaiki masalah yang Anda temukan dalam output dari `EXPLAIN`. *Pertama*, periksa jenis kolom dan pastikan semuanya sama. Hal ini berlaku khususnya untuk lebar kolom. Indeks tidak dapat digunakan untuk mencocokkan kolom jika lebarnya berbeda. Anda dapat memperbaikinya dengan mengubah jenis kolom yang akan dicocokkan, atau membangunnya dalam desain Anda sejak awal.

Kedua, Anda dapat memberi tahu pengoptimal gabungan untuk memeriksa distribusi kunci dan karenanya mengoptimalkan gabungan dengan lebih efisien menggunakan utilitas myisamchk. Anda dapat menjalankannya dengan mengetik:

```
>myisamchk --analyze pathtomysqldatabase/table
```

Anda dapat memeriksa beberapa tabel dengan mencantumkan semuanya pada baris perintah, atau dengan menggunakan

```
>myisamchk --analyze pathtomysqldatabase/*.MYI
```

Anda dapat memeriksa semua tabel di semua basis data dengan menjalankan perintah berikut, yang akan menghasilkan output yang ditunjukkan pada Gambar 5.3:

```
>myisamchk --analyze pathtomysqldatadirectory/*/*.MYI
```

```
+-------------+--------+---------------+---------+---------+---------------------+------+--------------------------+
| table       | type   | possible_keys | key     | key_len | ref _______________ | rows | Extra                    |
+-------------+--------+---------------+---------+---------+---------------------+------+--------------------------+
| books       | ALL    | PRIMARY       | NULL    |    NULL | NULL                |    4 | where used               |
| order_items | index  | PRIMARY       | PRIMARY |      17 | NULL                |    5 | where used; Using index |
| orders      | eq_ref | PRIMARY       | PRIMARY |       4 | order_items.orderid |    1 |                          |
| customers   | eq_ref | PRIMARY       | PRIMARY |       4 | orders.customerid   |    1 |                          |
+-------------+--------+---------------+---------+---------+---------------------+------+--------------------------+
```

Gambar 5.3 Ini adalah output dari EXPLAIN setelah menjalankan myisamchk.

Anda akan melihat bahwa cara kueri dievaluasi telah banyak berubah. Sekarang kita hanya menggunakan semua baris di salah satu tabel (books), yang tidak masalah. Secara khusus, sekarang kita menggunakan eq_ref untuk dua tabel dan index untuk yang lain. MySQL juga sekarang menggunakan seluruh kunci untuk order_items (17 karakter dibandingkan dengan 4 karakter sebelumnya).

Anda juga akan melihat jumlah baris yang digunakan sebenarnya telah meningkat. Ini mungkin disebabkan oleh fakta bahwa kita memiliki sedikit data dalam basis data aktual saat ini. Ingatlah bahwa jumlah baris yang tercantum hanyalah perkiraan—coba lakukan kueri aktual dan periksa ini. Jika angka-angka ini sangat meleset, manual MySQL menyarankan untuk menggunakan straight join dan mencantumkan tabel dalam klausa FROM Anda dalam urutan yang berbeda.

Ketiga, Anda mungkin ingin mempertimbangkan untuk menambahkan indeks baru ke tabel. Jika kueri ini a) lambat, dan b) umum, Anda harus mempertimbangkan ini dengan serius. Jika ini adalah permintaan satu kali yang tidak akan pernah Anda gunakan lagi, seperti laporan tidak jelas yang diminta sekali, maka tidak akan ada gunanya, karena akan memperlambat hal-hal lain. Kita akan melihat cara melakukannya di bagian berikutnya.

5.5 MEMPERCEPAT KUERI DENGAN INDEKS

Jika Anda berada dalam situasi yang disebutkan sebelumnya, di mana kolom possible_keys dari EXPLAIN berisi beberapa nilai NULL, Anda mungkin dapat meningkatkan kinerja kueri Anda dengan menambahkan indeks ke tabel yang dimaksud. Jika kolom yang Anda gunakan dalam klausa WHERE cocok untuk pengindeksan, Anda dapat membuat indeks baru untuknya menggunakan ALTER TABLE seperti ini:

```
ALTER TABLE table ADD INDEX (column);
```

Kiat Optimasi Umum

Selain kiat optimasi kueri sebelumnya, ada beberapa hal yang dapat Anda lakukan untuk meningkatkan kinerja basis data MySQL Anda secara umum.

Optimalisasi Desain

Pada dasarnya Anda ingin semua hal dalam basis data Anda sekecil mungkin. Anda dapat mencapainya sebagian dengan desain yang layak yang meminimalkan redundansi. Anda juga dapat mencapainya dengan menggunakan tipe data sekecil mungkin untuk kolom. Anda juga harus meminimalkan NULL sedapat mungkin, dan membuat kunci utama Anda sesingkat mungkin. Hindari kolom dengan panjang variabel jika memungkinkan (seperti VARCHAR, TEXT, dan BLOB). Jika tabel Anda memiliki kolom dengan panjang tetap, kolom tersebut akan lebih cepat digunakan tetapi mungkin akan menghabiskan lebih banyak ruang.

Izin

Selain menggunakan saran yang disebutkan di bagian sebelumnya pada EXPLAIN, Anda dapat meningkatkan kecepatan kueri dengan menyederhanakan izin Anda. Sebelumnya, kami

membahas cara kueri diperiksa dengan sistem izin sebelum dieksekusi. Semakin sederhana proses ini, semakin cepat kueri Anda akan berjalan.

Optimalisasi Tabel

Jika tabel telah digunakan selama beberapa waktu, data dapat terfragmentasi saat pembaruan dan penghapusan diproses. Hal ini akan menambah waktu yang dibutuhkan untuk menemukan hal-hal dalam tabel ini. Anda dapat memperbaikinya dengan menggunakan pernyataan:

```
OPTIMIZE TABLE tablename;
```

atau dengan mengetik

```
>myisamchk -r table
```

pada prompt perintah.

Anda juga dapat menggunakan utilitas `myisamchk` untuk mengurutkan indeks tabel dan data menurut indeks tersebut, seperti ini:

```
>myisamchk --sort-index --sort-records=1 pathtomysqldatadirectory/*/*.MYI
```

Menggunakan Indeks

Gunakan indeks jika diperlukan untuk mempercepat kueri Anda. Buatlah sederhana, dan jangan buat indeks yang tidak digunakan oleh kueri Anda. Anda dapat memeriksa indeks mana yang digunakan dengan menjalankan `EXPLAIN` seperti yang ditunjukkan sebelumnya.

Gunakan Nilai Default

Jika memungkinkan, gunakan nilai default untuk kolom, dan masukkan data hanya jika berbeda dari default. Ini mengurangi waktu yang dibutuhkan untuk menjalankan pernyataan `INSERT`.

Gunakan Koneksi Persisten

Kiat pengoptimalan khusus ini berlaku khususnya untuk basis data Web. Kami telah membahasnya di tempat lain jadi ini hanya pengingat.

Kiat Lainnya

Ada banyak perubahan kecil lainnya yang dapat Anda lakukan untuk meningkatkan kinerja dalam situasi tertentu dan saat Anda memiliki kebutuhan khusus. Situs Web MySQL menawarkan serangkaian kiat tambahan yang bagus. Anda dapat menemukannya di http://www.mysql.com

5.6 BERBAGAI JENIS TABEL

Satu hal terakhir yang berguna untuk dibahas sebelum kita meninggalkan MySQL untuk sementara waktu adalah keberadaan berbagai jenis tabel. Anda dapat memilih jenis tabel saat membuat tabel, menggunakan

```
CREATE TABLE table TYPE=type ....
```

Jenis tabel yang mungkin adalah

- MyISAM. Ini adalah default, dan yang telah kami gunakan hingga saat ini. Ini didasarkan pada ISAM, yang merupakan singkatan dari *Indexed Sequential Access Method,* metode standar untuk menyimpan rekaman dan file.
- HEAP. Tabel jenis ini disimpan dalam memori, dan indeksnya di-hash. Ini membuat tabel HEAP sangat cepat, tetapi, jika terjadi crash, data Anda akan hilang. Karakteristik ini membuat tabel HEAP ideal untuk menyimpan data sementara atau data turunan. Anda harus menentukan MAX_ROWS dalam pernyataan CREATE TABLE, atau tabel ini dapat menghabiskan semua memori Anda. Selain itu, mereka tidak dapat memiliki kolom BLOB, TEXT, atau AUTO INCREMENT.
- BDB. Tabel-tabel ini aman untuk transaksi; artinya, tabel-tabel ini menyediakan kapabilitas COMMIT dan ROLLBACK. Tabel-tabel ini lebih lambat digunakan daripada tabel MyISAM, dan didasarkan pada Berkeley DB. Pada saat artikel ini ditulis, tabel-tabel ini masih dalam tahap debug di MySQL versi 3.23.21, dan akan memerlukan unduhan tambahan agar dapat digunakan, yang tersedia di situs Web MySQL.

Tipe tabel tambahan ini dapat berguna saat Anda menginginkan kecepatan ekstra atau keamanan transaksi.

Memuat Data dari Berkas

Salah satu fitur MySQL yang bermanfaat yang belum kita bahas adalah pernyataan LOAD DATA INFILE. Pernyataan ini dapat digunakan untuk memuat data tabel dari berkas. Pernyataan ini dijalankan dengan sangat cepat.

Ini adalah perintah yang fleksibel dengan banyak opsi, tetapi penggunaan umumnya adalah seperti berikut:

```
LOAD DATA INFILE “newbooks.txt” INTO TABLE books;
```

Ini akan membaca data baris dari berkas newbooks.txt ke dalam tabel books. Secara default, bidang data dalam berkas harus dipisahkan oleh tab dan diapit tanda kutip tunggal, dan setiap baris harus dipisahkan oleh baris baru (\n). Karakter khusus harus di-escape dengan garis miring (\). Semua karakteristik ini dapat dikonfigurasi dengan berbagai opsi pernyataan LOAD—lihat manual MySQL untuk detail selengkapnya.

Untuk menggunakan pernyataan LOAD DATA INFILE, pengguna harus memiliki hak istimewa FILE yang dibahas sebelumnya.

BAB 6
MENJALANKAN SITUS E-COMMERCE

Bab ini memperkenalkan beberapa masalah yang terlibat dalam menentukan, merancang, membangun, dan memelihara situs e-dagang secara efektif. Kami akan memeriksa rencana Anda, kemungkinan risiko, dan beberapa cara untuk membuat situs web menghasilkan keuntungan sendiri.

Pembahasan dalam buku ini

- Apa yang ingin Anda capai dengan situs e-dagang Anda
- Jenis situs web komersial
- Risiko dan ancaman
- Menentukan strategi

Sebelum menghabiskan terlalu banyak waktu untuk mengkhawatirkan detail implementasi situs web Anda, Anda harus memiliki tujuan yang pasti dalam pikiran, dan rencana yang cukup terperinci untuk mencapai tujuan tersebut.

Dalam buku ini, kami berasumsi bahwa Anda sedang membangun situs web komersial. Jadi, mungkin menghasilkan uang adalah salah satu tujuan Anda. Ada banyak cara untuk mengambil pendekatan komersial terhadap Internet. Mungkin Anda ingin mengiklankan layanan offline Anda atau menjual produk dunia nyata secara online. Mungkin Anda memiliki produk yang dapat dijual dan disediakan secara online. Mungkin situs Anda tidak secara langsung ditujukan untuk menghasilkan pendapatan, tetapi sebaliknya mendukung aktivitas offline atau bertindak sebagai alternatif yang lebih murah untuk aktivitas yang ada.

6.1 JENIS SITUS WEB KOMERSIAL

Situs Web Komersial umumnya melakukan satu atau beberapa aktivitas berikut:

- Memublikasikan informasi perusahaan melalui brosur online
- Menerima pesanan barang atau jasa
- Menyediakan jasa atau barang digital
- Menambah nilai pada barang atau jasa
- Memotong biaya

Bagian dari banyak situs Web akan sesuai dengan lebih dari satu kategori ini. Berikut ini adalah deskripsi dari setiap kategori, dan cara umum untuk membuat masing-masing kategori menghasilkan pendapatan atau manfaat lain bagi organisasi Anda.

Tujuan dari bagian buku ini adalah untuk membantu Anda merumuskan tujuan Anda. Mengapa Anda menginginkan situs Web? Bagaimana setiap fitur yang dibangun di situs Web Anda akan berkontribusi pada bisnis Anda?

Brosur Daring

Hampir semua situs web komersial pada awal tahun 1990-an hanyalah brosur daring atau alat penjualan. Jenis situs ini masih merupakan bentuk situs web komersial yang paling

umum. Baik sebagai langkah awal ke Web, atau sebagai latihan periklanan berbiaya rendah, jenis situs ini masuk akal bagi banyak bisnis.

Situs brosur dapat berupa apa saja, mulai dari kartu nama yang ditampilkan sebagai halaman Web hingga kumpulan informasi pemasaran yang luas. Bagaimanapun, tujuan situs, dan alasan finansial keberadaannya, adalah untuk menarik pelanggan agar menghubungi bisnis Anda.

Jenis situs ini tidak menghasilkan pendapatan secara langsung, tetapi dapat menambah pendapatan yang diterima bisnis Anda melalui cara tradisional. Mengembangkan situs seperti ini menghadirkan sedikit tantangan teknis. Masalah yang dihadapi serupa dengan yang dihadapi dalam latihan pemasaran lainnya. Beberapa kesalahan umum pada situs jenis ini meliputi:

- Tidak menyediakan informasi penting
- Presentasi yang buruk
- Tidak menanggapi umpan balik yang diberikan oleh situs
- Membiarkan situs menua
- Tidak melacak keberhasilan situs
- Tidak Menyediakan Informasi Penting

Apa yang mungkin dicari pengunjung saat mengunjungi situs Anda? Bergantung pada seberapa banyak yang sudah mereka ketahui, mereka mungkin menginginkan spesifikasi produk yang terperinci, atau mereka mungkin hanya menginginkan informasi yang sangat mendasar seperti detail kontak.

Banyak situs web tidak menyediakan informasi yang berguna, atau tidak menyediakan informasi penting. Paling tidak, situs Anda perlu memberi tahu pengunjung apa yang Anda lakukan, wilayah geografis yang dilayani bisnis Anda, dan cara menghubungi mereka.

Presentasi yang Buruk

"Di Internet, tidak ada yang tahu Anda seekor anjing," atau begitulah pepatah lama.1 Sama seperti bisnis kecil, atau anjing, dapat terlihat lebih besar dan lebih mengesankan saat mereka menggunakan Internet, bisnis besar dapat terlihat kecil, tidak profesional, dan tidak mengesankan dengan situs web yang buruk.

Terlepas dari ukuran perusahaan Anda, pastikan situs web Anda memiliki standar yang tinggi. Teks harus ditulis dan diperiksa oleh seseorang yang sangat memahami bahasa yang digunakan. Grafik harus bersih, jernih, dan cepat diunduh. Di situs bisnis, Anda harus mempertimbangkan dengan saksama penggunaan grafik dan warna, dan pastikan semuanya sesuai dengan gambar yang ingin Anda tampilkan. Gunakan animasi dan suara dengan saksama jika memang diperlukan.

Meskipun Anda tidak akan dapat membuat situs Anda terlihat sama di semua mesin, sistem operasi, dan browser, pastikan situs tersebut dapat dilihat dan tidak memberikan kesalahan kepada sebagian besar pengguna.

Tidak Menjawab Umpan Balik yang Dihasilkan oleh Situs Web

Layanan pelanggan yang baik sama pentingnya dalam menarik dan mempertahankan pelanggan di Web seperti di dunia luar. Perusahaan besar dan kecil sering kali mencantumkan

alamat email di halaman Web, lalu mengabaikan untuk memeriksa atau menjawab email tersebut dengan segera.

Orang-orang memiliki ekspektasi yang berbeda tentang waktu respons email dibandingkan dengan surat pos. Jika Anda tidak memeriksa dan menanggapi email setiap hari, orang akan percaya bahwa pertanyaan mereka tidak penting bagi Anda.

Alamat email pada halaman Web biasanya harus generik, ditujukan ke jabatan atau departemen, bukan ke orang tertentu. Apa yang akan terjadi pada email yang dikirim ke fred.smith@company.com saat Fred keluar? Email yang ditujukan ke sales@company.com kemungkinan besar akan diteruskan ke penggantinya. Email juga dapat dikirimkan ke sekelompok orang, yang dapat membantu memastikan bahwa email dijawab dengan segera.

Membiarkan Situs Menjadi Tua

Anda harus berhati-hati untuk menjaga situs Web Anda tetap segar. Konten perlu diubah secara berkala. Perubahan dalam organisasi perlu tercermin di situs. Situs yang "berisi jaring laba-laba" mencegah kunjungan berulang, dan membuat orang curiga bahwa sebagian besar informasi mungkin sekarang tidak benar.

Salah satu cara untuk menghindari situs yang basi adalah dengan memperbarui halaman secara manual. Cara lainnya adalah dengan menggunakan bahasa skrip seperti PHP untuk membuat halaman yang dinamis. Jika skrip Anda memiliki akses ke informasi terkini, skrip tersebut dapat terus-menerus menghasilkan halaman terkini.

Tidak Melacak Keberhasilan Situs

Membuat situs web memang baik dan bagus, tetapi bagaimana Anda membenarkan upaya dan biaya tersebut? Terutama jika situs tersebut untuk perusahaan besar, akan tiba saatnya Anda diminta untuk menunjukkan atau mengukur nilainya bagi organisasi.

Untuk kampanye pemasaran tradisional, organisasi besar menghabiskan puluhan ribu dolar untuk riset pasar, baik sebelum meluncurkan kampanye maupun setelah kampanye untuk mengukur efektivitasnya. Bergantung pada skala dan anggaran usaha web Anda, langkah-langkah ini mungkin sama-sama tepat untuk membantu dalam desain dan pengukuran situs Anda.

Pilihan yang lebih sederhana atau lebih murah meliputi

Memeriksa Log Server: Server web menyimpan banyak data tentang setiap permintaan dari server Anda. Sebagian besar data ini tidak berguna, dan jumlahnya yang sangat banyak membuatnya tidak berguna dalam bentuk mentahnya. Untuk menyaring file log Anda menjadi ringkasan yang bermakna, Anda memerlukan penganalisis file log.

Memantau Penjualan: Brosur daring Anda seharusnya menghasilkan penjualan. Anda harus dapat memperkirakan pengaruhnya terhadap penjualan dengan membandingkan tingkat penjualan sebelum dan sesudah peluncuran situs. Ini jelas menjadi sulit jika jenis pemasaran lain menyebabkan fluktuasi dalam periode yang sama. Meminta Umpan Balik Pengguna: Jika Anda bertanya kepada mereka, pengguna akan memberi tahu Anda apa pendapat mereka tentang situs Anda. Menyediakan formulir umpan balik atau alamat email akan mengumpulkan beberapa pendapat yang bermanfaat. Untuk meningkatkan jumlah umpan

balik, Anda mungkin ingin menawarkan sedikit bujukan, seperti mengikuti undian berhadiah untuk semua responden.

Survei Perwakilan Pengguna: Menggelar kelompok fokus dapat menjadi teknik yang efektif untuk mengevaluasi situs Anda, atau bahkan prototipe situs yang Anda inginkan. Untuk menyelenggarakan kelompok fokus, Anda hanya perlu mengumpulkan beberapa sukarelawan, mendorong mereka untuk mengevaluasi situs, lalu mewawancarai mereka untuk mengukur dan mencatat pendapat mereka.

Kelompok fokus dapat menjadi kegiatan yang mahal, yang dilakukan oleh fasilitator profesional, yang mengevaluasi dan menyaring calon peserta untuk mencoba memastikan bahwa mereka secara akurat mewakili penyebaran demografi dan kepribadian di masyarakat yang lebih luas, lalu mewawancarai peserta dengan terampil.

Kelompok fokus juga dapat tidak dikenakan biaya, dijalankan oleh seorang amatir, dan diisi oleh sampel orang-orang yang relevansinya dengan pasar sasaran tidak diketahui. Membayar perusahaan riset pasar spesialis adalah salah satu cara untuk mendapatkan kelompok fokus yang dikelola dengan baik, dan memperoleh hasil yang bermanfaat, tetapi itu bukanlah satu-satunya cara. Jika Anda menjalankan kelompok fokus Anda sendiri, pilihlah moderator yang terampil. Moderator harus memiliki keterampilan interpersonal yang baik dan tidak memiliki bias atau kepentingan dalam hasil riset. Batasi jumlah anggota kelompok menjadi enam hingga sepuluh orang. Moderator harus dibantu oleh perekam atau sekretaris agar moderator bebas untuk memfasilitasi diskusi. Hasil yang Anda peroleh dari kelompok Anda hanya akan relevan jika sampel orang yang Anda gunakan. Jika Anda mengevaluasi produk Anda hanya dengan teman dan keluarga staf Anda, mereka tidak mungkin mewakili masyarakat umum.

Menerima Pesanan Barang atau Layanan

Jika iklan daring Anda menarik, langkah logis berikutnya adalah mengizinkan pelanggan Anda memesan saat masih daring. Tenaga penjualan tradisional tahu bahwa penting untuk membuat pelanggan mengambil keputusan sekarang. Semakin banyak waktu yang Anda berikan kepada orang untuk mempertimbangkan kembali keputusan pembelian, semakin besar kemungkinan mereka akan berbelanja di tempat lain atau berubah pikiran.

Jika seorang pelanggan menginginkan produk Anda, Anda sebaiknya melakukan pembelian secepat dan semudah mungkin. Memaksa orang untuk menutup modem dan menelepon nomor telepon atau mengunjungi toko akan menjadi kendala. Jika Anda memiliki iklan daring yang telah meyakinkan pemirsa untuk membeli, biarkan mereka membeli sekarang, tanpa meninggalkan situs web Anda.

Menerima pesanan di situs web masuk akal bagi banyak bisnis. Setiap bisnis menginginkan pesanan. Memungkinkan orang untuk memesan secara daring dapat memberikan penjualan tambahan, atau mengurangi beban kerja staf penjualan Anda. Tentu saja akan ada biaya yang terlibat. Membangun situs yang dinamis, mengatur fasilitas pembayaran, dan menyediakan layanan pelanggan semuanya membutuhkan biaya. Cobalah untuk menentukan apakah produk Anda cocok untuk situs e-commerce.

Produk yang umumnya dibeli menggunakan Internet meliputi buku dan majalah, perangkat lunak dan peralatan komputer, musik, pakaian, perjalanan, dan tiket acara hiburan. Hanya karena produk Anda tidak termasuk dalam salah satu kategori ini, jangan putus asa. Kategori tersebut sudah dipenuhi merek-merek mapan. Namun, Anda sebaiknya mempertimbangkan beberapa faktor yang membuat produk ini laris manis di pasaran daring.

Idealnya, produk e-commerce tidak mudah rusak dan mudah dikirim, cukup mahal sehingga biaya pengiriman tampak masuk akal, tetapi tidak terlalu mahal sehingga pembeli merasa perlu memeriksa barang secara fisik sebelum membeli.

Produk e-commerce terbaik adalah komoditas. Jika Anda membeli alpukat, Anda mungkin ingin melihat alpukat tersebut dan mungkin menyentuhnya. Semua alpukat tidaklah sama. Satu salinan buku, CD, atau program komputer biasanya identik dengan salinan lain dengan judul yang sama. Pembeli tidak perlu melihat barang tertentu yang akan mereka beli.

Selain itu, produk e-commerce harus menarik bagi orang-orang yang menggunakan Internet. Pada saat penulisan ini, audiens ini sebagian besar terdiri dari orang dewasa muda yang bekerja, dengan pendapatan di atas rata-rata, yang tinggal di wilayah metropolitan. Namun, seiring berjalannya waktu, populasi daring mulai tampak lebih seperti keseluruhan populasi.

Beberapa produk tidak akan pernah tercermin dalam survei pembelian e-commerce, tetapi tetap sukses. Jika Anda memiliki produk yang hanya menarik bagi pasar khusus, Internet mungkin merupakan cara yang ideal untuk menjangkau pembeli.

Beberapa produk tidak mungkin berhasil sebagai kategori e-commerce. Barang murah dan mudah rusak, seperti bahan makanan, tampaknya merupakan pilihan yang buruk, meskipun hal ini tidak menghalangi perusahaan untuk mencoba, yang sebagian besar tidak berhasil. Kategori lain sangat cocok untuk situs brosur, tetapi tidak untuk pemesanan daring. Barang besar dan mahal termasuk dalam kategori ini—barang seperti kendaraan dan real estat yang memerlukan banyak penelitian sebelum membeli, tetapi terlalu mahal untuk dipesan tanpa melihatnya dan tidak praktis untuk dikirim.

Ada sejumlah kendala untuk meyakinkan calon pembeli untuk menyelesaikan pesanan. Ini termasuk:

- Pertanyaan yang tidak terjawab
- Kepercayaan
- Kemudahan penggunaan
- Kompatibilitas

Jika pengguna frustrasi dengan salah satu kendala ini, ia cenderung pergi tanpa membeli.

Pertanyaan yang Tidak Terjawab

Jika calon pelanggan tidak dapat menemukan jawaban langsung untuk salah satu pertanyaannya, ia cenderung pergi. Ini memiliki sejumlah implikasi. Pastikan situs Anda terorganisasi dengan baik. Dapatkah pengunjung baru menemukan apa yang diinginkannya dengan mudah? Pastikan situs Anda komprehensif, tanpa membebani pengunjung. Di Web, orang lebih cenderung memindai daripada membaca dengan saksama, jadi buatlah ringkas.

Untuk sebagian besar media periklanan, ada batasan praktis tentang seberapa banyak informasi yang dapat Anda berikan. Ini tidak berlaku untuk situs Web.

Untuk situs Web, dua batasan utama adalah biaya pembuatan dan pembaruan informasi dan batasan yang diberlakukan oleh seberapa baik Anda dapat mengatur, melapisi, dan menghubungkan informasi agar tidak membebani pengunjung. Sangat menggoda untuk menganggap situs web sebagai penjual otomatis yang tidak dibayar dan tidak pernah tidur, tetapi layanan pelanggan tetap penting. Dorong pengunjung untuk mengajukan pertanyaan. Cobalah untuk memberikan jawaban segera atau hampir segera melalui telepon, email, atau cara lain yang mudah.

Kepercayaan

Jika pengunjung tidak mengenal nama merek Anda, mengapa mereka harus memercayai Anda? Siapa pun dapat membuat situs web. Orang tidak perlu memercayai Anda untuk membaca situs brosur Anda, tetapi melakukan pemesanan memerlukan sejumlah kepercayaan. Bagaimana pengunjung dapat mengetahui apakah Anda adalah organisasi yang memiliki reputasi baik, atau perusahaan yang tidak memiliki reputasi baik?

Orang-orang khawatir tentang sejumlah hal saat berbelanja online:

Apa yang akan Anda lakukan dengan informasi pribadi mereka? Apakah Anda akan menjualnya kepada orang lain, menggunakannya untuk mengirimi mereka sejumlah besar iklan, atau menyimpannya di suatu tempat yang tidak aman sehingga orang lain dapat mengaksesnya? Penting untuk memberi tahu orang-orang apa yang akan dan tidak akan Anda lakukan dengan data mereka. Ini disebut kebijakan privasi dan harus mudah diakses di situs Anda.

Apakah Anda seorang pebisnis yang memiliki reputasi baik? Jika bisnis Anda terdaftar di otoritas terkait di tempat tertentu, memiliki alamat fisik dan nomor telepon, serta telah menjalankan bisnis selama beberapa tahun, kecil kemungkinannya bisnis Anda merupakan penipuan dibandingkan bisnis yang hanya memiliki situs web dan mungkin kotak pos. Pastikan Anda menampilkan detail ini.

Apa yang terjadi jika pembeli tidak puas dengan pembeliannya? Dalam keadaan apa Anda akan memberikan pengembalian uang? Siapa yang membayar ongkos kirim? Pengecer pesanan lewat pos secara tradisional memiliki kebijakan pengembalian uang dan pengembalian barang yang lebih liberal daripada toko tradisional. Banyak yang menawarkan jaminan kepuasan tanpa syarat. Pertimbangkan biaya pengembalian barang dibandingkan dengan peningkatan penjualan yang akan tercipta dari kebijakan pengembalian barang yang liberal. Apa pun kebijakan Anda, pastikan kebijakan tersebut ditampilkan di situs Anda.

Haruskah pelanggan mempercayakan informasi kartu kredit mereka kepada Anda? Masalah kepercayaan terbesar bagi pembeli internet adalah rasa takut mengirimkan detail kartu kredit mereka melalui internet. Oleh karena itu, Anda perlu menangani kartu kredit dengan aman dan terlihat sadar akan keamanan. Paling tidak, ini berarti menggunakan SSL (*Secure Sockets Layer*) untuk mengirimkan detail dari browser pengguna ke server Web Anda

dan memastikan bahwa server Web Anda dikelola secara kompeten dan aman. Kita akan membahas ini secara lebih rinci nanti.

Kemudahan Penggunaan

Pengalaman komputer, bahasa, literasi umum, memori, dan penglihatan konsumen sangat beragam. Situs Anda harus semudah mungkin digunakan. Desain antarmuka pengguna memenuhi banyak buku dengan sendirinya, tetapi berikut ini beberapa panduannya:

Buat situs Anda sesederhana mungkin. Semakin banyak opsi, iklan, dan gangguan di setiap layar, semakin besar kemungkinan pengguna akan bingung. Buat teks tetap jelas. Gunakan font yang jelas dan tidak rumit. Jangan buat teks terlalu kecil dan ingatlah bahwa ukurannya akan berbeda pada berbagai jenis mesin.

Buat proses pemesanan Anda sesederhana mungkin. Intuisi dan bukti yang tersedia mendukung gagasan bahwa semakin banyak klik mouse yang harus dilakukan pengguna untuk memesan, semakin kecil kemungkinan mereka untuk menyelesaikan proses tersebut. Batasi jumlah langkah seminimal mungkin, tetapi perlu diingat bahwa Amazon.com memiliki paten AS pada proses yang hanya menggunakan satu klik, yang disebut 1-Klik. Paten ini ditentang keras oleh banyak pemilik situs web.

Cobalah untuk tidak membiarkan pengguna tersesat. Berikan penanda dan petunjuk navigasi untuk memberi tahu pengguna di mana mereka berada. Misalnya, jika pengguna berada di dalam subbagian situs, sorot navigasi untuk subbagian tersebut.

Jika Anda menggunakan metafora keranjang belanja yang menyediakan wadah virtual bagi pelanggan untuk mengumpulkan pembelian sebelum menyelesaikan penjualan, pastikan tautan ke keranjang tersebut selalu terlihat di layar.

Kompatibilitas

Pastikan untuk menguji situs Anda di sejumlah browser dan sistem operasi. Jika situs tidak berfungsi untuk browser atau sistem operasi yang populer, Anda akan terlihat tidak profesional dan kehilangan sebagian pasar potensial Anda.

Jika situs Anda sudah beroperasi, log server Web Anda dapat memberi tahu Anda browser apa yang digunakan pengunjung Anda. Sebagai aturan praktis, jika Anda menguji situs Anda di dua versi terakhir Microsoft Internet Explorer dan Netscape Navigator pada PC yang menjalankan Microsoft Windows, dua versi terakhir Netscape Navigator pada Apple Mac, versi Netscape Navigator saat ini di Linux, dan browser khusus teks seperti Lynx, Anda akan terlihat oleh sebagian besar pengguna. Cobalah untuk menghindari fitur dan fasilitas yang benar-benar baru, kecuali Anda bersedia menulis dan mengelola beberapa versi situs.

Menyediakan Layanan dan Barang Digital

Banyak produk atau layanan dapat dijual melalui Web dan dikirimkan ke pelanggan melalui kurir. Beberapa layanan dapat segera dikirimkan secara daring. Jika layanan atau barang dapat dikirimkan ke modem, layanan atau barang tersebut dapat dipesan, dibayar, dan dikirimkan secara instan, tanpa interaksi manusia.

Layanan yang paling jelas yang disediakan dengan cara ini adalah informasi. Terkadang informasi tersebut sepenuhnya gratis atau didukung oleh iklan. Beberapa informasi disediakan melalui langganan atau dibayar secara individual.

Barang digital meliputi buku elektronik dan musik dalam format elektronik seperti MP3. Gambar perpustakaan stok dapat didigitalkan dan diunduh. Perangkat lunak komputer tidak selalu harus ada di CD, di dalam bungkus plastik. Perangkat lunak tersebut dapat diunduh secara langsung.

Layanan yang dapat dijual dengan cara ini meliputi akses Internet atau hosting Web, dan beberapa layanan profesional yang dapat digantikan oleh sistem pakar. Jika Anda akan mengirimkan barang yang dipesan dari situs web Anda secara fisik, Anda memiliki kelebihan dan kekurangan dibandingkan barang dan layanan digital.

Mengirim barang fisik memerlukan biaya. Unduhan digital hampir gratis. Ini berarti bahwa jika Anda memiliki sesuatu yang dapat digandakan dan dijual secara digital, biaya yang Anda keluarkan sangat mirip, baik jika Anda menjual satu barang atau seribu barang. Tentu saja, ada batasannya—jika Anda memiliki tingkat penjualan dan lalu lintas yang memadai, Anda perlu berinvestasi pada lebih banyak perangkat keras atau bandwidth.

Produk atau layanan digital dapat dengan mudah dijual sebagai pembelian impulsif. Jika seseorang memesan barang fisik, barang tersebut akan sampai kepadanya dalam waktu satu hari atau lebih. Unduhan biasanya diukur dalam hitungan detik atau menit. Kedekatan dapat menjadi beban bagi pedagang. Jika Anda mengirimkan pembelian secara digital, Anda harus melakukannya segera. Anda tidak dapat mengawasi proses secara manual, atau menyebarkan puncak aktivitas sepanjang hari.

Oleh karena itu, sistem pengiriman langsung lebih rentan terhadap penipuan dan lebih membebani sumber daya komputer. Barang dan jasa digital ideal untuk e-commerce, tetapi jelas hanya sejumlah kecil barang dan jasa yang dapat dikirimkan dengan cara ini.

Menambah Nilai pada Barang atau Jasa

Beberapa area situs web komersial yang sukses sebenarnya tidak menjual barang atau jasa apa pun. Layanan seperti layanan pelacakan perusahaan kurir (UPS di www.ups.com atau Fedex di www.fedex.com) umumnya tidak dirancang untuk menghasilkan laba secara langsung. Layanan ini menambah nilai pada layanan yang sudah ada yang ditawarkan oleh organisasi. Memungkinkan pelanggan melacak paket atau saldo bank mereka dapat memberi perusahaan keunggulan kompetitif.

Forum dukungan juga termasuk dalam kategori ini. Ada alasan komersial yang kuat untuk memberi pelanggan area diskusi guna berbagi kiat pemecahan masalah tentang produk perusahaan Anda.

Pelanggan mungkin dapat memecahkan masalah mereka dengan melihat solusi yang diberikan kepada orang lain, pelanggan internasional dapat memperoleh dukungan tanpa membayar panggilan telepon jarak jauh, dan pelanggan mungkin dapat menjawab pertanyaan satu sama lain di luar jam kantor Anda. Memberikan dukungan dengan cara ini dapat meningkatkan kepuasan pelanggan Anda dengan biaya rendah.

Memotong Biaya

Salah satu penggunaan Internet yang populer adalah untuk memotong biaya. Penghematan dapat diperoleh dari pendistribusian informasi secara daring, memfasilitasi komunikasi, penggantian layanan, atau pemusatan operasi.

Jika saat ini Anda menyediakan informasi kepada banyak orang, Anda mungkin dapat melakukan hal yang sama secara lebih ekonomis melalui situs Web. Baik Anda menyediakan daftar harga, katalog, prosedur terdokumentasi, spesifikasi, atau yang lainnya, akan lebih murah untuk menyediakan informasi yang sama di Web daripada mencetak dan mengirimkan salinan kertas. Hal ini khususnya berlaku untuk informasi yang berubah secara berkala. Internet dapat menghemat uang Anda dengan memfasilitasi komunikasi. Apakah ini berarti tender dapat didistribusikan secara luas dan ditanggapi dengan cepat, atau apakah itu berarti pelanggan dapat berkomunikasi langsung dengan pedagang grosir atau produsen, sehingga menghilangkan perantara, hasilnya tetap sama. Harga dapat turun, atau laba dapat naik.

Mengganti layanan yang memerlukan biaya untuk dijalankan dengan versi elektronik dapat memangkas biaya. Contoh yang berani adalah Egghead.com. Mereka memilih untuk menutup jaringan toko komputer mereka, dan berkonsentrasi pada aktivitas e-commerce mereka. Meskipun membangun situs e-commerce yang signifikan jelas membutuhkan biaya, jaringan lebih dari 70 toko ritel memiliki biaya berkelanjutan yang jauh lebih tinggi. Mengganti layanan yang sudah ada mengandung risiko. Paling tidak, Anda akan kehilangan pelanggan yang tidak menggunakan Internet.

Sentralisasi dapat memangkas biaya. Jika Anda memiliki banyak lokasi fisik, Anda perlu membayar banyak biaya sewa dan overhead, staf di semua lokasi tersebut, dan biaya pemeliharaan inventaris di setiap lokasi. Bisnis internet dapat berada di satu lokasi, tetapi dapat diakses di seluruh dunia.

6.2 RISIKO DAN ANCAMAN

Setiap bisnis menghadapi risiko, pesaing, pencurian, preferensi publik yang berubah-ubah, dan bencana alam, di antara risiko lainnya. Daftarnya tidak ada habisnya. Namun, banyak risiko yang dihadapi perusahaan e-commerce yang risikonya lebih kecil, atau tidak relevan, dengan usaha lain. Risiko ini meliputi

- Cracker
- Gagal menarik cukup banyak pelanggan
- Kegagalan perangkat keras komputer
- Kegagalan daya, komunikasi, atau jaringan
- Ketergantungan pada layanan pengiriman
- Persaingan yang ketat
- Kesalahan perangkat lunak
- Kebijakan dan pajak pemerintah yang terus berkembang
- Batasan kapasitas sistem

Cracker

Ancaman yang paling banyak dipublikasikan terhadap e-commerce berasal dari pengguna komputer jahat yang dikenal sebagai cracker. Semua bisnis berisiko menjadi target penjahat, tetapi bisnis e-commerce yang terkenal pasti akan menarik perhatian para cracker dengan berbagai niat dan kemampuan.

Cracker mungkin menyerang untuk mendapatkan tantangan, ketenaran, menyabotase situs Anda, mencuri uang, atau mendapatkan barang atau layanan gratis. Mengamankan situs Anda melibatkan kombinasi dari:

- Menyimpan cadangan informasi penting
- Memiliki kebijakan perekrutan yang menarik staf yang jujur dan membuat mereka tetap loyal—serangan paling berbahaya dapat datang dari dalam
- Mengambil tindakan pencegahan berbasis perangkat lunak, seperti memilih perangkat lunak yang aman dan menjaganya tetap mutakhir
- Melatih staf untuk mengidentifikasi target dan kelemahan
- Melakukan audit dan pencatatan untuk mendeteksi pembobolan atau percobaan pembobolan

Serangan yang paling berhasil pada sistem komputer memanfaatkan kelemahan yang sudah diketahui seperti kata sandi yang mudah ditebak, kesalahan konfigurasi yang umum, dan versi perangkat lunak yang lama. Beberapa tindakan pencegahan yang masuk akal dapat menangkal serangan yang tidak dilakukan oleh ahli dan memastikan Anda memiliki cadangan jika hal terburuk terjadi.

Gagal Menarik Bisnis yang Cukup

Meskipun serangan oleh cracker sangat dikhawatirkan, sebagian besar kegagalan e-commerce berhubungan dengan faktor ekonomi tradisional. Membangun dan memasarkan situs e-commerce besar membutuhkan banyak uang.

Perusahaan bersedia kehilangan uang dalam jangka pendek, berdasarkan asumsi bahwa setelah merek mapan di pasar, jumlah pelanggan dan pendapatan akan meningkat. Pada saat penulisan ini, Amazon.com, yang bisa dibilang pengecer paling terkenal di Web, telah diperdagangkan dengan kerugian selama lima tahun berturut-turut, merugi Rp. 990 Miliar (AS) pada kuartal pertama tahun 2000. Rangkaian kegagalan yang terkenal termasuk boo.com Eropa, yang kehabisan uang dan berpindah tangan setelah menghabiskan Rp. 1.2 Triliun dalam enam bulan. Bukan berarti Boo tidak melakukan penjualan; hanya saja mereka menghabiskan lebih banyak uang daripada yang mereka hasilkan.

Kegagalan Perangkat Keras Komputer

Hampir tidak perlu dikatakan lagi bahwa jika bisnis Anda bergantung pada situs Web, kegagalan bagian penting dari salah satu komputer Anda akan berdampak. Situs Web yang sibuk atau penting membenarkan adanya beberapa sistem redundan sehingga kegagalan salah satunya tidak memengaruhi pengoperasian seluruh sistem. Seperti halnya semua ancaman, Anda perlu menentukan apakah kemungkinan kehilangan situs Web Anda selama sehari saat menunggu suku cadang atau perbaikan membenarkan pengeluaran peralatan redundan.

Kegagalan Daya, Komunikasi, Jaringan, atau Pengiriman

Jika Anda bergantung pada Internet, Anda bergantung pada jaringan penyedia layanan yang kompleks. Jika koneksi Anda ke seluruh dunia gagal, Anda tidak dapat melakukan apa pun selain menunggu pemasok Anda memulihkan layanan. Hal yang sama berlaku untuk gangguan layanan listrik, dan pemogokan atau penghentian lainnya oleh perusahaan pengiriman Anda.

Tergantung pada anggaran Anda, Anda mungkin memilih untuk mempertahankan beberapa layanan dari penyedia yang berbeda. Ini akan lebih mahal, tetapi berarti bahwa, jika salah satu penyedia Anda gagal, Anda masih akan memiliki yang lain. Pemadaman listrik singkat dapat diatasi dengan berinvestasi pada catu daya tak terputus.

Persaingan Luas

Jika Anda membuka gerai ritel di sudut jalan, Anda mungkin dapat membuat survei yang cukup akurat tentang lanskap persaingan. Pesaing Anda terutama akan menjadi bisnis yang menjual barang serupa di daerah sekitar. Pesaing baru akan sesekali membuka usaha. Dengan e-commerce, medannya kurang pasti.

Tergantung pada biaya pengiriman, pesaing Anda dapat berada di mana saja di dunia, dan tunduk pada fluktuasi mata uang dan biaya tenaga kerja yang berbeda. Internet sangat kompetitif dan berkembang pesat. Jika Anda bersaing dalam kategori populer, pesaing baru dapat muncul setiap hari.

Tidak banyak yang dapat Anda lakukan untuk menghilangkan risiko persaingan, tetapi, dengan mengikuti perkembangan, Anda dapat memastikan bahwa usaha Anda tetap kompetitif.

Kesalahan Perangkat Lunak

Bila bisnis Anda bergantung pada perangkat lunak, Anda rentan terhadap kesalahan dalam perangkat lunak tersebut. Anda dapat mengurangi kemungkinan kesalahan kritis dengan memilih perangkat lunak yang andal, menyediakan waktu yang cukup untuk pengujian setelah mengubah bagian sistem Anda, memiliki proses pengujian formal, dan tidak mengizinkan perubahan dilakukan pada sistem aktif Anda tanpa pengujian di tempat lain terlebih dahulu.

Anda dapat mengurangi tingkat keparahan hasil dengan memiliki cadangan data terkini, menyimpan konfigurasi perangkat lunak yang berfungsi saat membuat perubahan, dan memantau operasi sistem untuk mendeteksi masalah dengan cepat.

Kebijakan Pemerintah dan Pajak yang Berkembang

Tergantung di mana Anda tinggal, undang-undang yang berkaitan dengan bisnis berbasis internet mungkin belum ada, sedang dalam proses, atau belum matang. Hal ini tidak akan bertahan lama. Beberapa model bisnis mungkin terancam, diatur, atau dihilangkan oleh undang-undang di masa mendatang. Pajak mungkin ditambahkan.

Anda tidak dapat menghindari masalah ini. Satu-satunya cara untuk mengatasinya adalah dengan terus mengikuti perkembangan terkini dan menjaga situs Anda sesuai dengan undang-undang. Anda mungkin ingin mempertimbangkan untuk bergabung dengan kelompok lobi yang sesuai saat masalah muncul.

Batasan Kapasitas Sistem

Satu hal yang perlu diingat saat merancang sistem Anda adalah pertumbuhan. Sistem Anda diharapkan akan semakin sibuk. Sistem harus dirancang sedemikian rupa sehingga dapat diskalakan untuk memenuhi permintaan. Untuk pertumbuhan yang terbatas, Anda dapat meningkatkan kapasitas hanya dengan membeli perangkat keras yang lebih cepat. Ada batasan seberapa cepat komputer yang dapat Anda beli. Apakah perangkat lunak Anda ditulis

sedemikian rupa sehingga setelah mencapai titik ini, Anda dapat memisahkan bagian-bagiannya untuk berbagi beban pada beberapa sistem? Dapatkah basis data Anda menangani beberapa permintaan bersamaan dari mesin yang berbeda?

Hanya sedikit sistem yang dapat mengatasi pertumbuhan besar dengan mudah, tetapi jika Anda merancangnya dengan mempertimbangkan skalabilitas, Anda akan dapat mengidentifikasi dan menghilangkan hambatan seiring dengan pertumbuhan basis pelanggan Anda.

6.3 MENENTUKAN STRATEGI

Beberapa orang percaya bahwa Internet berubah terlalu cepat sehingga tidak memungkinkan perencanaan yang efektif. Kami berpendapat bahwa perubahan inilah yang membuat perencanaan menjadi penting. Tanpa menetapkan tujuan dan menentukan strategi, Anda akan bereaksi terhadap perubahan yang terjadi, alih-alih mampu bertindak untuk mengantisipasi perubahan.

Setelah memeriksa beberapa tujuan umum untuk situs web komersial, dan beberapa ancaman utama, mudah-mudahan Anda memiliki beberapa strategi untuk Anda sendiri.

Strategi Anda perlu mengidentifikasi model bisnis. Model tersebut biasanya merupakan sesuatu yang telah terbukti berhasil di tempat lain, tetapi terkadang merupakan ide baru yang Anda yakini. Apakah Anda akan mengadaptasi model bisnis Anda yang ada ke Web, meniru pesaing yang ada, atau secara agresif menciptakan layanan perintis?

BAB 7
MASALAH KEAMANAN E-COMMERCE

Bab ini membahas peran keamanan dalam e-commerce. Kita akan membahas siapa saja yang mungkin tertarik dengan informasi Anda dan bagaimana mereka mungkin mencoba mendapatkannya, prinsip-prinsip yang terlibat dalam pembuatan kebijakan untuk menghindari masalah semacam ini, dan beberapa teknologi yang tersedia untuk menjaga keamanan situs Web termasuk enkripsi, autentikasi, dan pelacakan.

Pembahasan dalam Bab ini

- Seberapa penting informasi Anda?
- Ancaman keamanan
- Membuat kebijakan keamanan
- Menyeimbangkan kegunaan, kinerja, biaya, dan keamanan
- Prinsip autentikasi
- Menggunakan autentikasi
- Dasar-dasar enkripsi
- Enkripsi Kunci Pribadi
- Enkripsi Kunci Publik
- Tanda tangan digital
- Sertifikat digital
- Server Web yang aman
- Audit dan pencatatan
- Firewall
- Mencadangkan data
- Keamanan fisik

Seberapa Penting Informasi Anda? Saat mempertimbangkan keamanan, hal pertama yang perlu Anda evaluasi adalah pentingnya apa yang Anda lindungi. Anda perlu mempertimbangkan pentingnya hal tersebut bagi Anda dan bagi para peretas potensial.

Mungkin tergoda untuk percaya bahwa tingkat keamanan setinggi mungkin diperlukan untuk semua situs setiap saat, tetapi perlindungan itu ada harganya. Sebelum memutuskan seberapa besar upaya atau biaya yang diperlukan untuk keamanan Anda, Anda perlu memutuskan seberapa berharganya informasi Anda.

Nilai informasi yang disimpan di komputer pengguna hobi, bisnis, bank, dan organisasi militer jelas berbeda-beda. Sejauh mana penyerang mungkin akan berusaha untuk mendapatkan akses ke informasi tersebut juga berbeda-beda. Seberapa menarikkah isi mesin Anda bagi pengunjung yang berniat jahat?

Pengguna yang hobi mungkin memiliki waktu terbatas untuk mempelajari atau berupaya mengamankan sistem mereka. Mengingat bahwa informasi yang disimpan di komputer mereka kemungkinan besar tidak terlalu berharga bagi siapa pun selain pemiliknya,

serangan kemungkinan jarang terjadi dan memerlukan upaya yang terbatas. Namun, semua pengguna komputer jaringan harus mengambil tindakan pencegahan yang wajar. Bahkan komputer dengan data yang paling tidak menarik pun tetap memiliki daya tarik yang signifikan sebagai landasan peluncuran anonim untuk serangan pada sistem lain.

Komputer militer merupakan target yang jelas bagi individu dan pemerintah asing. Karena pemerintah yang menyerang mungkin memiliki sumber daya yang besar, akan lebih bijaksana untuk menginvestasikan personel dan sumber daya lainnya untuk memastikan bahwa semua tindakan pencegahan praktis diambil dalam domain ini. Jika Anda bertanggung jawab atas situs e-commerce, daya tariknya bagi para cracker mungkin berada di antara kedua ekstrem ini.

7.1 ANCAMAN KEAMANAN

Apa yang berisiko di situs Anda? Ancaman apa yang ada di luar sana?

Kami membahas beberapa ancaman terhadap bisnis e-commerce di Bab 8, “Menjalankan Situs E-commerce.” Banyak di antaranya terkait dengan keamanan.
Bergantung pada situs Web Anda, ancaman keamanan mungkin mencakup:

- Terungkapnya data rahasia
- Hilangnya atau rusaknya data
- Modifikasi data
- Penolakan layanan
- Kesalahan dalam perangkat lunak
- Penolakan
- Mari kita bahas masing-masing ancaman ini.

Terungkapnya Data Rahasia

Data yang disimpan di komputer Anda, atau yang sedang dikirim ke atau dari komputer Anda, mungkin bersifat rahasia. Mungkin berupa informasi yang hanya boleh dilihat oleh orang tertentu seperti daftar harga grosir. Mungkin berupa informasi rahasia yang diberikan oleh pelanggan, seperti kata sandinya, detail kontak, dan nomor kartu kredit.

Semoga Anda tidak menyimpan informasi di server Web yang tidak ingin dilihat oleh siapa pun. Server Web bukanlah tempat yang tepat untuk menyimpan informasi rahasia. Jika Anda menyimpan catatan penggajian atau rencana Anda untuk menguasai dunia di komputer, sebaiknya Anda menggunakan komputer selain server Web Anda. Server Web pada dasarnya adalah mesin yang dapat diakses publik, dan hanya boleh memuat informasi yang perlu disediakan untuk publik atau yang baru saja dikumpulkan dari publik.

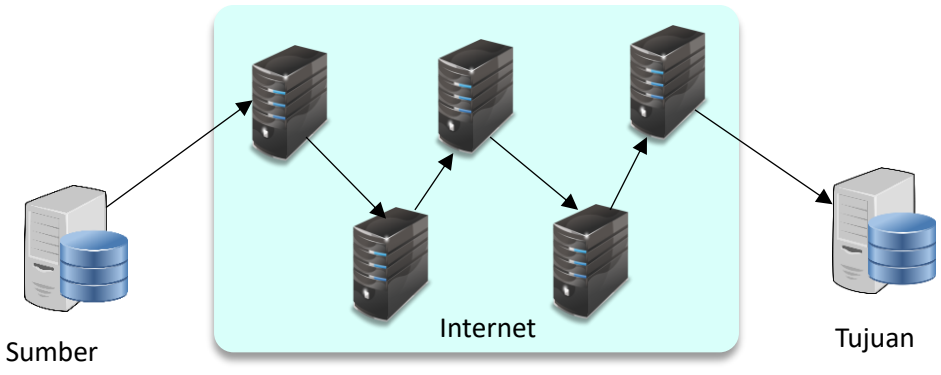
Untuk mengurangi risiko paparan, Anda perlu membatasi metode akses informasi dan membatasi orang yang dapat mengaksesnya. Ini melibatkan perancangan dengan mempertimbangkan keamanan, konfigurasi server dan perangkat lunak Anda dengan benar, pemrograman dengan hati-hati, pengujian secara menyeluruh, penghapusan layanan yang tidak diperlukan dari server Web, dan persyaratan autentikasi.

Rancang, konfigurasikan, buat kode, dan uji dengan hati-hati untuk mengurangi risiko serangan kriminal yang berhasil dan, yang sama pentingnya, untuk mengurangi kemungkinan kesalahan yang akan membuat informasi Anda terbuka terhadap paparan yang tidak disengaja. Hapus layanan yang tidak diperlukan dari server Web Anda untuk mengurangi jumlah titik lemah yang potensial. Setiap layanan yang Anda jalankan mungkin memiliki kerentanan. Masing-masing layanan perlu diperbarui untuk memastikan bahwa kerentanan yang diketahui tidak ada. Layanan yang tidak Anda gunakan mungkin lebih berbahaya. Jika Anda tidak pernah menggunakan perintah rcp, mengapa layanan tersebut diinstal? Jika Anda memberi tahu penginstal bahwa komputer Anda adalah host jaringan, distribusi Linux utama dan Windows NT menginstal sejumlah besar layanan yang tidak Anda perlukan dan harus dihapus.

Autentikasi berarti meminta orang untuk membuktikan identitas mereka. Ketika sistem mengetahui siapa yang membuat permintaan, sistem dapat memutuskan apakah orang tersebut diizinkan mengakses. Ada sejumlah metode autentikasi yang memungkinkan, tetapi hanya dua bentuk yang umum digunakan—kata sandi dan tanda tangan digital. Kita akan membahas keduanya lebih lanjut nanti.

CD Universe memberikan contoh yang baik tentang biaya baik dalam bentuk dolar maupun reputasi yang ditimbulkan oleh informasi rahasia yang terekspos. Pada akhir tahun 1999, seorang cracker yang menyebut dirinya Maxus dilaporkan menghubungi CD Universe, mengklaim telah mencuri 300.000 nomor kartu kredit dari situs mereka. Ia meminta tebusan sebesar Rp. 1.000.000.000 (AS) dari situs tersebut untuk menghancurkan nomor-nomor tersebut. Mereka menolak, dan mendapati diri mereka dalam liputan memalukan di halaman depan surat kabar utama saat Maxus membagikan angka-angka untuk disalahgunakan orang lain.

Data juga berisiko terekspos saat melintasi jaringan. Meskipun jaringan TCP/IP memiliki banyak fitur hebat yang menjadikannya standar de facto untuk menghubungkan berbagai jaringan bersama-sama sebagai Internet, keamanan bukanlah salah satunya. TCP/IP bekerja dengan memotong data Anda menjadi paket-paket, dan kemudian meneruskan paket-paket tersebut dari satu mesin ke mesin lainnya hingga mencapai tujuannya. Ini berarti bahwa data Anda melewati banyak mesin dalam perjalanan, seperti yang diilustrasikan dalam Gambar 7.1. Salah satu dari mesin-mesin tersebut dapat melihat data Anda saat data tersebut lewat.



Gambar 7.1 Mentransmisikan informasi melalui Internet akan mengirimkan informasi Anda melalui sejumlah host yang berpotensi tidak dapat dipercaya.

Untuk melihat jalur yang diambil data dari Anda ke mesin tertentu, Anda dapat menggunakan perintah traceroute (pada mesin UNIX). Perintah ini akan memberi Anda alamat mesin yang dilalui data Anda untuk mencapai host tersebut. Untuk host di negara Anda sendiri, data kemungkinan akan melewati 10 mesin yang berbeda. Untuk mesin internasional, mungkin ada lebih dari 20 perantara. Jika organisasi Anda memiliki jaringan yang besar dan kompleks, data Anda mungkin melewati lima mesin bahkan sebelum meninggalkan gedung.

Untuk melindungi informasi rahasia, Anda dapat mengenkripsinya sebelum dikirim melalui jaringan, dan mendekripsinya di ujung lainnya. Server web sering menggunakan Secure Socket Layer (SSL), yang dikembangkan oleh Netscape, untuk mencapai hal ini saat data berpindah antara server Web dan browser. Ini adalah cara yang cukup murah dan mudah untuk mengamankan transmisi, tetapi karena server Anda perlu mengenkripsi dan mendekripsi data, bukan sekadar mengirim dan menerimanya, jumlah pengunjung per detik yang dapat dilayani mesin akan turun drastis.

7.2 KEHILANGAN ATAU PENGHANCURAN DATA

Kehilangan data bisa lebih merugikan Anda daripada jika data tersebut terungkap. Jika Anda telah menghabiskan waktu berbulan-bulan untuk membangun situs Anda, serta mengumpulkan data pengguna dan pesanan, berapa biaya yang akan Anda keluarkan, dalam hal waktu, reputasi, dan uang, untuk kehilangan semua informasi tersebut? Jika Anda tidak memiliki cadangan data apa pun, Anda perlu menulis ulang situs web dengan tergesa-gesa dan memulai dari awal.

Ada kemungkinan peretas akan membobol sistem Anda dan memformat hard drive Anda. Sangat mungkin seorang programmer atau administrator yang ceroboh akan menghapus sesuatu secara tidak sengaja, dan hampir dapat dipastikan bahwa Anda terkadang akan kehilangan hard disk drive. Hard disk drive berputar ribuan kali per menit, dan, terkadang, rusak. Hukum Murphy akan memberi tahu Anda bahwa yang gagal akan menjadi yang paling penting, jauh setelah Anda terakhir kali membuat cadangan.

Anda dapat mengambil berbagai tindakan untuk mengurangi kemungkinan kehilangan data. Amankan server Anda dari peretas. Batasi jumlah staf yang dapat mengakses komputer Anda. Pekerjakan hanya orang yang kompeten dan cermat. Beli drive berkualitas baik. Gunakan RAID sehingga beberapa drive dapat berfungsi seperti satu drive yang lebih cepat dan lebih andal.

Apa pun penyebabnya, hanya ada satu perlindungan nyata terhadap kehilangan data—pencadangan. Mencadangkan data bukanlah ilmu roket. Sebaliknya, itu membosankan, menjemukan, dan mudah-mudahan tidak berguna, tetapi sangat penting. Pastikan data Anda dicadangkan secara teratur, dan pastikan Anda telah menguji prosedur pencadangan untuk memastikan bahwa Anda dapat memulihkannya. Pastikan cadangan Anda disimpan jauh dari komputer Anda. Meskipun kecil kemungkinan tempat Anda akan terbakar atau mengalami nasib buruk lainnya, menyimpan cadangan di luar lokasi adalah polis asuransi yang cukup murah.

Modifikasi Data

Meskipun hilangnya data dapat merusak, modifikasi dapat lebih buruk. Bagaimana jika seseorang memperoleh akses ke sistem Anda dan mengubah file? Meskipun penghapusan besar-besaran mungkin akan diketahui, dan dapat diperbaiki dari cadangan Anda, berapa lama waktu yang Anda perlukan untuk melihat modifikasinya?

Modifikasi pada file dapat mencakup perubahan pada file data atau file yang dapat dieksekusi. Motivasi seorang cracker untuk mengubah file data mungkin untuk membuat grafiti di situs Anda atau untuk mendapatkan keuntungan yang curang.

Mengganti file yang dapat dieksekusi dengan versi yang disabotase dapat memberikan cracker yang telah memperoleh akses sekali pintu belakang rahasia untuk kunjungan berikutnya.

Anda dapat melindungi data dari modifikasi saat data tersebut bergerak melalui jaringan dengan menghitung tanda tangan. Ini tidak menghentikan seseorang untuk memodifikasi data, tetapi jika penerima memeriksa bahwa tanda tangan masih cocok saat file tiba, ia akan tahu apakah file tersebut telah dimodifikasi. Jika data dienkripsi untuk melindunginya dari tampilan yang tidak sah, ini juga akan membuatnya sangat sulit untuk dimodifikasi selama perjalanan tanpa terdeteksi.

Melindungi file yang disimpan di server Anda dari modifikasi mengharuskan Anda menggunakan fasilitas izin file yang disediakan sistem operasi Anda dan melindungi sistem dari akses yang tidak sah. Dengan menggunakan izin berkas, pengguna dapat diberi wewenang untuk menggunakan sistem, tetapi tidak diberi kebebasan untuk mengubah berkas sistem dan berkas pengguna lain. Kurangnya sistem izin yang tepat merupakan salah satu alasan mengapa Windows 95 dan 98 tidak cocok sebagai sistem operasi server.

Mendeteksi modifikasi bisa jadi sulit. Jika pada suatu saat Anda menyadari bahwa keamanan sistem Anda telah dilanggar, bagaimana Anda akan tahu apakah berkas penting telah diubah? Beberapa berkas, seperti berkas data yang menyimpan basis data Anda, dimaksudkan untuk berubah seiring waktu. Banyak berkas lainnya dimaksudkan untuk tetap sama sejak Anda menginstalnya, kecuali Anda sengaja memutakhirkannya. Modifikasi program dan data dapat dilakukan secara diam-diam, tetapi meskipun program dapat diinstal ulang jika Anda mencurigai adanya modifikasi, Anda tidak dapat mengetahui versi data mana yang "bersih".

Perangkat lunak penilaian integritas berkas, seperti Tripwire, mencatat informasi tentang berkas penting dalam status aman yang diketahui, mungkin segera setelah penginstalan, dan dapat digunakan nanti untuk memverifikasi bahwa berkas tidak berubah. Anda dapat mengunduh versi komersial atau gratis bersyarat dari http://www.tripwire.com Penolakan Layanan Salah satu ancaman yang paling sulit untuk diwaspadai adalah penolakan layanan.

Penolakan Layanan (DoS) terjadi ketika tindakan seseorang membuat pengguna sulit atau tidak dapat mengakses layanan, atau menunda akses mereka ke layanan yang sangat penting. Di awal tahun 2000, terjadi serentetan serangan Penolakan Layanan Terdistribusi (DDoS) yang terkenal terhadap situs web terkenal. Sasarannya termasuk Yahoo!, eBay,

Amazon, E-Trade, dan Buy.com. Situs seperti ini terbiasa dengan tingkat lalu lintas yang sebagian besar dari kita hanya dapat impikan, tetapi masih rentan untuk ditutup selama berjam-jam oleh serangan DoS. Meskipun para peretas pada umumnya tidak akan mendapatkan banyak keuntungan dari menutup situs web, pemiliknya mungkin akan kehilangan uang, waktu, dan reputasi.

Salah satu alasan mengapa serangan ini sangat sulit dicegah adalah karena ada banyak sekali cara untuk melakukannya. Metode yang dapat dilakukan termasuk memasang program pada mesin target yang menggunakan sebagian besar waktu prosesor sistem, melakukan reverse spamming, atau menggunakan salah satu alat otomatis. Reverse spam melibatkan seseorang yang mengirimkan spam palsu dengan target yang tercantum sebagai pengirim. Dengan cara ini, target akan menghadapi ribuan balasan marah.

Alat otomatis tersedia untuk meluncurkan serangan DoS terdistribusi pada target. Tanpa memerlukan banyak pengetahuan, seseorang dapat memindai sejumlah besar mesin untuk mengetahui kerentanan yang diketahui, menyusupi mesin, dan memasang alat tersebut. Karena prosesnya otomatis, penyerang dapat memasang alat tersebut pada satu host dalam waktu kurang dari lima detik. Ketika cukup banyak mesin yang telah direbut, semuanya diperintahkan untuk membanjiri target dengan lalu lintas jaringan.

Melindungi diri dari serangan DoS secara umum sulit. Dengan sedikit riset, Anda dapat menemukan port default yang digunakan oleh alat DDoS umum dan menutupnya. Router Anda mungkin menyediakan mekanisme seperti membatasi persentase lalu lintas yang menggunakan protokol tertentu seperti ICMP. Mendeteksi host di jaringan Anda yang digunakan untuk menyerang orang lain lebih mudah daripada melindungi mesin Anda dari serangan. Jika setiap administrator jaringan dapat diandalkan untuk memantau jaringannya sendiri dengan waspada, DDoS tidak akan menjadi masalah besar.

Karena ada begitu banyak metode serangan yang memungkinkan, satu-satunya pertahanan yang benar-benar efektif adalah memantau perilaku lalu lintas normal dan memiliki sekelompok ahli yang tersedia untuk mengambil tindakan pencegahan ketika hal-hal yang tidak normal terjadi.

Kesalahan dalam Perangkat Lunak

Ada kemungkinan perangkat lunak yang Anda beli, peroleh, atau tulis memiliki kesalahan serius di dalamnya. Mengingat waktu pengembangan yang singkat yang biasanya diberikan untuk proyek Web, sangat mungkin perangkat lunak ini memiliki beberapa kesalahan. Setiap bisnis yang sangat bergantung pada proses komputerisasi rentan terhadap perangkat lunak yang bermasalah.

Kesalahan dalam perangkat lunak dapat menyebabkan segala macam perilaku yang tidak terduga termasuk tidak tersedianya layanan, pelanggaran keamanan, kerugian finansial, dan layanan yang buruk kepada pelanggan.

Penyebab umum kesalahan yang dapat Anda cari termasuk spesifikasi yang buruk, asumsi yang salah yang dibuat oleh pengembang, dan pengujian yang tidak memadai.

Spesifikasi yang Buruk

Semakin jarang atau ambigu dokumentasi desain Anda, semakin besar kemungkinan Anda akan berakhir dengan kesalahan dalam produk akhir. Meskipun mungkin tampak berlebihan bagi Anda untuk menentukan bahwa ketika kartu kredit pelanggan ditolak, pesanan tidak boleh dikirim ke pelanggan, setidaknya satu situs beranggaran besar memiliki bug ini. Semakin sedikit pengalaman yang dimiliki pengembang Anda dengan jenis sistem yang mereka kerjakan, semakin tepat spesifikasi yang Anda butuhkan.

Asumsi yang Dibuat oleh Pengembang

Perancang dan pemrogram suatu sistem perlu membuat banyak asumsi. Mudah-mudahan, mereka akan mendokumentasikan asumsi mereka dan biasanya benar. Namun, terkadang, orang membuat asumsi yang buruk. Ini mungkin termasuk asumsi bahwa data masukan akan valid, tidak akan menyertakan karakter yang tidak biasa, atau akan berukuran kurang dari tertentu. Ini juga dapat mencakup asumsi tentang waktu seperti kemungkinan dua tindakan yang saling bertentangan terjadi pada saat yang sama atau bahwa tugas pemrosesan yang rumit akan selalu membutuhkan waktu lebih lama daripada tugas yang sederhana.

Asumsi seperti ini dapat lolos karena biasanya benar. Seorang cracker dapat memanfaatkan buffer overrun karena seorang programmer mengasumsikan panjang maksimum untuk data masukan, atau pengguna yang sah dapat memperoleh pesan kesalahan yang membingungkan dan pergi karena tidak terpikir oleh pengembang Anda bahwa nama seseorang mungkin mengandung apostrof. Kesalahan semacam ini dapat ditemukan dan diperbaiki dengan kombinasi pengujian yang baik dan peninjauan kode yang terperinci.

Secara historis, kelemahan sistem operasi atau tingkat aplikasi yang dieksploitasi oleh cracker biasanya terkait dengan buffer overflow atau kondisi race.

Pengujian yang Buruk

Sangat jarang memungkinkan untuk menguji semua kemungkinan kondisi input, pada semua kemungkinan jenis perangkat keras, menjalankan semua kemungkinan sistem operasi dengan semua kemungkinan pengaturan pengguna. Hal ini bahkan lebih benar daripada biasanya dengan sistem berbasis Web.

Yang dibutuhkan adalah rencana pengujian yang dirancang dengan baik yang menguji semua fungsi perangkat lunak Anda pada sampel representatif dari jenis mesin umum. Serangkaian pengujian yang direncanakan dengan baik harus bertujuan untuk menguji setiap baris kode dalam proyek Anda setidaknya sekali. Idealnya, rangkaian pengujian ini harus diotomatisasi sehingga dapat dijalankan pada mesin pengujian yang Anda pilih dengan sedikit usaha.

Masalah terbesar dengan pengujian adalah bahwa pengujian itu tidak menarik dan berulang-ulang. Meskipun beberapa orang senang merusak sesuatu, hanya sedikit orang yang senang merusak hal yang sama berulang-ulang. Penting bagi orang lain selain pengembang asli untuk terlibat dalam pengujian. Salah satu tujuan utama pengujian adalah untuk mengungkap asumsi yang salah yang dibuat oleh pengembang. Orang yang baru lebih cenderung memiliki asumsi yang berbeda. Selain itu, para profesional jarang ingin menemukan kekurangan dalam pekerjaan mereka sendiri.

Penolakan

Risiko terakhir yang akan kita pertimbangkan adalah penolakan. Penolakan terjadi ketika pihak yang terlibat dalam transaksi menyangkal telah mengambil bagian. Contoh e-dagang antara lain adalah seseorang memesan barang lewat situs Web, lalu menyangkal telah mengotorisasi penagihan pada kartu kreditnya; atau seseorang menyetujui sesuatu lewat email, lalu mengklaim bahwa orang lain memalsukan email tersebut.

Idealnya, transaksi keuangan harus memberikan ketenangan pikiran bagi kedua belah pihak karena tidak dapat disangkal. Tidak ada pihak yang dapat menyangkal peran mereka dalam transaksi, atau, lebih tepatnya, kedua belah pihak dapat membuktikan tindakan pihak lain secara meyakinkan kepada pihak ketiga, seperti pengadilan. Dalam praktiknya, hal ini jarang terjadi.

Autentikasi memberikan kepastian tentang dengan siapa Anda berurusan. Jika dikeluarkan oleh organisasi tepercaya, sertifikat autentikasi digital dapat memberikan keyakinan yang lebih besar. Pesan yang dikirim oleh masing-masing pihak juga harus antirusak. Tidak banyak gunanya untuk dapat menunjukkan bahwa Corp Pty Ltd mengirimi Anda pesan jika Anda juga tidak dapat menunjukkan bahwa apa yang Anda terima benar-benar apa yang mereka kirim. Seperti yang disebutkan sebelumnya, menandatangani atau mengenkripsi pesan membuatnya sulit untuk diubah secara diam-diam.

Untuk transaksi antara pihak-pihak yang memiliki hubungan yang sedang berlangsung, sertifikat digital bersama dengan komunikasi yang dienkripsi atau ditandatangani merupakan cara yang efektif untuk membatasi penyangkalan. Untuk transaksi satu kali, seperti kontak awal antara situs web e-commerce dan orang asing yang membawa kartu kredit, hal itu tidak begitu praktis.

Perusahaan e-commerce harus bersedia menyerahkan bukti identitasnya dan beberapa ratus dolar kepada otoritas sertifikasi seperti VeriSign (http://www.verisign.com/) atau Thawte (http://www.thawte.com/) untuk meyakinkan pengunjung tentang kredibilitas perusahaan. Apakah perusahaan yang sama akan bersedia menolak setiap pelanggan yang tidak bersedia melakukan hal yang sama untuk membuktikan identitasnya? Untuk transaksi kecil, pedagang umumnya bersedia menerima risiko penipuan atau penolakan tertentu daripada menolak bisnis.

Sebuah aliansi antara VISA, sejumlah organisasi keuangan, dan perusahaan perangkat lunak, telah mempromosikan sebuah standar yang disebut Transaksi Elektronik Aman sejak tahun 1997. Salah satu komponen sistem SET adalah pemegang kartu dapat memperoleh sertifikat digital dari penerbit kartu mereka. Jika SET berhasil, hal itu dapat mengurangi risiko penolakan dan penipuan kartu kredit lainnya dalam transaksi Internet.

Sayangnya, meskipun spesifikasi tersebut telah ada selama bertahun-tahun, tampaknya hanya ada sedikit dorongan dari bank untuk menerbitkan sertifikat yang sesuai dengan SET kepada pemegang kartu mereka. Tidak ada pengecer yang tampaknya bersedia menolak semua pelanggan tanpa perangkat lunak SET, dan hanya ada sedikit antusiasme dari konsumen untuk mengadopsi perangkat lunak tersebut. Sangat sedikit alasan bagi konsumen untuk mengantre di bank lokal mereka dan menghabiskan waktu memasang perangkat lunak

dompet digital di mesin mereka kecuali pengecer akan menolak pelanggan mereka tanpa perangkat lunak tersebut.

7.3 MENYEIMBANGKAN KEGUNAAN, PERFORMA, BIAYA, DAN KEAMANAN

Pada dasarnya, Web berisiko. Web dirancang untuk memungkinkan banyak pengguna anonim meminta layanan dari mesin Anda. Sebagian besar permintaan tersebut akan menjadi permintaan yang sah untuk halaman Web, tetapi menghubungkan komputer Anda ke Internet akan memungkinkan orang untuk mencoba jenis koneksi lainnya.

Meskipun mungkin tergoda untuk berasumsi bahwa tingkat keamanan setinggi mungkin sudah tepat, hal ini jarang terjadi. Jika Anda ingin benar-benar aman, Anda akan mematikan semua komputer Anda, memutuskan sambungan dari semua jaringan, di brankas terkunci. Agar komputer Anda tersedia dan dapat digunakan, diperlukan sedikit pelonggaran keamanan.

Ada trade-off yang harus dibuat antara keamanan, kegunaan, biaya, dan kinerja. Membuat layanan lebih aman dapat mengurangi kegunaan dengan, misalnya, membatasi apa yang dapat dilakukan orang atau mengharuskan mereka untuk mengidentifikasi diri mereka sendiri. Meningkatkan keamanan juga dapat mengurangi tingkat kinerja komputer Anda. Menjalankan perangkat lunak untuk membuat sistem Anda lebih aman—seperti enkripsi, sistem deteksi intrusi, pemindai virus, dan pencatatan ekstensif—menggunakan sumber daya. Dibutuhkan lebih banyak daya pemrosesan untuk menyediakan sesi terenkripsi, seperti koneksi SSL ke situs Web, daripada menyediakan sesi normal. Kerugian kinerja ini dapat diatasi dengan menghabiskan lebih banyak uang untuk mesin yang lebih cepat atau perangkat keras yang dirancang khusus untuk enkripsi.

Anda dapat melihat kinerja, kegunaan, biaya, dan keamanan sebagai tujuan yang saling bersaing. Anda perlu memeriksa trade-off yang diperlukan dan membuat keputusan yang masuk akal untuk mencapai kompromi. Bergantung pada nilai informasi Anda, anggaran Anda, berapa banyak pengunjung yang Anda harapkan untuk dilayani, dan hambatan apa yang menurut Anda akan dihadapi oleh pengguna yang sah, Anda dapat mencapai posisi kompromi.

Membuat Kebijakan Keamanan

Kebijakan keamanan adalah dokumen yang menjelaskan

- Filosofi umum terhadap keamanan di organisasi Anda
- Apa yang harus dilindungi—perangkat lunak, perangkat keras, data
- Siapa yang bertanggung jawab untuk melindungi item-item ini
- Standar keamanan dan metrik, yang mengukur seberapa baik standar tersebut dipenuhi

Pedoman yang baik untuk menulis kebijakan keamanan Anda adalah seperti menulis serangkaian persyaratan fungsional untuk perangkat lunak. Kebijakan tersebut tidak boleh membahas implementasi atau solusi tertentu, tetapi tentang tujuan dan persyaratan keamanan di lingkungan Anda. Kebijakan tersebut tidak perlu diperbarui terlalu sering.

Anda harus menyimpan dokumen terpisah yang menetapkan pedoman tentang bagaimana persyaratan kebijakan keamanan dipenuhi di lingkungan tertentu. Anda dapat

memiliki pedoman yang berbeda untuk bagian yang berbeda dari organisasi Anda. Ini lebih seperti dokumen desain atau manual prosedur yang mendokumentasikan apa yang sebenarnya dilakukan untuk memastikan tingkat keamanan yang Anda perlukan.

Prinsip Autentikasi

Autentikasi mencoba membuktikan bahwa seseorang benar-benar seperti yang diklaimnya. Ada banyak cara yang memungkinkan untuk menyediakan autentikasi, tetapi seperti banyak tindakan pengamanan lainnya, metode yang lebih aman lebih sulit digunakan.

Teknik autentikasi meliputi kata sandi, tanda tangan digital, tindakan biometrik seperti pemindaian sidik jari, dan tindakan yang melibatkan perangkat keras seperti kartu pintar. Hanya dua yang umum digunakan di Web: kata sandi dan tanda tangan digital.

Tindakan biometrik dan sebagian besar solusi perangkat keras melibatkan perangkat input khusus dan akan membatasi pengguna yang sah pada mesin tertentu yang memiliki perangkat tersebut. Ini mungkin dapat diterima, atau bahkan diinginkan, untuk akses ke sistem internal organisasi, tetapi menghilangkan banyak keuntungan dari penyediaan sistem melalui Web.

Kata sandi mudah diterapkan, mudah digunakan, dan tidak memerlukan perangkat input khusus. Kata sandi menyediakan beberapa tingkat autentikasi, tetapi mungkin tidak sesuai untuk sistem keamanan tinggi.

Kata sandi adalah konsep sederhana. Anda dan sistem mengetahui kata sandi Anda. Jika pengunjung mengaku sebagai Anda, dan mengetahui kata sandi Anda, sistem memiliki alasan untuk percaya bahwa dia adalah Anda. Selama tidak ada orang lain yang mengetahui atau dapat menebak kata sandinya, ini aman. Kata sandi sendiri memiliki sejumlah kelemahan potensial dan tidak menyediakan autentikasi yang kuat.

Banyak kata sandi yang mudah ditebak. Jika dibiarkan memilih kata sandi mereka sendiri, sekitar 50% pengguna akan memilih kata sandi yang mudah ditebak. Kata sandi umum yang sesuai dengan deskripsi ini mencakup kata-kata kamus atau nama pengguna untuk akun tersebut. Dengan mengorbankan kegunaan, Anda dapat memaksa pengguna untuk menyertakan angka atau tanda baca dalam kata sandi mereka, tetapi ini akan menyebabkan beberapa pengguna mengalami kesulitan mengingat kata sandi mereka. Mendidik pengguna untuk memilih kata sandi yang lebih baik dapat membantu, tetapi bahkan ketika diberi pendidikan, sekitar 25% pengguna akan tetap memilih kata sandi yang mudah ditebak.

Anda dapat menerapkan kebijakan kata sandi yang menghentikan pengguna memilih kombinasi yang mudah ditebak dengan memeriksa kata sandi baru terhadap kamus, atau mengharuskan beberapa angka atau simbol tanda baca atau campuran huruf besar dan kecil. Salah satu bahayanya adalah bahwa aturan kata sandi yang ketat akan menghasilkan kata sandi yang tidak dapat diingat oleh banyak pengguna yang sah. Kata sandi yang sulit diingat meningkatkan kemungkinan pengguna melakukan tindakan yang tidak aman seperti menulis "nama pengguna fred password rover" pada catatan tempel di monitor mereka.

Pengguna perlu diberi edukasi untuk tidak menuliskan kata sandi mereka atau melakukan hal-hal konyol lainnya seperti memberikannya kepada orang melalui telepon yang menelepon dan mengaku sedang bekerja di sistem.

Kata sandi juga dapat ditangkap secara elektronik. Dengan menjalankan program untuk menangkap penekanan tombol di terminal atau menggunakan packet sniffer untuk menangkap lalu lintas jaringan, peretas dapat—dan memang—menangkap pasangan nama login dan kata sandi yang dapat digunakan. Anda dapat membatasi peluang untuk menangkap kata sandi dengan mengenkripsi lalu lintas jaringan.

Dengan segala potensi kekurangannya, kata sandi adalah cara yang sederhana dan relatif efektif untuk mengautentikasi pengguna Anda. Kata sandi memberikan tingkat kerahasiaan yang mungkin tidak sesuai untuk keamanan nasional, tetapi ideal untuk memeriksa status pengiriman pesanan pelanggan.

Menggunakan Autentikasi

Mekanisme autentikasi sudah terpasang di sebagian besar peramban web dan server web yang populer. Server web mungkin memerlukan nama pengguna dan kata sandi bagi orang yang meminta file dari direktori tertentu di server.

Saat dimintai nama login dan kata sandi, browser Anda akan menampilkan kotak dialog yang tampak seperti yang ditunjukkan pada Gambar 7.2.

Gambar 7.2 Peramban web meminta pengguna untuk melakukan autentikasi saat mereka mencoba mengunjungi direktori terbatas di server web.

Baik server web Apache maupun IIS Microsoft memungkinkan Anda untuk melindungi sebagian atau seluruh situs dengan cara ini. Dengan menggunakan PHP atau MySQL, ada banyak cara lain untuk mencapai efek yang sama. Menggunakan MySQL lebih cepat daripada autentikasi bawaan. Dengan menggunakan PHP, kita dapat menyediakan autentikasi yang lebih fleksibel atau menyajikan permintaan dengan cara yang lebih menarik.

Dasar-dasar Enkripsi

Algoritma enkripsi adalah proses matematika untuk mengubah informasi menjadi serangkaian data yang tampak acak. Data yang Anda mulai sering disebut teks biasa, meskipun tidak penting bagi proses tersebut apa yang diwakili oleh informasi tersebut—apakah itu benar-benar teks, atau jenis data lainnya.

Serupa dengan itu, informasi terenkripsi disebut ciphertext, tetapi jarang terlihat seperti teks. Gambar 7.3 menunjukkan proses enkripsi sebagai diagram alir sederhana. Teks biasa dimasukkan ke mesin enkripsi, yang mungkin dulunya merupakan perangkat mekanis,

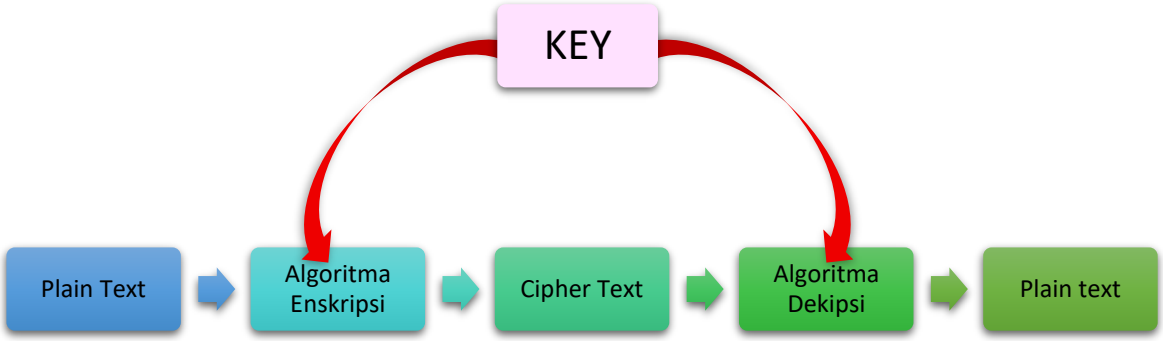
seperti mesin Engima Perang Dunia II, tetapi sekarang hampir selalu berupa program komputer. Mesin tersebut menghasilkan teks sandi.

Gambar 7.3 Enkripsi mengambil teks biasa dan mengubahnya menjadi teks sandi yang tampak acak

Untuk membuat direktori terproteksi yang perintah autentikasinya ditunjukkan pada Gambar 7.2, kami menggunakan jenis autentikasi Apache yang paling dasar. (Anda akan melihat cara menggunakannya di bab berikutnya.) Ini mengenkripsi kata sandi sebelum menyimpannya. Kami membuat pengguna dengan kata sandi password. Ini dienkripsi dan disimpan sebagai aWDuA3X3H.mc2. Anda dapat melihat bahwa teks biasa dan teks sandi tidak memiliki kemiripan yang jelas satu sama lain.

Metode enkripsi khusus ini tidak dapat dibalik. Banyak kata sandi disimpan menggunakan algoritma enkripsi satu arah. Untuk melihat apakah upaya memasukkan kata sandi benar, kami tidak perlu mendekripsi kata sandi yang tersimpan. Sebaliknya, kami dapat mengenkripsi upaya tersebut dan membandingkannya dengan versi yang tersimpan.

Banyak, tetapi tidak semua proses enkripsi dapat dibalik. Proses sebaliknya disebut dekripsi. Gambar 7.4 menunjukkan proses enkripsi dua arah.



Gambar 7.4 Enkripsi mengambil teks biasa dan mengubahnya menjadi teks sandi yang tampak acak. Dekripsi mengambil teks sandi dan mengubahnya kembali menjadi teks biasa.

Kriptografi berusia hampir 4000 tahun, tetapi mencapai puncaknya pada Perang Dunia II. Pertumbuhannya sejak saat itu mengikuti pola yang sama dengan adopsi jaringan komputer, awalnya hanya digunakan oleh perusahaan militer dan keuangan, kemudian lebih banyak digunakan oleh perusahaan mulai tahun 1970-an, dan menjadi umum pada tahun 1990-an. Dalam beberapa tahun terakhir, enkripsi telah berubah dari konsep yang hanya dilihat orang awam dalam film Perang Dunia II dan film mata-mata menjadi sesuatu yang mereka baca di surat kabar dan gunakan setiap kali mereka membeli sesuatu dengan peramban Web mereka. Tersedia banyak algoritme enkripsi yang berbeda. Beberapa, seperti DES, menggunakan kunci

rahasia atau kunci pribadi; beberapa, seperti RSA, menggunakan kunci publik dan kunci pribadi yang terpisah.

Enkripsi Kunci Pribadi

Enkripsi kunci pribadi bergantung pada orang yang berwenang yang mengetahui atau memiliki akses ke kunci. Kunci ini harus dirahasiakan. Jika kunci jatuh ke tangan yang salah, orang yang tidak berwenang juga dapat membaca pesan terenkripsi Anda. Seperti yang ditunjukkan pada Gambar 7.4, baik pengirim (yang mengenkripsi pesan) maupun penerima (yang mendekripsi pesan) memiliki kunci yang sama.

Algoritme kunci rahasia yang paling banyak digunakan adalah Data Encryption Standard (DES). Skema ini dikembangkan oleh IBM pada tahun 1970-an dan diadopsi sebagai standar Amerika untuk komunikasi pemerintah komersial dan tidak rahasia. Kecepatan komputasi jauh lebih cepat sekarang daripada tahun 1970, dan DES sudah usang setidaknya sejak tahun 1998.

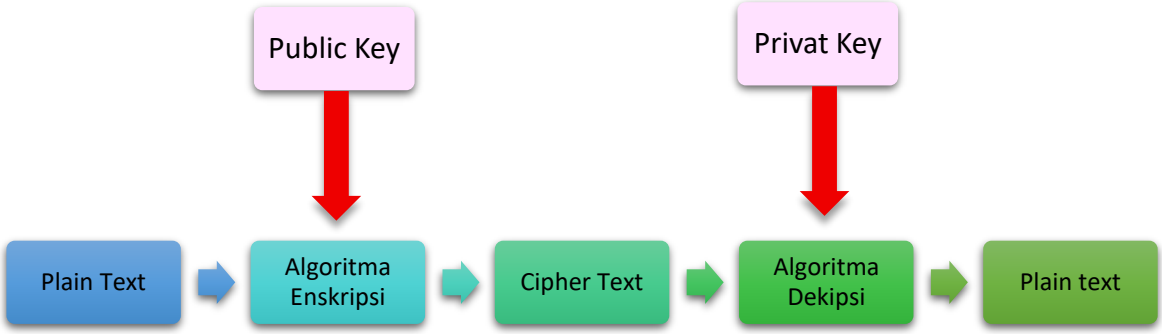
Sistem kunci rahasia terkenal lainnya termasuk RC2, RC4, RC5, triple DES, dan IDEA. Triple DES cukup aman. Ia menggunakan algoritme yang sama seperti DES, diterapkan tiga kali dengan hingga tiga kunci yang berbeda.

Pesan teks biasa dienkripsi dengan kunci satu, didekripsi dengan kunci dua, dan kemudian dienkripsi dengan kunci tiga. Satu kelemahan yang jelas dari enkripsi kunci rahasia adalah, untuk mengirim pesan aman kepada seseorang, Anda memerlukan cara aman untuk menyampaikan kunci rahasia kepadanya. Jika Anda memiliki cara aman untuk menyampaikan kunci, mengapa tidak menyampaikan pesan dengan cara itu saja?

Untungnya, ada terobosan pada tahun 1976, ketika Diffie dan Hellman menerbitkan skema kunci publik pertama.

Enkripsi Kunci Publik

Enkripsi kunci publik bergantung pada dua kunci yang berbeda, kunci publik dan kunci privat. Seperti yang ditunjukkan pada Gambar 7.5, kunci publik digunakan untuk mengenkripsi pesan, dan kunci privat untuk mendekripsinya.



Gambar 7.5 Enkripsi kunci publik menggunakan kunci terpisah untuk enkripsi dan dekripsi.

Keuntungan sistem ini adalah bahwa kunci publik, seperti namanya, dapat didistribusikan secara publik. Siapa pun yang Anda beri kunci publik dapat mengirimi Anda pesan yang aman. Selama hanya Anda yang memiliki kunci pribadi, maka hanya Anda yang

dapat mendekripsi pesan tersebut. Algoritma kunci publik yang paling umum adalah RSA, yang dikembangkan oleh Rivest, Shamir, dan Adelman di MIT dan dipublikasikan pada tahun 1978. RSA merupakan sistem hak milik, tetapi patennya berakhir pada bulan September 2000.

Kemampuan untuk mengirimkan kunci publik secara jelas dan tidak perlu khawatir akan terlihat oleh pihak ketiga merupakan keuntungan besar, tetapi sistem kunci rahasia masih umum digunakan. Sering kali, sistem hibrida digunakan. Sistem kunci publik digunakan untuk mengirimkan kunci bagi sistem kunci rahasia yang akan digunakan untuk sisa komunikasi sesi. Kompleksitas tambahan ini ditoleransi karena sistem kunci rahasia sekitar 1000 kali lebih cepat daripada sistem kunci publik.

Tanda Tangan Digital

Tanda tangan digital terkait dengan kriptografi kunci publik, tetapi membalikkan peran kunci publik dan privat. Pengirim dapat mengenkripsi dan menandatangani pesan secara digital dengan kunci rahasianya. Saat pesan diterima, penerima dapat mendekripsinya dengan kunci publik pengirim. Karena pengirim adalah satu-satunya orang yang memiliki akses ke kunci rahasia, penerima dapat cukup yakin dari siapa pesan itu berasal dan bahwa pesan itu belum diubah.

Tanda tangan digital dapat sangat berguna. Tanda tangan digital memungkinkan penerima yakin bahwa pesan itu tidak dirusak, dan mempersulit pengirim untuk menolak, atau menyangkal pengiriman, pesan tersebut.

Penting untuk dicatat bahwa meskipun pesan telah dienkripsi, pesan itu dapat dibaca oleh siapa saja yang memiliki kunci publik. Meskipun teknik dan kunci yang sama digunakan, tujuan enkripsi di sini adalah untuk mencegah pemalsuan dan penolakan, bukan untuk mencegah pembacaan. Karena enkripsi kunci publik cukup lambat untuk pesan besar, jenis algoritma lain, yang disebut fungsi hash, biasanya digunakan untuk meningkatkan efisiensi.

Fungsi hash menghitung intisari pesan atau nilai hash untuk setiap pesan yang diberikan. Tidak penting nilai apa yang dihasilkan algoritma. Yang penting adalah outputnya deterministik, yaitu, outputnya sama setiap kali input tertentu digunakan, outputnya kecil, dan algoritmanya cepat.

Fungsi hash yang paling umum adalah MD5 dan SHA.

Fungsi hash menghasilkan intisari pesan yang cocok dengan pesan tertentu. Jika Anda memiliki pesan dan intisari pesan, Anda dapat memverifikasi bahwa pesan tersebut belum dirusak, selama Anda yakin bahwa intisari tersebut belum dirusak.

Untuk tujuan ini, cara umum untuk membuat tanda tangan digital adalah dengan membuat intisari pesan untuk seluruh pesan menggunakan fungsi hash yang cepat, lalu mengenkripsi hanya intisari singkat menggunakan algoritma enkripsi kunci publik yang lambat. Tanda tangan sekarang dapat dikirim bersama pesan melalui metode normal yang tidak aman.

Saat pesan yang ditandatangani diterima, pesan tersebut dapat diperiksa. Tanda tangan didekripsi menggunakan kunci publik pengirim. Nilai hash dibuat untuk pesan tersebut menggunakan metode yang sama dengan yang digunakan pengirim. Jika nilai hash yang

didekripsi cocok dengan nilai hash yang Anda buat, maka pesan tersebut berasal dari pengirim dan belum diubah.

Sertifikat Digital

Adalah baik untuk dapat memverifikasi bahwa pesan belum diubah dan bahwa serangkaian pesan semuanya berasal dari pengguna atau mesin tertentu. Untuk interaksi komersial, akan lebih baik lagi jika dapat menghubungkan pengguna atau server tersebut ke badan hukum nyata seperti orang atau perusahaan.

Sertifikat digital menggabungkan kunci publik dan detail individu atau organisasi dalam format digital yang ditandatangani. Dengan sertifikat, Anda memiliki kunci publik pihak lain, jika Anda ingin mengirim pesan terenkripsi, dan Anda memiliki detail pihak tersebut, yang Anda tahu belum diubah.

Masalahnya di sini adalah bahwa informasi tersebut hanya dapat dipercaya jika orang yang menandatanganinya. Siapa pun dapat membuat dan menandatangani sertifikat yang mengklaim sebagai siapa pun yang disukainya. Untuk transaksi komersial, akan berguna untuk meminta pihak ketiga tepercaya memverifikasi identitas peserta dan detail yang tercatat dalam sertifikat mereka.

Pihak ketiga ini disebut Otoritas Sertifikasi (CA). Otoritas Sertifikasi menerbitkan sertifikat digital kepada individu dan perusahaan yang tunduk pada pemeriksaan identitas. Dua CA yang paling terkenal adalah VeriSign (http://www.verisign.com/) dan Thawte (http://www.thawte.com/), tetapi ada sejumlah otoritas lainnya. VeriSign dan Thawte keduanya dimiliki oleh perusahaan yang sama, dan hanya ada sedikit perbedaan praktis di antara keduanya. Beberapa otoritas yang kurang dikenal, seperti Equifax Secure (www.equifaxsecure.com), jauh lebih murah.

Otoritas menandatangani sertifikat untuk memverifikasi bahwa mereka telah melihat bukti identitas orang atau perusahaan tersebut. Perlu dicatat bahwa sertifikat tersebut bukanlah referensi atau pernyataan kelayakan kredit. Itu tidak menjamin bahwa Anda berurusan dengan seseorang yang memiliki reputasi baik. Artinya, jika Anda ditipu, Anda memiliki peluang yang cukup besar untuk memiliki alamat fisik yang sebenarnya dan seseorang untuk dituntut.

Sertifikat menyediakan jaringan kepercayaan. Dengan asumsi Anda memilih untuk mempercayai CA, Anda kemudian dapat memilih untuk mempercayai orang-orang yang mereka pilih untuk dipercayai dan kemudian mempercayai orang-orang yang dipilih oleh pihak yang disertifikasi untuk dipercayai.

Gambar 7.6 menunjukkan jalur sertifikat yang ditampilkan Internet Explorer untuk sertifikat tertentu. Dari sini, Anda dapat melihat bahwa www.equifaxsecure.com memiliki sertifikat yang diterbitkan oleh Equifax Secure E-Business Certifying Authority. CA ini, pada gilirannya, memiliki sertifikat yang diterbitkan oleh Thawte Server Certifying Authority.

Gambar 7.6 Jalur sertifikat untuk www.equifaxsecure.com menunjukkan jaringan kepercayaan yang memungkinkan kita mempercayai situs ini.

Penggunaan sertifikat digital yang paling umum adalah untuk memberikan kesan terhormat pada situs e-commerce. Dengan sertifikat yang dikeluarkan oleh CA terkenal, peramban web dapat membuat koneksi SSL ke situs Anda tanpa memunculkan dialog peringatan. Server web yang mengaktifkan koneksi SSL sering disebut server web aman.

7.4 SERVER WEB AMAN

Anda dapat menggunakan server web Apache, Microsoft IIS, atau sejumlah server web gratis atau komersial lainnya untuk komunikasi aman dengan peramban melalui Secure Sockets Layer. Agar dapat menggunakan SSL secara efektif, Anda juga memerlukan sertifikat yang dikeluarkan oleh otoritas sertifikasi.

Proses pasti untuk mendapatkannya bervariasi di antara CA, tetapi secara umum, Anda perlu membuktikan kepada CA bahwa Anda adalah semacam bisnis yang diakui secara hukum dengan alamat fisik dan bahwa bisnis yang dimaksud memiliki nama domain yang relevan.

Anda perlu membuat Permintaan Penandatanganan Sertifikat. Proses untuk ini akan bervariasi dari satu server ke server lainnya. Petunjuknya ada di situs Web CA. Stronghold dan IIS menyediakan proses yang digerakkan oleh kotak dialog, sedangkan Apache mengharuskan Anda mengetik perintah. Namun, prosesnya pada dasarnya sama untuk semua server. Hasil akhirnya adalah permintaan penandatanganan sertifikat (CSR) terenkripsi. CSR Anda akan terlihat seperti ini:

```
-----BEGIN NEW CERTIFICATE REQUEST-----
MIIBuwIBAAKBgQCLn1XX8faMHhtzStp9wY6BVTPuEU9bpMmhrb6vgaNZy4dTe6VS
84p7wGepq5CQjfOL4Hjda+g12xzto8uxBkCDO98Xg9q86CY45HZk+q6GyGOLZSOD
8cQHwh1oUP65s5Tz018OFBzpI3bHxfO6aYelWYziDiFKp1BrUdua+pK4SQIVAPLH
SV9FSz8Z7IHOg1Zr5H82oQOlAoGAWSPWyfVXPAF8h2GDb+cf97k44VkHZ+Rxpe8G
```

```
ghlfBn9L3ESWUZNOJMfDLlny7dStYU98VTVNekidYuaBsvyEkFrny7NCUmiuaSnX
4UjtFDkNhX9j5YbCRGLmsc865AT54KRu31O2/dKHLo6NgFPirijHy99HJ4LRY9Z9
HkXVzswCgYBwBFH2QfK88C6JKW3ah+6cHQ4Deoiltxi627WN5HcQLwkPGn+WtYSZ
jG5tw4tqqogmJ+IP2F/5G6FI2DQP7QDvKNeAU8jXcuijuWo27S2sbhQtXgZRTZvO
jGn89BC0mIHgHQMkI7vz35mx1Skk3VNq3ehwhGCvJlvoeiv2J8X2IQIVAOTRp7zp
En7QlXnXw1s7xXbbuKP0
-----END NEW CERTIFICATE REQUEST-----
```

Berbekal CSR, biaya yang sesuai, dan dokumentasi untuk membuktikan keberadaan Anda, serta setelah memverifikasi bahwa nama domain yang Anda gunakan memiliki nama yang sama dengan yang tercantum dalam dokumentasi bisnis, Anda dapat mendaftar untuk mendapatkan sertifikat dengan CA.

Saat CA menerbitkan sertifikat, Anda perlu menyimpannya di sistem dan memberi tahu server Web tempat menemukannya. Sertifikat akhir berupa berkas teks yang sangat mirip dengan CSR yang ditampilkan sebelumnya.

Pengauditan dan Pencatatan

Sistem operasi akan memungkinkan Anda mencatat semua jenis peristiwa. Peristiwa yang mungkin menarik bagi Anda dari sudut pandang keamanan meliputi kesalahan jaringan, akses ke berkas data tertentu seperti berkas konfigurasi atau registri NT, dan panggilan ke program seperti su (yang digunakan untuk menjadi pengguna lain, biasanya root, pada sistem UNIX).

Berkas log dapat membantu Anda mendeteksi perilaku yang salah atau berbahaya saat terjadi. Berkas log juga dapat memberi tahu Anda bagaimana masalah atau pembobolan terjadi jika Anda memeriksanya setelah menemukan masalah. Ada dua masalah utama dengan berkas log: ukuran dan kebenaran. Jika Anda menetapkan kriteria untuk mendeteksi dan mencatat masalah pada tingkat yang paling paranoid, Anda akan berakhir dengan log besar yang sangat sulit diperiksa. Untuk membantu berkas log yang besar, Anda benar-benar perlu menggunakan alat yang ada atau memperoleh beberapa skrip audit dari kebijakan keamanan Anda untuk mencari log untuk peristiwa yang "menarik". Proses audit dapat terjadi secara real-time, atau dapat dilakukan secara berkala.

Berkas log rentan terhadap serangan. Jika seorang penyusup memiliki akses root atau administrator ke sistem Anda, ia bebas mengubah berkas log untuk menutupi jejaknya. UNIX menyediakan fasilitas untuk mencatat peristiwa ke mesin yang terpisah. Ini berarti bahwa seorang cracker perlu membahayakan setidaknya dua mesin untuk menutupi jejaknya. Fungsionalitas serupa dimungkinkan di NT, tetapi tidak mudah.

Administrator sistem Anda mungkin melakukan audit rutin, tetapi Anda mungkin ingin melakukan audit eksternal secara berkala untuk memeriksa perilaku administrator.

Firewall

Firewall dalam jaringan dirancang untuk memisahkan jaringan Anda dari dunia yang lebih luas. Dengan cara yang sama seperti firewall di gedung atau mobil menghentikan api agar tidak menyebar ke kompartemen lain, firewall jaringan menghentikan kekacauan agar tidak menyebar ke jaringan Anda. Firewall dirancang untuk melindungi mesin di jaringan Anda dari

serangan luar. Firewall menyaring dan menolak lalu lintas yang tidak memenuhi aturannya. Firewall membatasi aktivitas orang dan mesin di luar firewall.

Terkadang, firewall juga digunakan untuk membatasi aktivitas orang-orang di dalamnya. Firewall dapat membatasi protokol jaringan yang dapat digunakan orang, membatasi host yang dapat mereka hubungi, atau memaksa mereka menggunakan server proxy untuk menekan biaya bandwidth. Firewall dapat berupa perangkat keras, seperti router dengan aturan penyaringan, atau program perangkat lunak yang berjalan di mesin.

Dalam kasus apa pun, firewall memerlukan antarmuka ke dua jaringan dan serangkaian aturan. Firewall memantau semua lalu lintas yang mencoba melewati satu jaringan ke jaringan lainnya. Jika lalu lintas memenuhi aturan, lalu lintas tersebut diarahkan ke jaringan lain; jika tidak, lalu lintas tersebut dihentikan atau ditolak. Paket dapat difilter menurut jenisnya, alamat sumber, alamat tujuan, atau informasi port. Beberapa paket akan dibuang begitu saja sementara kejadian tertentu dapat memicu entri log atau alarm.

Mencadangkan Data

Anda tidak dapat meremehkan pentingnya pencadangan dalam rencana pemulihan bencana apa pun. Peranti keras dan bangunan dapat diasuransikan dan diganti, atau situs dihosting di tempat lain, tetapi jika perangkat lunak Web yang Anda kembangkan sendiri hilang, tidak ada perusahaan asuransi yang dapat menggantinya untuk Anda.

Anda perlu mencadangkan semua komponen situs Web Anda--halaman statis, skrip, dan basis data--secara teratur. Seberapa sering Anda melakukan ini bergantung pada seberapa dinamis situs Anda. Jika semuanya statis, Anda dapat mencadangkannya saat berubah. Namun, jenis situs yang kita bahas dalam buku ini cenderung sering berubah, terutama jika Anda menerima pesanan daring.

Sebagian besar situs dengan ukuran yang wajar perlu dihosting di server dengan RAID (*Redundant Array of Inexpensive Disks*), yang dapat mendukung mirroring. Ini mencakup situasi saat Anda mungkin mengalami kegagalan hard disk. Namun, pertimbangkan apa yang mungkin terjadi dalam situasi saat sesuatu terjadi pada seluruh array, mesin, atau gedung.

Anda harus menjalankan pencadangan terpisah pada frekuensi yang sesuai dengan volume pembaruan Anda. Pencadangan ini harus disimpan di media terpisah, dan sebaiknya di lokasi yang aman dan terpisah, untuk berjaga-jaga jika terjadi kebakaran, pencurian, atau bencana alam.

Banyak sumber daya di luar sana tentang pencadangan dan pemulihan. Kami akan berkonsentrasi pada cara Anda dapat mencadangkan situs yang dibangun dengan PHP dan basis data MySQL.

Mencadangkan File Umum

Mencadangkan HTML, PHP, gambar, dan file non-basis data lainnya dapat dilakukan dengan cukup mudah di sebagian besar sistem dengan menggunakan perangkat lunak pencadangan. Utilitas yang tersedia secara gratis yang paling banyak digunakan adalah AMANDA, Advanced Maryland Automated Network Disk Archiver, yang dikembangkan oleh University of Maryland. Ia disertakan dengan banyak distribusi UNIX dan juga dapat digunakan untuk mencadangkan mesin Windows melalui SAMBA.

Keamanan Fisik

Ancaman keamanan yang telah kita bahas sejauh ini terkait dengan hal-hal yang tidak berwujud seperti perangkat lunak, tetapi Anda tidak boleh mengabaikan keamanan fisik sistem Anda. Anda memerlukan pendingin udara, dan perlindungan terhadap kebakaran, orang-orang (baik yang ceroboh maupun penjahat), pemadaman listrik, dan kegagalan jaringan.

Sistem Anda harus dikunci dengan aman. Bergantung pada skala operasi Anda, ini bisa berarti ruangan, kandang, atau lemari. Personel yang tidak memerlukan akses ke ruang mesin ini tidak boleh memilikinya. Orang yang tidak berwenang mungkin dengan sengaja atau tidak sengaja mencabut kabel atau mencoba melewati mekanisme keamanan menggunakan disk yang dapat di-boot.

Penyiram air dapat menyebabkan kerusakan pada perangkat elektronik seperti halnya kebakaran. Di masa lalu, sistem pencegah kebakaran halon digunakan untuk menghindari masalah ini. Produksi halon sekarang dilarang berdasarkan Protokol Montreal tentang Zat-zat yang Merusak Lapisan Ozon, jadi sistem pencegah kebakaran baru harus menggunakan alternatif lain yang tidak terlalu berbahaya seperti argon atau karbon dioksida.

Seperti pemadaman listrik, pemadaman jaringan selama beberapa menit atau jam berada di luar kendali Anda dan pasti akan terjadi sesekali. Jika jaringan Anda vital, masuk akal untuk memiliki koneksi ke lebih dari satu penyedia layanan Internet. Akan lebih mahal untuk memiliki dua koneksi, tetapi seharusnya berarti bahwa, jika terjadi kegagalan, Anda memiliki kapasitas yang berkurang daripada menjadi tidak terlihat. Masalah-masalah semacam ini adalah beberapa alasan mengapa Anda mungkin ingin mempertimbangkan untuk menempatkan mesin-mesin Anda di fasilitas khusus. Meskipun satu bisnis berukuran sedang mungkin tidak dapat membenarkan penggunaan UPS yang akan beroperasi selama lebih dari beberapa menit, beberapa koneksi jaringan redundan, dan sistem pemadam kebakaran, fasilitas berkualitas yang menampung mesin-mesin dari seratus bisnis serupa dapat melakukannya.

BAB 8
MENERAPKAN AUTENTIKASI DENGAN PHP DAN MYSQL

Bab ini akan membahas cara menerapkan berbagai teknik PHP dan MySQL untuk mengautentikasi pengguna.

Yang akan dibahas dalam bab ini meliputi:

- Mengidentifikasi pengunjung
- Mengimplementasikan kontrol akses
- Autentikasi dasar
- Menggunakan autentikasi dasar dalam PHP
- Menggunakan autentikasi dasar .htaccess Apache
- Menggunakan autentikasi dasar dengan IIS
- Menggunakan autentikasi mod_auth_mysql
- Membuat autentikasi kustom Anda sendiri

8.1 MENGIDENTIFIKASI PENGUNJUNG

Web adalah media yang cukup anonim, tetapi sering kali berguna untuk mengetahui siapa yang mengunjungi situs Anda. Untungnya untuk privasi pengunjung, Anda dapat mengetahui sangat sedikit tentang mereka tanpa bantuan mereka. Dengan sedikit usaha, server dapat mengetahui cukup banyak tentang komputer dan jaringan yang terhubung dengannya. Peramban web biasanya akan mengidentifikasi dirinya sendiri, memberi tahu server peramban, versi peramban, dan sistem operasi apa yang Anda jalankan. Anda dapat menentukan resolusi dan kedalaman warna layar pengunjung dan seberapa besar jendela peramban Web mereka.

Setiap komputer yang terhubung ke Internet memiliki alamat IP yang unik. Dari alamat IP pengunjung, Anda mungkin dapat menyimpulkan sedikit tentangnya. Anda dapat mengetahui siapa yang memiliki IP dan terkadang memiliki perkiraan yang masuk akal mengenai lokasi geografis pengunjung. Beberapa alamat akan lebih berguna daripada yang lain. Umumnya orang dengan koneksi Internet permanen akan memiliki alamat permanen. Pelanggan yang menghubungi ISP biasanya hanya akan mendapatkan penggunaan sementara dari salah satu alamat ISP. Lain kali Anda melihat alamat itu, alamat itu mungkin digunakan oleh komputer yang berbeda, dan lain kali Anda melihat pengunjung itu, ia mungkin akan menggunakan alamat IP yang berbeda.

Beruntung bagi pengguna Web, tidak ada informasi yang diberikan peramban mereka yang mengidentifikasi mereka. Jika Anda ingin mengetahui nama pengunjung atau detail lainnya, Anda harus bertanya kepadanya.

Banyak situs Web memberikan alasan yang kuat untuk meminta pengguna memberikan detail mereka. Surat kabar New York Times (http://www.nytimes.com) menyediakan kontennya secara gratis, tetapi hanya untuk orang-orang yang bersedia memberikan informasi seperti nama, jenis kelamin, dan total pendapatan rumah tangga. Situs

berita dan diskusi Slashdot (http://www.slashdot.org) memungkinkan pengguna terdaftar untuk berpartisipasi dalam diskusi dengan nama panggilan dan menyesuaikan antarmuka yang mereka lihat. Sebagian besar situs e-commerce mencatat informasi pelanggan mereka saat mereka melakukan pemesanan pertama. Ini berarti bahwa pelanggan tidak perlu mengetik informasi mereka setiap saat.

Setelah meminta dan menerima informasi dari pengunjung Anda, Anda memerlukan cara untuk mengaitkan informasi tersebut dengan pengguna yang sama saat ia berkunjung lagi. Jika Anda bersedia membuat asumsi bahwa hanya satu orang yang mengunjungi situs Anda dari akun tertentu pada komputer tertentu dan bahwa setiap pengunjung hanya menggunakan satu komputer, Anda dapat menyimpan cookie pada komputer pengguna untuk mengidentifikasi pengguna. Ini tentu saja tidak berlaku untuk semua pengguna—sering kali, banyak orang berbagi komputer, dan banyak orang menggunakan lebih dari satu komputer. Setidaknya pada beberapa waktu, Anda perlu bertanya kepada pengunjung tentang siapa dia lagi. Selain bertanya tentang siapa pengguna, Anda juga perlu meminta pengguna untuk memberikan sejumlah bukti bahwa dia adalah orang yang dia klaim.

Seperti yang dibahas dalam Bab 7, “Masalah Keamanan E-commerce,” meminta pengguna untuk membuktikan identitasnya disebut autentikasi. Metode autentikasi yang umum digunakan di situs web adalah meminta pengunjung untuk memberikan nama login dan kata sandi yang unik. Autentikasi biasanya digunakan untuk mengizinkan atau melarang akses ke halaman atau sumber daya tertentu, tetapi dapat bersifat opsional, atau digunakan untuk tujuan lain seperti personalisasi.

8.2 MENERAPKAN KONTROL AKSES

Kontrol akses sederhana tidak sulit diterapkan. Kode yang ditunjukkan pada Daftar listing 8.1 memberikan satu dari tiga kemungkinan keluaran.

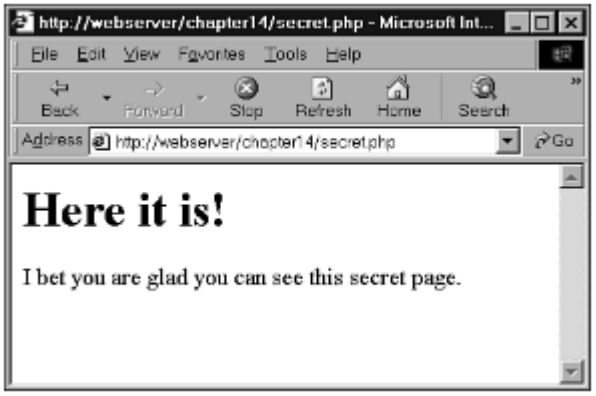
Gambar 8.1 Formulir HTML kami meminta pengunjung memasukkan nama pengguna dan kata sandi untuk akses.

Jika berkas dimuat tanpa parameter, berkas akan menampilkan formulir HTML yang meminta nama pengguna dan kata sandi. Jenis formulir ini ditunjukkan pada Gambar 8.1. Jika

parameter ada tetapi tidak benar, formulir akan menampilkan pesan kesalahan. Pesan kesalahan kami ditunjukkan pada Gambar 8.2.

Gambar 8.2 Saat pengguna memasukkan detail yang salah, kami perlu memberi mereka pesan kesalahan. Di situs yang sebenarnya, Anda mungkin ingin memberikan pesan yang lebih bersahabat.

Jika parameter ini ada dan benar, konten rahasia akan ditampilkan. Konten pengujian kami ditunjukkan pada Gambar 8.3.



Gambar 8.3 Saat diberikan detail yang benar, skrip kami akan menampilkan konten.

Kode untuk membuat fungsionalitas yang ditunjukkan pada Gambar 8.1, 8.2, dan 8.3 ditunjukkan pada Daftar 8.1.

Listing 14.1 secret.php—PHP dan HTML untuk Menyediakan Mekanisme Autentikasi Sederhana

```
<?
  if(!isset($name)&&!isset($password))
  {
    //Visitor needs to enter a name and password
?>
```

```
    <h1>Please Log In</h1>
    This page is secret.
  <form method = post action = “secret.php”>
  <table border = 1>
  <tr>
    <th> Username </th>
    <td> <input type = text name = name> </td>
  </tr>
  <tr>
    <th> Password </th>
    <td> <input type = password name = password> </td>
  </tr>
  <tr>
    <td colspan =2 align = center>
      <input type = submit value = “Log In”>
    </td>
  </tr>
  </table>
  </form>

<?
  }
  else if($name==”user”&&$password==”pass”)
  {
    // visitor’s name and password combination are correct
    echo “<h1>Here it is!</h1>”;
    echo “I bet you are glad you can see this secret page.”;
  }
  else
  {
    // visitor’s name and password combination are not correct
    echo “<h1>Go Away!</h1>”;
    echo “You are not authorized to view this resource.”;
  }
?>
```

Kode dari Listing 14.1 akan memberi Anda mekanisme autentikasi sederhana untuk mengizinkan pengguna yang berwenang melihat halaman, tetapi ada beberapa masalah yang signifikan.

Skrip ini

- Memiliki satu nama pengguna dan kata sandi yang dikodekan secara permanen ke dalam skrip
- Menyimpan kata sandi sebagai teks biasa
- Hanya melindungi satu halaman
- Mengirimkan kata sandi sebagai teks biasa

Semua masalah ini dapat diatasi dengan berbagai tingkat upaya dan keberhasilan.

8.3 MENYIMPAN KATA SANDI

Ada banyak tempat yang lebih baik untuk menyimpan nama pengguna dan kata sandi selain di dalam skrip. Di dalam skrip, sulit untuk mengubah data. Hal itu mungkin dilakukan, tetapi menulis skrip untuk mengubah dirinya sendiri adalah ide yang buruk. Itu berarti memiliki skrip di server Anda, yang dijalankan di server Anda, tetapi dapat ditulis atau diubah oleh orang lain. Menyimpan data di file lain di server akan memudahkan Anda menulis program untuk menambah dan menghapus pengguna serta mengubah kata sandi.

Di dalam skrip atau file data lain, ada batasan jumlah pengguna yang dapat Anda miliki tanpa memengaruhi kecepatan skrip secara serius. Jika Anda mempertimbangkan untuk menyimpan dan mencari melalui sejumlah besar item dalam sebuah file, Anda harus mempertimbangkan untuk menggunakan basis data, seperti yang dibahas sebelumnya. Sebagai aturan praktis, jika Anda ingin menyimpan dan mencari melalui daftar yang berisi lebih dari 100 item, item tersebut harus berada dalam basis data, bukan file datar.

Menggunakan basis data untuk menyimpan nama pengguna dan kata sandi tidak akan membuat skrip menjadi jauh lebih rumit, tetapi akan memungkinkan Anda untuk mengautentikasi banyak pengguna yang berbeda dengan cepat. Ini juga akan memudahkan Anda untuk menulis skrip untuk menambahkan pengguna baru, menghapus pengguna, dan mengizinkan pengguna untuk mengubah kata sandi mereka.

Skrip untuk mengautentikasi pengunjung ke suatu halaman terhadap suatu basis data diberikan dalam Listing 14.2.

Listing 14.2 secretdb.php—Kami Telah Menggunakan MySQL untuk Meningkatkan Mekanisme Autentikasi Sederhana Kami

```
<?
  if(!isset($name)&&!isset($password))
  {
    //Visitor needs to enter a name and password
?>
    <h1>Please Log In</h1>
    This page is secret.
    <form method = post action = “secretdb.php”>
    <table border = 1>
    <tr>
      <th> Username </th>
      <td> <input type = text name = name> </td>
    </tr>
    <tr>
      <th> Password </th>
      <td> <input type = password name = password> </td>
    </tr>
    <tr>
```

```
        <td colspan =2 align = center>
          <input type = submit value = “Log In”>
        </td>
      </tr>
      </table>
      </form>
<?
    }
    else
    {
      // connect to mysql
      $mysql = mysql_connect( ‘localhost’, ‘webauth’, ‘webauth’ );
      if(!$mysql)
    {
      echo ‘Cannot connect to database.’;
      exit;
    }
    // select the appropriate database
    $mysql = mysql_select_db( ‘auth’ );
    if(!$mysql)
    {
      echo ‘Cannot select database.’;
      exit;
    }

    // query the database to see if there is a record which matches
    $query = “select count(*) from auth where
              name = ‘$name’ and
              pass = ‘$password’”;

    $result = mysql_query( $query );
    if(!$result)
    {
      echo ‘Cannot run query.’;
      exit;
    }

    $count = mysql_result( $result, 0, 0 );

    if ( $count > 0 )
    {
      // visitor’s name and password combination are correct
      echo “<h1>Here it is!</h1>”;
      echo “I bet you are glad you can see this secret page.”;
    }
    else
    {
```

```
      // visitor's name and password combination are not correct
      echo "<h1>Go Away!</h1>";
      echo "You are not authorized to view this resource.";
    }
  }
  ?>
```

Basis data yang kita gunakan dapat dibuat dengan menghubungkan ke MySQL sebagai pengguna root MySQL dan menjalankan konten Listing 8.3.

Listing 8.3 `createauthdb.sql`—Kueri MySQL ini Membuat Basis Data auth, Tabel auth, dan Dua Contoh Pengguna

```
create database auth;

use auth;
create table auth (
       name          varchar(10) not null,
       pass          varchar(30) not null,
       primary key   (name)
);

insert into auth values
  ('user', 'pass');

insert into auth values
  ('testuser', password('test123') );

grant select, insert, update, delete
on auth.*
to webauth@localhost
identified by 'webauth';
```

8.4 MENGENKRIPSI KATA SANDI

Terlepas dari apakah kita menyimpan data dalam basis data atau berkas, menyimpan kata sandi sebagai teks biasa merupakan risiko yang tidak perlu. Algoritma hash satu arah dapat memberikan sedikit keamanan dengan sedikit usaha ekstra.

Fungsi PHP `crypt()` menyediakan fungsi hash kriptografi satu arah. Prototipe untuk fungsi ini adalah:

```
string crypt (string str [, string salt])
```

Jika string str diberikan, fungsi akan mengembalikan string pseudo-acak. Misalnya, jika string "`pass`" dan salt "`xx`" diberikan, `crypt()` akan mengembalikan "`xxkT1mYjlikoII`". String ini tidak dapat didekripsi dan diubah kembali menjadi "`pass`" bahkan oleh pembuatnya, jadi

mungkin tampak tidak terlalu berguna pada pandangan pertama. Properti yang membuat `crypt()` berguna adalah bahwa output bersifat deterministik. Jika string dan salt yang sama diberikan, `crypt()` akan mengembalikan hasil yang sama setiap kali dijalankan.

Daripada memiliki kode PHP seperti

```
if( $username == “user” && $password == “pass” )
{
  //OK passwords match
}
```

kita bisa memiliki kode seperti

```
if( $username == ‘user’ && crypt($password,’xx’) == ‘xxkT1mYjlikoII’ )
{
  //OK passwords match
}
```

Kita tidak perlu tahu seperti apa tampilan ‘`xxkT1mYjlikoII`’ sebelum kita menggunakan `crypt()` di dalamnya. Kita hanya perlu tahu apakah kata sandi yang diketik sama dengan kata sandi yang awalnya dijalankan melalui `crypt()`.

Seperti yang telah disebutkan, mengodekan nama pengguna dan kata sandi yang dapat diterima ke dalam skrip adalah ide yang buruk. Kita harus menggunakan file terpisah atau basis data untuk menyimpannya.

Jika kita menggunakan basis data MySQL untuk menyimpan data autentikasi, kita dapat menggunakan fungsi PHP `crypt()` atau fungsi MySQL `PASSWORD()`. Fungsi-fungsi ini tidak menghasilkan keluaran yang sama, tetapi dimaksudkan untuk melayani tujuan yang sama. Baik `crypt()` maupun `PASSWORD()` mengambil string dan menerapkan algoritma hash yang tidak dapat dibalik.

Untuk menggunakan `PASSWORD()`, kita dapat menulis ulang kueri SQL dalam Listing 8.2 sebagai

```
select count(*) from auth where
       name = ‘$name’ and
       pass = password(‘$password’)
```

Kueri ini akan menghitung jumlah baris dalam tabel auth yang memiliki nilai nama yang sama dengan konten `$name` dan nilai pass yang sama dengan output yang diberikan oleh `PASSWORD()` yang diterapkan pada konten `$password`. Dengan asumsi bahwa kita memaksa orang untuk memiliki nama pengguna yang unik, hasil dari kueri ini akan menjadi 0 atau 1.

Melindungi Beberapa Halaman

Membuat skrip seperti ini untuk melindungi lebih dari satu halaman sedikit lebih sulit. Karena HTTP tidak memiliki status, tidak ada tautan atau asosiasi otomatis antara permintaan

berikutnya dari orang yang sama. Hal ini mempersulit data, seperti informasi autentikasi yang dimasukkan pengguna, untuk dibawa dari satu halaman ke halaman lainnya.

Cara termudah untuk melindungi beberapa halaman adalah dengan menggunakan mekanisme kontrol akses yang disediakan oleh server Web Anda. Kita akan membahasnya sebentar lagi.

Untuk membuat sendiri fungsi ini, kita dapat menyertakan bagian skrip yang ditunjukkan dalam Listing 8.1 di setiap halaman yang ingin kita lindungi. Dengan menggunakan `auto_prepend_file` dan `auto_append_file`, kita dapat secara otomatis menambahkan kode yang diperlukan ke setiap file dalam direktori tertentu.

Jika kita menggunakan pendekatan ini, apa yang terjadi ketika pengunjung kita membuka beberapa halaman dalam situs kita? Tidaklah dapat diterima jika kita meminta mereka memasukkan kembali nama dan kata sandi mereka untuk setiap halaman yang ingin mereka lihat. Kita dapat menambahkan detail yang mereka masukkan ke setiap hyperlink di halaman. Karena pengguna mungkin memiliki spasi, atau karakter lain yang tidak diizinkan di URL, kita harus menggunakan fungsi `urlencode()` untuk mengodekan karakter ini dengan aman.

Namun, masih akan ada beberapa masalah dengan pendekatan ini. Karena data akan disertakan dalam halaman Web yang dikirim ke pengguna, dan URL yang mereka kunjungi, halaman yang dilindungi yang mereka kunjungi akan terlihat oleh siapa saja yang menggunakan komputer yang sama dan menelusuri kembali halaman yang di-cache atau melihat daftar riwayat browser. Karena kita mengirim kata sandi bolak-balik ke browser dengan setiap halaman yang diminta atau dikirim, informasi sensitif ini dikirimkan lebih sering dari yang diperlukan.

Ada dua cara yang baik untuk mengatasi masalah ini: autentikasi dasar HTTP dan sesi. Autentikasi dasar mengatasi masalah caching, tetapi browser masih mengirimkan kata sandi ke browser dengan setiap permintaan. Kontrol sesi mengatasi kedua masalah ini.

Autentikasi Dasar Untungnya, mengautentikasi pengguna adalah tugas umum, jadi ada fasilitas autentikasi yang terpasang di HTTP. Skrip atau server Web dapat meminta autentikasi dari browser Web. Browser Web kemudian bertanggung jawab untuk menampilkan kotak dialog atau perangkat serupa untuk mendapatkan informasi yang diperlukan dari pengguna. Meskipun server Web meminta detail autentikasi baru untuk setiap permintaan pengguna, browser Web tidak perlu meminta detail pengguna untuk setiap halaman. Browser umumnya menyimpan detail ini selama pengguna membuka jendela browser dan secara otomatis mengirimkannya kembali ke server Web sesuai kebutuhan tanpa interaksi pengguna. Fitur HTTP ini disebut autentikasi dasar. Anda dapat memicu autentikasi dasar menggunakan PHP, atau menggunakan mekanisme yang terpasang di server Web Anda. Kita akan melihat metode PHP, metode Apache, dan metode IIS.

Otentikasi dasar mengirimkan nama dan kata sandi pengguna dalam bentuk teks biasa, sehingga tidak terlalu aman. HTTP 1.1 berisi metode yang agak lebih aman yang dikenal sebagai autentikasi digest, yang menggunakan algoritma hashing (biasanya MD5) untuk menyamarkan detail transaksi. Otentikasi digest didukung oleh banyak server Web, tetapi tidak

didukung oleh sejumlah besar browser Web. Otentikasi digest telah didukung oleh Microsoft Internet Explorer sejak versi 5.0. Pada saat penulisan, autentikasi ini tidak didukung oleh versi Netscape Navigator mana pun, tetapi mungkin disertakan dalam versi 6.0.

Selain tidak didukung dengan baik oleh browser Web yang terpasang, autentikasi digest masih tidak terlalu aman. Baik autentikasi dasar maupun autentikasi digest memberikan tingkat keamanan yang rendah. Keduanya tidak memberikan jaminan apa pun kepada pengguna bahwa ia berurusan dengan mesin yang ingin diaksesnya. Keduanya mungkin mengizinkan cracker untuk memutar ulang permintaan yang sama ke server. Karena autentikasi dasar mengirimkan kata sandi pengguna sebagai teks biasa, cracker mana pun yang mampu menangkap paket dapat menyamar sebagai pengguna untuk membuat permintaan apa pun.

Autentikasi dasar menyediakan tingkat keamanan (rendah) yang mirip dengan yang umum digunakan untuk terhubung ke mesin melalui Telnet atau FTP, dengan mengirimkan kata sandi dalam bentuk teks biasa. Autentikasi intisari sedikit lebih aman, dengan mengenkripsi kata sandi sebelum mengirimkannya. Dengan menggunakan SSL dan sertifikat digital, semua bagian transaksi Web dapat dilindungi oleh keamanan yang kuat.

Jika Anda menginginkan keamanan yang kuat, Anda harus membaca bab berikutnya, Bab 9, “Menerapkan Transaksi Aman dengan PHP dan MySQL.” Namun, untuk banyak situasi, metode yang cepat, tetapi relatif tidak aman, seperti autentikasi dasar adalah tepat.

Autentikasi dasar melindungi ranah bernama dan mengharuskan pengguna untuk memberikan nama pengguna dan kata sandi yang valid. Ranah diberi nama sehingga lebih dari satu ranah dapat berada di server yang sama. File atau direktori yang berbeda di server yang sama dapat menjadi bagian dari ranah yang berbeda, masing-masing dilindungi oleh serangkaian nama dan kata sandi yang berbeda. Ranah bernama juga memungkinkan Anda mengelompokkan beberapa direktori pada satu host atau host virtual sebagai ranah dan melindungi semuanya dengan satu kata sandi.

8.5 MENGGUNAKAN AUTENTIKASI DASAR DALAM PHP

Skrip PHP umumnya lintas platform, tetapi penggunaan autentikasi dasar bergantung pada variabel lingkungan yang ditetapkan oleh server. Agar skrip autentikasi HTTP dapat berjalan di Apache menggunakan PHP sebagai Modul Apache atau di IIS menggunakan PHP sebagai modul ISAPI, skrip tersebut perlu mendeteksi jenis server dan berperilaku sedikit berbeda. Skrip dalam Listing 8.4 akan berjalan di kedua server.

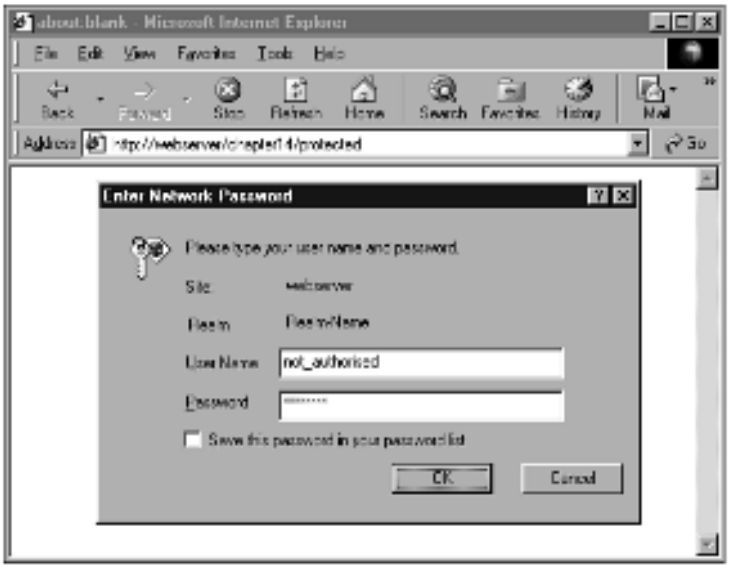
Listing 8.4 `http.php`—PHP Dapat Memicu Autentikasi Dasar HTTP

```
<?

// if we are using IIS, we need to set $PHP_AUTH_USER and $PHP_AUTH_PW
if (substr($SERVER_SOFTWARE, 0, 9) == “Microsoft” &&
     !isset($PHP_AUTH_USER) &&
     !isset($PHP_AUTH_PW) &&
     substr($HTTP_AUTHORIZATION, 0, 6) == “Basic “
```

```
    )
  {
    list($PHP_AUTH_USER, $PHP_AUTH_PW) =
      explode(“:”, base64_decode(substr($HTTP_AUTHORIZATION, 6)));
  }

  // Replace this if statement with a database query or similar
  if ($PHP_AUTH_USER != “user” || $PHP_AUTH_PW != “pass”)
  {
    // visitor has not yet given details, or their
    // name and password combination are not correct

    header(‘WWW-Authenticate: Basic realm=”Realm-Name”’);
    if (substr($SERVER_SOFTWARE, 0, 9) == “Microsoft”)
      header(“Status: 401 Unauthorized”);
    else
      header(“HTTP/1.0 401 Unauthorized”);

    echo “<h1>Go Away!</h1>”;
    echo “You are not authorized to view this resource.”;
  }
  else
  {
    // visitor has provided correct details
    echo “<h1>Here it is!</h1>”;
    echo “<p>I bet you are glad you can see this secret page.”;
  }
  ?>
```

Kode dalam Daftar 8.4 bertindak dengan cara yang sangat mirip dengan daftar sebelumnya dalam bab ini. Jika pengguna belum memberikan informasi autentikasi, maka akan diminta. Jika pengguna memberikan informasi yang salah, maka akan diberikan pesan penolakan. Jika pengguna memberikan pasangan nama-kata sandi yang cocok, maka akan diberikan konten halaman.

Pengguna akan melihat antarmuka yang agak berbeda dari daftar sebelumnya. Kami tidak menyediakan formulir HTML untuk informasi login. Peramban pengguna akan menampilkan kotak dialog. Sebagian orang melihat ini sebagai peningkatan; yang lain lebih suka memiliki kendali penuh atas aspek visual antarmuka. Kotak dialog login yang disediakan Internet Explorer ditunjukkan pada Gambar 8.4.



Gambar 8.4 Peramban pengguna bertanggung jawab atas tampilan kotak dialog saat menggunakan autentikasi HTTP.

Karena autentikasi dibantu oleh fitur-fitur yang ada di dalam peramban, peramban memilih untuk menjalankan beberapa kebijaksanaan dalam cara menangani upaya otorisasi yang gagal. Internet Explorer memungkinkan pengguna mencoba mengautentikasi tiga kali sebelum menampilkan halaman penolakan.

Netscape Navigator akan memungkinkan pengguna mencoba sebanyak yang tidak terbatas, memunculkan kotak dialog yang menanyakan, “Otorisasi gagal. Coba lagi?” di antara percobaan. Netscape hanya menampilkan halaman penolakan jika pengguna mengklik Batal.
Seperti halnya kode yang diberikan dalam Listing 8.1 dan 8.2, kita dapat menyertakan kode ini di halaman yang ingin kita lindungi, atau secara otomatis menambahkannya di depan setiap berkas dalam direktori.

Menggunakan Autentikasi Dasar dengan File .htaccess Apache

Kita dapat memperoleh hasil yang sangat mirip dengan skrip sebelumnya tanpa menulis skrip PHP. Server Web Apache berisi sejumlah modul autentikasi berbeda yang dapat digunakan untuk memutuskan validitas data yang dimasukkan oleh pengguna. Yang paling mudah digunakan adalah mod_auth, yang membandingkan pasangan nama-sandi dengan baris dalam file teks di server.

Untuk mendapatkan hasil yang sama seperti skrip sebelumnya, kita perlu membuat dua file HTML terpisah, satu untuk konten dan satu untuk halaman penolakan. Kita melewatkan beberapa elemen HTML dalam contoh sebelumnya, tetapi seharusnya menyertakan tag `<html>` dan `<body>` saat kita membuat HTML.

Listing 8.5 berisi konten yang dapat dilihat oleh pengguna yang berwenang. Kita menyebut file ini content.html. Listing 8.6 berisi halaman penolakan. Kita menyebutnya rejection.html. Memiliki halaman untuk ditampilkan jika terjadi kesalahan adalah opsional, tetapi akan lebih baik jika Anda menambahkan sesuatu yang berguna di dalamnya.

Mengingat bahwa halaman ini akan ditampilkan saat pengguna mencoba memasuki area yang dilindungi tetapi ditolak, konten yang bermanfaat mungkin mencakup petunjuk tentang cara mendaftar untuk mendapatkan kata sandi, atau cara mendapatkan pengaturan ulang kata sandi dan mengirimkannya melalui email jika kata sandi tersebut terlupa.

Listing 8.5 `content.html`—Contoh Konten Kami

```
<html><body>
<h1>Here it is!</h1>
<p>I bet you are glad you can see this secret page.
</body></html>
```

Listing 8.6 `rejection.html`–Contoh Halaman Kesalahan 401 Kami

```
<html><body>
<h1>Go Away!</h1>
<p>You are not authorized to view this resource.
</body></html>
```

Tidak ada yang baru dalam berkas-berkas ini. Berkas yang menarik untuk contoh ini adalah Listing 8.6. Berkas ini perlu disebut .htaccess, dan akan mengontrol akses ke berkas-berkas dan subdirektori apa pun dalam direktorinya.

Listing 8.7 `.htaccess`–File `.htaccess` Dapat Mengatur Banyak Pengaturan Konfigurasi Apache, Termasuk Mengaktifkan Autentikasi

```
ErrorDocument 401 /chapter14/rejection.html
AuthUserFile /home/book/.htpass
AuthGroupFile /dev/null
AuthName “Realm-Name”
AuthType Basic
require valid-user
```

Listing 8.7 adalah file `.htaccess` untuk mengaktifkan autentikasi dasar dalam sebuah direktori. Banyak pengaturan yang dapat dilakukan dalam file `.htaccess`, tetapi keenam baris dalam contoh kita semuanya terkait dengan autentikasi.

```
ErrorDocument 401 /chapter14/rejection.html
```

Baris pertama memberi tahu Apache dokumen apa yang akan ditampilkan bagi pengunjung yang gagal melakukan autentikasi. Anda dapat menggunakan perintah `ErrorDocument` lain untuk menyediakan halaman Anda sendiri untuk kesalahan HTTP lainnya seperti 404. Sintaksnya adalah `ErrorDocument error_number URL`

Agar halaman dapat menangani kesalahan 401, penting bahwa URL yang diberikan tersedia untuk umum. Tidak akan terlalu berguna dalam menyediakan halaman kesalahan

yang disesuaikan untuk memberi tahu orang-orang bahwa otorisasi mereka gagal jika halaman terkunci dalam direktori tempat mereka perlu berhasil melakukan autentikasi untuk melihatnya. Baris `AuthUserFile /home/book/.htpass` memberi tahu Apache tempat menemukan file yang berisi kata sandi pengguna yang diotorisasi. Ini sering disebut .htpass, tetapi Anda dapat memberinya nama apa pun yang Anda inginkan. Tidak penting nama berkas ini, tetapi yang penting adalah tempat penyimpanannya. Berkas ini tidak boleh disimpan di dalam pohon Web—suatu tempat yang dapat diunduh orang melalui server Web. Contoh berkas `.htpass` kami ditampilkan dalam Listing 8.8.

Selain menentukan pengguna individual yang berwenang, Anda dapat menentukan bahwa hanya pengguna berwenang yang termasuk dalam kelompok tertentu yang dapat mengakses sumber daya. Kami memilih untuk tidak melakukannya, jadi baris `AuthGroupFile /dev/null` mengatur `AuthGroupFile` kami untuk menunjuk ke `/dev/null`, berkas khusus pada sistem UNIX yang dijamin bernilai `null`.

Seperti contoh PHP, untuk menggunakan autentikasi HTTP, kita perlu memberi nama ranah kita sebagai berikut:

```
AuthName "Realm-Name"
```

Anda dapat memilih nama ranah apa pun yang Anda inginkan, tetapi perlu diingat bahwa nama tersebut akan ditampilkan kepada pengunjung Anda. Agar jelas bahwa nama dalam contoh tersebut harus diubah, nama kita adalah "`Realm-Name`".

Karena sejumlah metode autentikasi yang berbeda didukung, kita perlu menentukan metode autentikasi mana yang kita gunakan. Kita menggunakan autentikasi Dasar sebagaimana ditentukan oleh arahan ini:

```
AuthType Basic
```

Kita perlu menentukan siapa yang diizinkan mengakses. Kita dapat menentukan pengguna tertentu, grup tertentu, atau seperti yang telah kita lakukan, cukup izinkan akses pengguna yang diautentikasi.
Baris

```
require valid-user
```

menentukan bahwa setiap pengguna yang valid akan diizinkan mengakses.

Listing 8.8 .htpass—File Kata Sandi Menyimpan Nama Pengguna dan Kata Sandi Terenkripsi Setiap Pengguna

```
user1:0nRp9M80GS7zM
user2:nC13sOTOhp.ow
user3:yjQMCPWjXFTzU
user4:LOmlMEi/hAme2
```

Setiap baris dalam file `.htpass` berisi nama pengguna, titik dua, dan kata sandi terenkripsi milik pengguna tersebut. Isi pasti dari file `.htpass` Anda akan bervariasi. Untuk membuatnya, Anda menggunakan program kecil bernama `htpasswd` yang disertakan dalam distribusi Apache. Program `htpasswd` digunakan dalam salah satu cara berikut:

```
htpasswd [-cmdps] passwordfile username
```

atau

```
htpasswd -b[cmdps] passwordfile username password
```

Satu-satunya sakelar yang perlu Anda gunakan adalah -c. Menggunakan -c memberi tahu `htpasswd` untuk membuat file. Anda harus menggunakan ini untuk pengguna pertama yang Anda tambahkan. Berhati-hatilah untuk tidak menggunakannya untuk pengguna lain karena jika file tersebut ada, `htpasswd` akan menghapusnya dan membuat yang baru.

Sakelar m, d, p, atau s opsional dapat digunakan jika Anda ingin menentukan algoritme enkripsi mana (termasuk tanpa enkripsi) yang ingin Anda gunakan. Sakelar b memberi tahu program untuk mengharapkan kata sandi sebagai parameter, alih-alih memintanya.

Ini berguna jika Anda ingin memanggil `htpasswd` secara noninteraktif sebagai bagian dari proses batch, tetapi tidak boleh digunakan jika Anda memanggil `htpasswd` dari baris perintah. Perintah berikut membuat file yang ditampilkan dalam Listing 14.8:

```
htpasswd -bc /home/book/.htpass user1 pass1
htpasswd -b /home/book/.htpass user2 pass2
htpasswd -b /home/book/.htpass user4 pass3
htpasswd -b /home/book/.htpass user4 pass4
```

Jenis autentikasi ini mudah diatur, tetapi ada beberapa masalah dengan penggunaan file `.htaccess` dengan cara ini.

Pengguna dan kata sandi disimpan dalam file teks. Setiap kali browser meminta file yang dilindungi oleh file `.htaccess`, server harus mengurai file `.htaccess`, lalu mengurai file kata sandi, mencoba mencocokkan nama pengguna dan kata sandi. Daripada menggunakan file `.htaccess`, kita dapat menentukan hal yang sama dalam file `httpd.conf`—file konfigurasi utama untuk server Web. File `.htaccess` diurai setiap kali file diminta. File `httpd.conf` hanya diurai saat server pertama kali dimulai. Ini akan lebih cepat, tetapi berarti bahwa jika kita ingin membuat perubahan, kita perlu menghentikan dan memulai ulang server.

Terlepas dari tempat kita menyimpan perintah server, file kata sandi tetap perlu dicari untuk setiap permintaan. Artinya, seperti teknik lain yang telah kita bahas yang menggunakan berkas datar, ini tidak akan sesuai untuk ratusan atau ribuan pengguna.

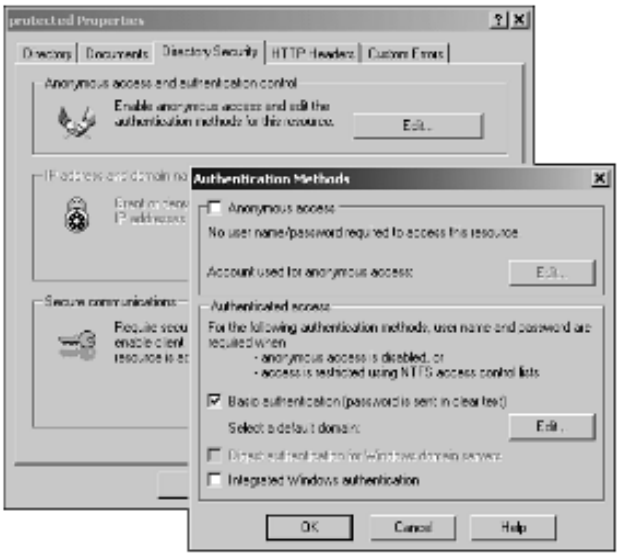
Menggunakan Autentikasi Dasar dengan IIS

Seperti Apache, IIS mendukung autentikasi HTTP. Apache menggunakan pendekatan UNIX dan dikontrol dengan mengedit berkas teks, dan seperti yang Anda duga, memilih opsi dalam kotak dialog mengontrol pengaturan IIS.

Menggunakan Windows 2000, Anda mengubah konfigurasi Internet Information Server 5 (IIS5) menggunakan Internet Services Manager. Anda dapat menemukan utilitas ini dengan memilih Alat Administratif di Panel Kontrol. Internet Services Manager akan terlihat seperti gambar yang ditunjukkan pada Gambar 8.5. Kontrol pohon di sisi kiri menunjukkan bahwa pada mesin bernama windows-server, kita menjalankan sejumlah layanan. Layanan yang kita minati adalah situs Web default. Di dalam situs Web ini, kita memiliki direktori yang disebut protected. Di dalam direktori ini terdapat berkas yang disebut content.html.

Gambar 8.5 Microsoft Management Console memungkinkan kita mengonfigurasi Internet Information Server 5.

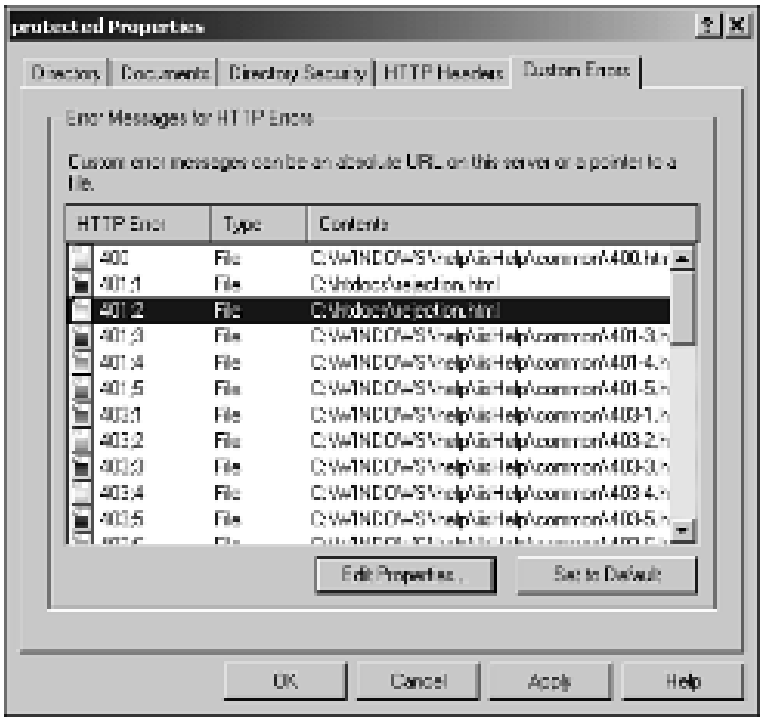
Untuk menambahkan autentikasi dasar ke direktori yang dilindungi, klik kanan padanya dan pilih Properties dari menu konteks. Dialog Properties memungkinkan kita mengubah banyak pengaturan untuk direktori ini. Dua tab yang kita minati adalah Directory Security dan Custom Errors. Salah satu opsi pada tab Directory Security adalah Anonymous Access and Authentication Control. Menekan tombol Edit ini akan memunculkan kotak dialog yang ditunjukkan pada Gambar 8.6.



Gambar 8.6 IIS5 memungkinkan akses anonim secara default, tetapi memungkinkan kita untuk mengaktifkan autentikasi.

Dalam dialog ini, kita dapat menonaktifkan akses anonim dan mengaktifkan autentikasi dasar. Dengan pengaturan yang ditunjukkan pada Gambar 8.6, hanya orang yang memberikan nama dan kata sandi yang sesuai yang dapat melihat file dalam direktori ini. Untuk menduplikasi perilaku contoh sebelumnya, kami juga akan menyediakan halaman untuk memberi tahu pengguna bahwa detail autentikasi mereka tidak benar. Menutup kotak dialog Metode autentikasi akan memungkinkan kami memilih tab Kesalahan Kustom.

Tab Kesalahan Kustom, yang ditunjukkan pada Gambar 8.7, mengaitkan kesalahan dengan pesan kesalahan. Di sini, kami telah menyimpan berkas penolakan yang sama yang kami gunakan sebelumnya, rejection.html, yang ditunjukkan pada Listing 8.6. IIS memberi kami kemampuan untuk memberikan pesan kesalahan yang lebih spesifik daripada yang dilakukan Apache, dengan memberikan kode kesalahan HTTP yang terjadi dan alasan mengapa hal itu terjadi. Untuk kesalahan 401, yang menunjukkan kegagalan autentikasi, IIS memberikan lima alasan berbeda. Kami dapat memberikan pesan yang berbeda untuk masing-masing, tetapi telah memilih untuk hanya mengganti dua pesan yang akan terjadi dalam contoh ini dengan halaman penolakan kami.



Gambar 14.7 Tab Kesalahan Kustom memungkinkan kami mengaitkan halaman kesalahan kustom dengan peristiwa kesalahan.

Hanya itu yang perlu kami lakukan untuk meminta autentikasi untuk direktori ini menggunakan IIS5. Seperti banyak perangkat lunak Windows, perangkat lunak ini lebih mudah diatur daripada perangkat lunak UNIX yang serupa, tetapi lebih sulit disalin dari satu mesin ke mesin lain atau dari satu direktori ke direktori lain. Perangkat lunak ini juga mudah diatur secara tidak sengaja sehingga membuat mesin Anda tidak aman.

Kelemahan utama pendekatan IIS adalah ia mengautentikasi pengguna Web dengan membandingkan detail login mereka dengan akun di mesin tersebut. Jika kita ingin mengizinkan pengguna "john" untuk login dengan kata sandi "password", kita perlu membuat akun pengguna di mesin tersebut, atau di domain, dengan nama dan kata sandi ini. Anda harus sangat berhati-hati saat membuat akun untuk autentikasi Web sehingga pengguna hanya memiliki hak akun yang mereka perlukan untuk melihat halaman Web dan tidak memiliki hak lain seperti akses Telnet.

Menggunakan Autentikasi `mod_auth_mysql`

Seperti yang telah disebutkan, penggunaan `mod_auth` dengan Apache mudah diatur dan efektif. Karena menyimpan pengguna dalam berkas teks, hal ini tidak terlalu praktis untuk situs yang sibuk dengan banyak pengguna.

Untungnya, Anda dapat menikmati sebagian besar kemudahan `mod_auth`, dan kecepatan basis data menggunakan `mod_auth_mysql`. Modul ini bekerja dengan cara yang hampir sama seperti `mod_auth`, tetapi karena menggunakan basis data MySQL, bukan berkas teks, modul ini dapat mencari daftar pengguna yang besar dengan cepat.

Untuk menggunakannya, Anda perlu mengompilasi dan memasang modul di sistem Anda atau meminta administrator sistem untuk memasangnya.

Memasang `mod_auth_mysql`

Untuk menggunakan `mod_auth_mysql`, Anda perlu menyiapkan Apache dan MySQL, tetapi berikut ini ringkasannya.

1. Dapatkan arsip distribusi untuk modul tersebut.
   http://www.zend.com atau sebagai alternatif
   http://www.mysql.com/downloads/contrib.html
2. Ekstrak zip dan untar kode sumber.
3. Ubah ke direktori mod_auth_mysql dan jalankan configure. Anda perlu memberi tahu tempat untuk menemukan instalasi MySQL dan kode sumber Apache Anda. Agar sesuai dengan struktur direktori pada komputer saya, saya mengetik

   ```
   ./configure --with-mysql=/var/mysql --with-apache=/src/apache_1.3.12
   ```

   tetapi lokasi Anda mungkin berbeda.
4. Jalankan make, lalu make install. Anda perlu menambahkan `--activate-module=src/modules/auth_mysql/libauth_mysql.a` ke parameter yang Anda berikan untuk configure saat Anda mengonfigurasi Apache. Untuk pengaturan pada sistem saya, saya menggunakan

   ```
   ./configure --enable-module=ssl \
   --activate-module=src/modules/php4/libphp4.a \
   --enable-module=php4 --prefix=/usr/local/apache --enable-shared=ssl \
   --activate-module=src/modules/auth_mysql/libauth_mysql.a
   ```
5. Setelah mengikuti langkah-langkah, Anda perlu membuat basis data dan tabel di MySQL untuk memuat informasi autentikasi. Ini tidak perlu berupa basis data atau

tabel terpisah; Anda dapat menggunakan tabel yang sudah ada seperti basis data auth dari contoh sebelumnya dalam bab ini.

6. Tambahkan baris ke berkas httpd.conf Anda untuk memberi `mod_auth_mysql` parameter yang dibutuhkannya untuk terhubung ke MySQL. Perintah tersebut akan terlihat seperti
   `Auth_MySQL_Info hostname user password`

Berhasilkah?

Cara termudah untuk memeriksa apakah kompilasi Anda berhasil adalah dengan melihat apakah Apache akan mulai. Untuk memulai Apache, ketik

```
/usr/local/apache/bin/apachectl startssl
```

Jika dimulai dengan perintah Auth_MySQL_Info dalam berkas httpd.conf, mod_auth_mysql berhasil ditambahkan.

Menggunakan `mod_auth_mysql`

Setelah Anda berhasil menginstal modul, menggunakannya tidak lebih sulit daripada menggunakan `mod_auth`. Listing 8.9 menunjukkan contoh file `.htaccess` yang akan mengautentikasi pengguna dengan kata sandi terenkripsi yang tersimpan dalam basis data yang dibuat sebelumnya dalam bab ini.

Listing 8.9 .htaccess—File .htaccess Ini Mengautentikasi Pengguna terhadap Basis Data MySQL

```
ErrorDocument 401 /chapter14/rejection.html

AuthName “Realm Name”
AuthType Basic

Auth_MySQL_DB auth
Auth_MySQL_Encryption_Types MySQL
Auth_MySQL_Password_Table auth
Auth_MySQL_Username_Field name
Auth_MySQL_Password_Field pass

require valid-user
```

Anda dapat melihat bahwa sebagian besar Listing 8.9 sama dengan Listing 8.7. Kami masih menentukan dokumen kesalahan untuk ditampilkan jika terjadi kesalahan 401 (ketika autentikasi gagal). Kami kembali menentukan autentikasi dasar dan memberikan nama wilayah. Seperti dalam Listing 8..7, kami akan mengizinkan akses pengguna yang sah dan terautentikasi.

Karena kami menggunakan `mod_auth_mysql` dan tidak ingin menggunakan semua pengaturan default, kami memiliki beberapa arahan untuk menentukan cara kerjanya. `Auth_MySQL_DB`, `Auth_MySQL_Password_Table`, `Auth_MySQL_Username_Field`, dan

`Auth_MySQL_Password_Field` menentukan nama basis data, tabel, kolom nama pengguna, dan kolom kata sandi.

Kami menyertakan arahan `Auth_MySQL_Encryption_Types` untuk menentukan bahwa kami ingin menggunakan enkripsi kata sandi MySQL. Nilai yang dapat diterima adalah Plaintext, Crypt_DES, atau MySQL. Crypt_DES adalah default, dan menggunakan kata sandi terenkripsi UNIX DES standar.

Dari sudut pandang pengguna, contoh `mod_auth_mysql` ini akan bekerja dengan cara yang persis sama seperti contoh `mod_auth.` Pengguna akan disajikan dengan kotak dialog oleh peramban Web-nya. Jika berhasil mengautentikasi, pengguna akan diperlihatkan kontennya. Jika gagal, pengguna akan diberikan halaman kesalahan.

Untuk banyak situs Web, `mod_auth_mysql` adalah yang ideal. Cepat, relatif mudah diimplementasikan, dan memungkinkan Anda menggunakan mekanisme yang mudah digunakan untuk menambahkan entri basis data bagi pengguna baru. Untuk fleksibilitas yang lebih baik, dan kemampuan untuk menerapkan kontrol yang lebih rinci pada bagian halaman, Anda mungkin ingin menerapkan autentikasi Anda sendiri menggunakan PHP dan MySQL.

BAB 9
MENERAPKAN TRANSAKSI AMAN DENGAN PHP DAN MYSQL

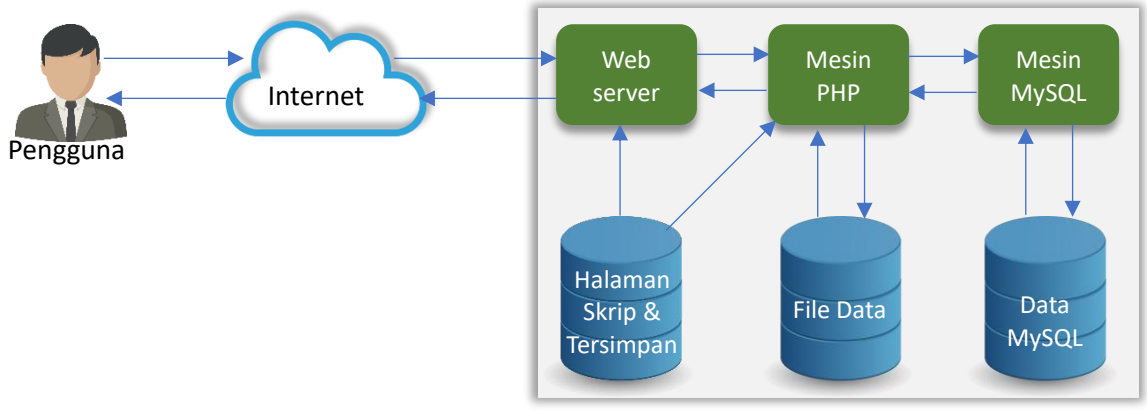
Dalam bab ini, kami akan menjelaskan cara menangani data pengguna dengan aman dari input, melalui transmisi, dan penyimpanan. Ini akan memungkinkan kami untuk menerapkan transaksi antara kami dan pengguna dengan aman dari awal hingga akhir. Topik meliputi

- Menyediakan transaksi aman
- Menggunakan Secure Sockets Layer (SSL)
- Menyediakan penyimpanan aman
- Mengapa Anda menyimpan nomor kartu kredit?
- Menggunakan enkripsi dalam PHP

9.1 MENYEDIAKAN TRANSAKSI AMAN

Menyediakan transaksi aman menggunakan Internet adalah masalah memeriksa aliran informasi dalam sistem Anda dan memastikan bahwa di setiap titik, informasi Anda aman. Dalam konteks keamanan jaringan, tidak ada yang mutlak. Tidak ada sistem yang tidak akan pernah bisa ditembus. Yang kami maksud dengan aman adalah tingkat upaya yang diperlukan untuk membahayakan sistem atau transmisi tinggi dibandingkan dengan nilai informasi yang terlibat.

Jika kita ingin mengarahkan upaya keamanan kita secara efektif, kita perlu memeriksa aliran informasi melalui semua bagian sistem kita. Alur informasi pengguna dalam aplikasi umum, yang ditulis menggunakan PHP dan MySQL, ditunjukkan pada Gambar 9.1.



Gambar 9.1 Informasi pengguna disimpan atau diproses oleh elemen-elemen berikut dari lingkungan aplikasi Web umum.

Rincian setiap transaksi yang terjadi di sistem Anda akan bervariasi, bergantung pada desain sistem Anda dan pada data dan tindakan pengguna yang memicu transaksi. Anda dapat memeriksa semua ini dengan cara yang sama. Setiap transaksi antara aplikasi Web dan

pengguna dimulai dengan browser pengguna yang mengirimkan permintaan melalui Internet ke server Web. Jika halaman tersebut adalah skrip PHP, server Web akan mendelegasikan pemrosesan halaman tersebut ke mesin PHP.

Skrip PHP dapat membaca atau menulis data ke disk. Skrip tersebut juga dapat menyertakan() atau memerlukan() file PHP atau HTML lainnya. Skrip tersebut juga akan mengirimkan kueri SQL ke daemon MySQL dan menerima respons. Mesin MySQL bertanggung jawab untuk membaca dan menulis datanya sendiri di disk. Sistem ini memiliki tiga bagian utama:

- Mesin pengguna
- Internet
- Sistem Anda

Kita akan membahas pertimbangan keamanan untuk masing-masing secara terpisah, tetapi jelas bahwa mesin pengguna dan Internet sebagian besar berada di luar kendali Anda.

Mesin Pengguna

Dari sudut pandang kami, mesin pengguna menjalankan peramban Web. Kami tidak memiliki kendali atas faktor-faktor lain seperti seberapa aman mesin tersebut disiapkan. Kita perlu mengingat bahwa mesin tersebut mungkin sangat tidak aman atau bahkan merupakan terminal bersama di perpustakaan, sekolah, atau kafe.

Banyak peramban yang tersedia, masing-masing memiliki kemampuan yang sedikit berbeda. Jika kita hanya mempertimbangkan versi terbaru dari dua peramban paling populer, sebagian besar perbedaan di antara keduanya hanya memengaruhi cara HTML akan dirender dan ditampilkan, tetapi ada masalah keamanan atau fungsionalitas yang perlu kita pertimbangkan.

Anda harus memperhatikan bahwa beberapa orang akan menonaktifkan fitur yang mereka anggap sebagai risiko keamanan atau privasi, seperti Java, cookie, atau JavaScript. Jika Anda menggunakan fitur-fitur ini, Anda harus menguji apakah aplikasi Anda dapat berjalan dengan baik bagi orang-orang yang tidak memiliki fitur-fitur ini, atau mempertimbangkan untuk menyediakan antarmuka yang kurang kaya fitur yang memungkinkan orang-orang ini menggunakan situs Anda.

Pengguna di luar Amerika Serikat dan Kanada mungkin memiliki peramban web yang hanya mendukung enkripsi 40-bit. Meskipun Pemerintah AS mengubah undang-undang pada bulan Januari 2000 untuk mengizinkan ekspor enkripsi yang kuat (ke negara-negara yang tidak diembargo) dan versi 128-bit sekarang tersedia bagi sebagian besar pengguna, beberapa di antaranya tidak akan ditingkatkan. Kecuali Anda memberikan jaminan keamanan kepada pengguna dalam teks situs Anda, hal ini tidak perlu terlalu menjadi perhatian Anda sebagai pengembang web. SSL akan secara otomatis bernegosiasi agar Anda mengaktifkan server Anda dan peramban pengguna untuk berkomunikasi pada tingkat paling aman yang dapat dipahami oleh keduanya.

Kami tidak dapat memastikan bahwa kami berurusan dengan peramban web yang terhubung ke situs kami melalui antarmuka yang kami maksud. Permintaan ke situs kami

mungkin berasal dari situs lain yang mencuri gambar atau konten, atau dari seseorang yang menggunakan perangkat lunak seperti cURL untuk melewati langkah-langkah keamanan.

Kita akan melihat pustaka cURL, yang dapat digunakan untuk mensimulasikan koneksi dari browser. Meskipun kita tidak dapat mengubah atau mengendalikan cara pengaturan mesin pengguna kita, kita perlu mengingatnya. Variabilitas mesin pengguna mungkin menjadi faktor dalam seberapa banyak fungsionalitas yang kita sediakan melalui skrip sisi server (seperti PHP) dan seberapa banyak yang kita sediakan melalui skrip sisi klien (seperti JavaScript).

Fungsionalitas yang disediakan oleh PHP dapat kompatibel dengan setiap browser pengguna, karena hasil akhirnya hanyalah halaman HTML. Menggunakan JavaScript apa pun kecuali yang sangat mendasar akan melibatkan pertimbangan kemampuan yang berbeda dari masing-masing versi browser.

Dari perspektif keamanan, kita lebih baik menggunakan skrip sisi server untuk hal-hal seperti validasi data karena, dengan cara itu, kode sumber kita tidak akan terlihat oleh pengguna. Jika kita memvalidasi data dalam JavaScript, pengguna akan dapat melihat kode tersebut dan mungkin menghindarinya. Data yang perlu disimpan dapat disimpan di komputer kita sendiri, sebagai file atau catatan basis data, atau di komputer pengguna kita sebagai cookie.

Sebagian besar data yang kita simpan harus berada di server Web, atau di basis data kita. Ada sejumlah alasan bagus untuk menyimpan informasi sesedikit mungkin di komputer pengguna. Jika informasi tersebut berada di luar sistem Anda, Anda tidak memiliki kendali atas seberapa aman informasi tersebut disimpan, Anda tidak dapat memastikan bahwa pengguna tidak akan menghapusnya, dan Anda tidak dapat menghentikan pengguna untuk mengubahnya dalam upaya membingungkan sistem Anda.

9.2 INTERNET

Seperti komputer pengguna, Anda memiliki kendali yang sangat kecil atas karakteristik Internet, tetapi, seperti komputer pengguna, ini tidak berarti bahwa Anda dapat mengabaikan karakteristik ini saat merancang sistem Anda.

Internet memiliki banyak fitur yang bagus, tetapi pada dasarnya merupakan jaringan yang tidak aman. Saat mengirim informasi dari satu titik ke titik lain, Anda perlu mengingat bahwa orang lain dapat melihat atau mengubah informasi yang Anda kirimkan. Dengan mengingat hal ini, Anda dapat memutuskan tindakan apa yang harus diambil.
Respons Anda mungkin adalah:

- Tetap kirimkan informasi tersebut, meskipun Anda tahu bahwa informasi tersebut mungkin tidak bersifat pribadi.
- Enkripsi atau tanda tangani informasi tersebut sebelum mengirimkannya untuk menjaga kerahasiaannya atau melindunginya dari gangguan.
- Putuskan bahwa informasi Anda terlalu sensitif untuk berisiko disadap dan cari cara lain untuk mendistribusikan informasi Anda.

Internet juga merupakan media yang cukup anonim. Sulit untuk memastikan apakah orang yang Anda hadapi adalah orang yang mereka klaim. Bahkan jika Anda dapat meyakinkan diri sendiri tentang pengguna sesuai keinginan Anda, mungkin sulit untuk membuktikannya melampaui tingkat keraguan yang cukup di forum seperti pengadilan. Hal ini menyebabkan masalah dengan penolakan. Singkatnya, privasi dan penolakan merupakan masalah besar saat melakukan transaksi melalui Internet.

Setidaknya ada dua cara berbeda untuk mengamankan informasi yang mengalir ke dan dari server Web Anda melalui Internet:

- SSL (*Secure Sockets Layer*)
- S-HTTP (*Secure Hypertext Transfer Protocol*)

Kedua teknologi ini menawarkan pesan dan autentikasi yang bersifat pribadi dan tahan gangguan, tetapi SSL tersedia dan digunakan secara luas sedangkan S-HTTP belum benar-benar populer. Kita akan membahas SSL secara terperinci nanti di bab ini.

Sistem Anda

Bagian alam semesta yang dapat Anda kendalikan adalah sistem Anda. Sistem Anda diwakili oleh komponen-komponen dalam garis putus-putus seperti yang ditunjukkan sebelumnya pada Gambar 9.1. Komponen-komponen ini mungkin terpisah secara fisik pada suatu jaringan, atau semuanya ada pada satu mesin fisik.

Cukup aman untuk tidak mengkhawatirkan keamanan informasi sementara berbagai produk pihak ketiga yang kami gunakan untuk menyampaikan konten Web kami menanganinya. Para pembuat perangkat lunak tertentu mungkin telah memikirkannya lebih matang daripada Anda.

Selama Anda menggunakan versi terbaru dari suatu produk yang terkenal, Anda akan dapat menemukan masalah-masalah yang terkenal dengan menggunakan mesin pencari Web favorit Anda. Anda harus memprioritaskan untuk selalu memperbarui informasi ini. Jika instalasi dan konfigurasi merupakan bagian dari peran Anda, Anda perlu mengkhawatirkan cara perangkat lunak diinstal dan dikonfigurasi. Banyak kesalahan yang dibuat dalam keamanan merupakan akibat dari tidak mengikuti peringatan dalam dokumentasi, atau melibatkan masalah administrasi sistem umum yang merupakan topik untuk buku lain. Belilah buku yang bagus tentang pengelolaan sistem operasi yang ingin Anda gunakan, atau pekerjakan seorang administrator sistem yang ahli.

Satu hal khusus yang perlu dipertimbangkan saat menginstal PHP adalah bahwa secara umum lebih aman, serta jauh lebih efisien, untuk menginstal PHP sebagai modul SAPI untuk server Web Anda daripada menjalankannya melalui antarmuka CGI.

Hal utama yang perlu Anda khawatirkan adalah apa yang dilakukan atau tidak dilakukan oleh skrip Anda sendiri. Data sensitif apa yang mungkin dikirimkan aplikasi kita kepada pengguna melalui Internet?

Data sensitif apa yang kita minta agar dikirimkan pengguna kepada kita? Jika kita mengirimkan informasi yang seharusnya merupakan transaksi pribadi antara kita dan pengguna kita atau yang seharusnya sulit diubah oleh perantara, kita harus mempertimbangkan untuk menggunakan SSL.

Kita telah membahas tentang penggunaan SSL antara komputer pengguna dan server. Anda juga harus memikirkan situasi saat Anda mengirimkan data dari satu komponen sistem Anda ke komponen lain melalui jaringan. Contoh umum muncul saat basis data MySQL Anda berada di mesin yang berbeda dari server Web Anda. PHP akan terhubung ke server MySQL Anda melalui TCP/IP, dan koneksi ini tidak akan dienkripsi. Jika kedua mesin ini berada di jaringan area lokal pribadi, Anda perlu memastikan bahwa jaringan tersebut aman. Jika mesin berkomunikasi melalui Internet, sistem Anda mungkin akan berjalan lambat, dan Anda perlu memperlakukan koneksi ini dengan cara yang sama seperti koneksi lain melalui Internet.

PHP tidak memiliki cara asli untuk membuat koneksi ini melalui SSL. Perintah `fopen()` mendukung HTTP tetapi tidak HTTPS. Namun, Anda dapat menggunakan SSL melalui pustaka cURL.

Penting bahwa ketika pengguna kita mengira mereka berurusan dengan kita, mereka memang berurusan dengan kita. Mendaftar untuk sertifikat digital akan melindungi pengunjung kita dari spoofing (orang lain meniru situs kita), memungkinkan kita menggunakan SSL tanpa pengguna melihat pesan peringatan, dan memberikan kesan terhormat pada usaha daring kita.

Apakah skrip kita dengan hati-hati memeriksa data yang dimasukkan pengguna?
Apakah kita berhati-hati dalam menyimpan informasi dengan aman?

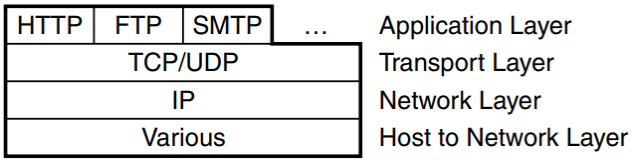
Kita akan menjawab pertanyaan-pertanyaan ini di beberapa bagian berikutnya dari bab ini.

9.3 MENGGUNAKAN SECURE SOCKETS LAYER (SSL)

Rangkaian protokol Secure Sockets Layer awalnya dirancang oleh Netscape untuk memfasilitasi komunikasi yang aman antara server Web dan peramban Web. Sejak saat itu, protokol ini telah diadopsi sebagai metode standar tidak resmi bagi peramban dan server untuk bertukar informasi sensitif.

Baik SSL versi 2 maupun versi 3 didukung dengan baik. Sebagian besar server Web menyertakan fungsionalitas SSL, atau dapat menerimanya sebagai modul tambahan. Internet Explorer dan Netscape Navigator telah mendukung SSL sejak versi 3.

Protokol jaringan dan perangkat lunak yang mengimplementasikannya biasanya disusun sebagai tumpukan lapisan. Setiap lapisan dapat meneruskan data ke lapisan di atas atau di bawahnya, dan meminta layanan dari lapisan di atas atau di bawahnya. Gambar 9.2 menunjukkan tumpukan protokol tersebut.

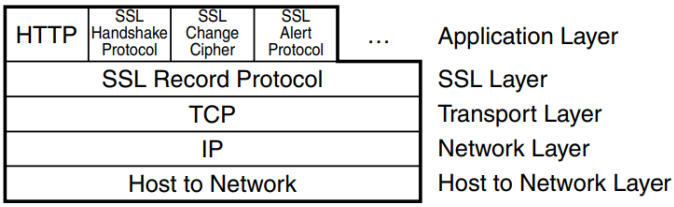


Gambar 9.2 Tumpukan protokol yang digunakan oleh protokol lapisan aplikasi seperti Hypertext Transfer Protocol.

Saat Anda menggunakan HTTP untuk mentransfer informasi, protokol HTTP memanggil *Transmission Control Protocol* (TCP), yang selanjutnya bergantung pada *Internet Protocol* (IP). Protokol ini selanjutnya memerlukan protokol yang sesuai untuk perangkat keras jaringan yang digunakan untuk mengambil paket data dan mengirimkannya sebagai sinyal listrik ke tujuan kita.

HTTP disebut protokol lapisan aplikasi. Ada banyak protokol lapisan aplikasi lainnya seperti FTP, SMTP dan telnet (seperti yang ditunjukkan pada gambar), dan lainnya seperti POP dan IMAP. TCP adalah salah satu dari dua protokol lapisan transport yang digunakan dalam jaringan TCP/IP. IP adalah protokol pada lapisan jaringan. Lapisan host ke jaringan bertanggung jawab untuk menghubungkan host (komputer) kita ke jaringan. Tumpukan protokol TCP/IP tidak menentukan protokol yang digunakan untuk lapisan ini, karena kita memerlukan protokol yang berbeda untuk berbagai jenis jaringan.

Saat mengirim data, data dikirim melalui tumpukan dari aplikasi ke media jaringan fisik. Saat menerima data, data bergerak dari jaringan fisik, melalui tumpukan, ke aplikasi. Penggunaan SSL menambahkan lapisan transparan tambahan ke model ini. Lapisan SSL berada di antara lapisan transport dan lapisan aplikasi. Hal ini ditunjukkan pada Gambar 9.3. Lapisan SSL memodifikasi data dari aplikasi HTTP kita sebelum memberikannya ke lapisan transport untuk mengirimkannya ke tujuannya.



Gambar 9.3 SSL menambahkan lapisan tambahan ke tumpukan protokol serta protokol lapisan aplikasi untuk mengendalikan operasinya sendiri.

Secara teori SSL mampu menyediakan lingkungan transmisi yang aman untuk protokol selain HTTP, tetapi biasanya hanya digunakan untuk HTTP. Protokol lain dapat digunakan karena lapisan SSL pada dasarnya transparan. Lapisan SSL menyediakan antarmuka yang sama ke protokol di atasnya seperti lapisan transport yang mendasarinya. Kemudian, lapisan ini secara transparan menangani jabat tangan, enkripsi, dan dekripsi.

Saat peramban web terhubung ke server web yang aman melalui HTTP, keduanya perlu mengikuti protokol jabat tangan untuk menyetujui hal-hal seperti autentikasi dan enkripsi. Urutan jabat tangan melibatkan langkah-langkah berikut:

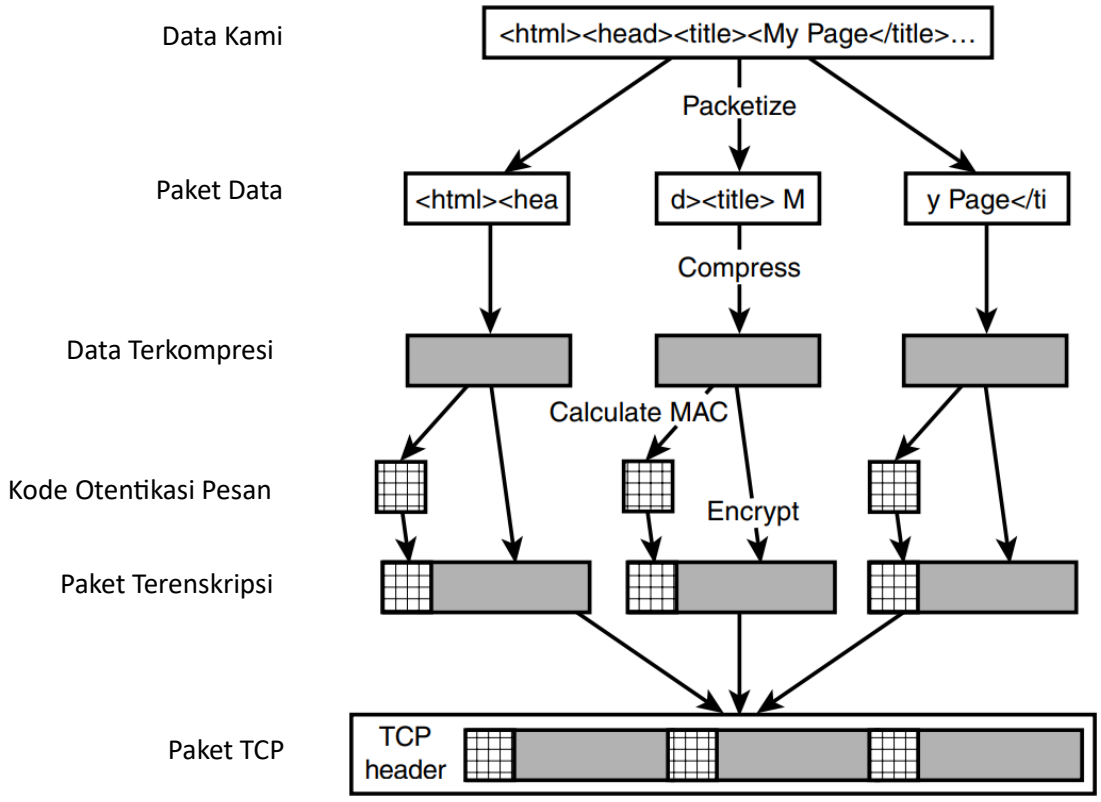
1. Peramban terhubung ke server yang mendukung SSL dan meminta server untuk mengautentikasi dirinya sendiri.
2. Server mengirimkan sertifikat digitalnya.
3. Server mungkin secara opsional (dan jarang) meminta peramban untuk mengautentikasi dirinya sendiri.
4. Peramban menyajikan daftar algoritma enkripsi dan fungsi hash yang didukungnya. Server memilih enkripsi terkuat yang juga didukungnya.

5. Peramban dan server membuat kunci sesi:
    - Peramban memperoleh kunci publik server dari sertifikat digitalnya dan menggunakannya untuk mengenkripsi nomor yang dibuat secara acak.
    - Server merespons dengan lebih banyak data acak yang dikirim dalam bentuk teks biasa (kecuali jika peramban telah memberikan sertifikat digital atas permintaan server, dalam hal ini server akan menggunakan kunci publik peramban).
    - Kunci enkripsi untuk sesi dibuat dari data acak ini menggunakan fungsi hash.

Membuat data acak berkualitas baik, mendekripsi sertifikat digital, dan membuat kunci serta menggunakan kriptografi kunci publik memerlukan waktu, jadi prosedur jabat tangan ini memerlukan waktu. Untungnya, hasilnya di-cache, jadi jika peramban dan server yang sama ingin bertukar beberapa pesan aman, proses jabat tangan dan waktu pemrosesan yang diperlukan hanya terjadi satu kali.

Saat data dikirim melalui koneksi SSL, langkah-langkah berikut terjadi:

1. Data dipecah menjadi paket-paket yang dapat dikelola.
2. Setiap paket (opsional) dikompresi.
3. Setiap paket memiliki kode autentikasi pesan (MAC) yang dihitung menggunakan algoritma hashing.
4. MAC dan data terkompresi digabungkan dan dienkripsi.
5. Paket terenkripsi digabungkan dengan informasi header dan dikirim ke jaringan. Seluruh proses ditunjukkan pada Gambar 9.4.



Gambar 9.4 SSL memecah, mengompresi, membuat hash, dan mengenkripsi data sebelum mengirimkannya.

Satu hal yang mungkin Anda perhatikan dari diagram adalah bahwa header TCP ditambahkan setelah data dienkripsi. Ini berarti bahwa informasi perutean masih berpotensi dirusak, dan meskipun pengintai tidak dapat mengetahui informasi apa yang kita tukarkan, mereka dapat melihat siapa yang menukarnya.

Alasan mengapa SSL menyertakan kompresi sebelum enkripsi adalah karena meskipun sebagian besar lalu lintas jaringan dapat (dan sering kali) dikompresi sebelum ditransmisikan melalui jaringan, data terenkripsi tidak terkompresi dengan baik.

Skema kompresi bergantung pada identifikasi pengulangan atau pola dalam data. Mencoba menerapkan algoritme kompresi setelah data diubah menjadi susunan bit yang acak secara efektif melalui enkripsi biasanya tidak ada gunanya. Akan sangat disayangkan jika SSL, yang dirancang untuk meningkatkan keamanan jaringan, memiliki efek samping berupa peningkatan lalu lintas jaringan secara drastis.

Meskipun SSL relatif rumit, pengguna dan pengembang terlindungi dari sebagian besar hal yang terjadi, karena antarmuka eksternalnya meniru protokol yang ada. Dalam waktu dekat, SSL 3.0 kemungkinan akan digantikan oleh TLS 1.0 (*Transport Layer Security*), tetapi pada saat artikel ini ditulis, TLS masih berupa rancangan standar dan tidak didukung oleh server atau browser mana pun. TLS dimaksudkan untuk menjadi standar yang benar-benar terbuka, bukan standar yang ditetapkan oleh satu organisasi tetapi disediakan untuk organisasi lain. TLS didasarkan langsung pada SSL 3.0, tetapi berisi penyempurnaan yang dimaksudkan untuk mengatasi kelemahan SSL.

Menyaring Masukan Pengguna

Salah satu prinsip membangun aplikasi Web yang aman adalah Anda tidak boleh memercayai masukan pengguna. Selalu saring data pengguna sebelum memasukkannya ke dalam file atau basis data atau meneruskannya melalui perintah eksekusi sistem.

Kami telah membahas beberapa teknik yang dapat Anda gunakan untuk menyaring masukan pengguna di seluruh buku ini. Kami akan mencantumkannya secara singkat di sini sebagai referensi.

- Fungsi `addslashes()` harus digunakan untuk menyaring data pengguna sebelum diteruskan ke basis data. Fungsi ini akan menghilangkan karakter yang mungkin mengganggu basis data. Anda dapat menggunakan fungsi `stripslashes()` untuk mengembalikan data ke bentuk aslinya.
- `magic_quotes_gpc`. Anda dapat mengaktifkan perintah `magic_quotes_gpc` dan `magic_quotes_runtime` di file php.ini Anda. Perintah ini akan secara otomatis menambahkan dan menghapus garis miring untuk Anda. `magic_quotes_gpc` akan menerapkan format ini ke variabel GET, POST, dan cookie yang masuk, dan `magic_quote_runtime` akan menerapkannya ke data yang masuk dan keluar dari basis data.
- Fungsi `escapeshellcmd()` harus digunakan saat Anda meneruskan data pengguna ke panggilan `system()` atau `exec()` atau ke backticks. Fungsi ini akan menghilangkan semua metakarakter yang dapat digunakan untuk memaksa sistem Anda menjalankan perintah sembarangan yang dimasukkan oleh pengguna jahat.

- Anda dapat menggunakan fungsi `strip_tags()` untuk menghilangkan tag HTML dan PHP dari string. Fungsi ini akan mencegah pengguna menanam skrip jahat dalam data pengguna yang mungkin Anda gema kembali ke browser.
- Anda dapat menggunakan fungsi `htmlspecialchars()`, yang akan mengubah karakter menjadi padanan entitas HTML-nya. Misalnya, < akan diubah menjadi <. Fungsi ini akan mengubah semua tag skrip menjadi karakter yang tidak berbahaya.

9.4 MENYEDIAKAN PENYIMPANAN YANG AMAN

Tiga jenis data yang disimpan (file HTML atau PHP, data terkait skrip, dan data MySQL) akan sering disimpan di area yang berbeda pada disk yang sama, tetapi ditampilkan secara terpisah pada Gambar 9.1. Setiap jenis penyimpanan memerlukan tindakan pencegahan yang berbeda dan akan diperiksa secara terpisah.

Jenis data paling berbahaya yang kami simpan adalah konten yang dapat dieksekusi. Di situs Web, ini biasanya berarti skrip. Anda harus sangat berhati-hati agar izin berkas Anda ditetapkan dengan benar dalam hierarki Web Anda. Yang kami maksud dengan ini adalah pohon direktori yang dimulai dari htdocs pada server Apache atau inetpub pada server IIS. Orang lain perlu memiliki izin untuk membaca skrip Anda agar dapat melihat output mereka, tetapi mereka tidak boleh menulis ulang atau mengeditnya.

Ketentuan yang sama berlaku untuk direktori dalam hierarki Web. Hanya kita yang boleh menulis ke direktori ini. Pengguna lain, termasuk pengguna yang menjalankan server Web, tidak boleh memiliki izin untuk menulis atau membuat berkas baru di direktori yang dapat dimuat dari server Web. Jika Anda mengizinkan orang lain menulis berkas di sini, mereka dapat menulis skrip berbahaya dan menjalankannya dengan memuatnya melalui server Web.

Jika skrip Anda memerlukan izin untuk menulis ke berkas, buat direktori di luar pohon Web untuk tujuan ini. Hal ini khususnya berlaku untuk skrip unggah berkas. Skrip dan data yang ditulisnya tidak boleh dicampur.

Saat menulis data sensitif, Anda mungkin tergoda untuk mengenkripsinya terlebih dahulu. Namun, pendekatan ini biasanya tidak terlalu berguna. Kita akan menjelaskannya seperti ini: Jika Anda memiliki berkas bernama `creditcardnumbers.txt` di server Web Anda dan seorang peretas memperoleh akses ke server Anda dan dapat membacanya, apa lagi yang dapat dibacanya? Untuk mengenkripsi dan mendekripsi data, Anda memerlukan program untuk mengenkripsi data, program untuk mendekripsi data, dan satu atau beberapa berkas kunci. Jika peretas dapat membaca data Anda, mungkin tidak ada yang dapat menghentikannya untuk membaca kunci dan berkas lainnya.

Mengenkripsi data dapat berguna di server Web, tetapi hanya jika perangkat lunak dan kunci untuk mendekripsi data tidak disimpan di server Web, tetapi hanya ada di komputer lain. Salah satu cara untuk menangani data sensitif dengan aman adalah dengan mengenkripsinya di server, lalu mengirimkannya ke komputer lain, mungkin melalui email.

Data basis data mirip dengan berkas data. Jika Anda mengatur MySQL dengan benar, hanya MySQL yang dapat menulis ke berkas datanya. Ini berarti bahwa kita hanya perlu khawatir tentang akses dari pengguna di dalam MySQL. Kita telah membahas sistem izin

MySQL sendiri, yang menetapkan hak-hak tertentu untuk nama pengguna tertentu di host tertentu.

Satu hal yang perlu disebutkan secara khusus adalah bahwa Anda sering kali perlu menulis kata sandi MySQL dalam skrip PHP. Skrip PHP Anda umumnya dapat dimuat secara publik. Ini tidak seburuk yang mungkin terlihat pada awalnya. Kecuali konfigurasi server Web Anda rusak, sumber PHP Anda tidak akan terlihat dari luar.

Jika server Web Anda dikonfigurasi untuk mengurai file dengan ekstensi .php menggunakan interpreter PHP, orang luar tidak akan dapat melihat sumber yang tidak ditafsirkan. Namun, Anda harus berhati-hati saat menggunakan ekstensi lain. Jika Anda menempatkan file .inc di direktori Web Anda, siapa pun yang memintanya akan menerima sumber yang tidak diurai. Anda perlu menempatkan file include di luar pohon Web, mengonfigurasi server Anda untuk tidak mengirimkan file dengan ekstensi ini, atau menggunakan .php sebagai ekstensi pada direktori ini juga.

Jika Anda berbagi server Web dengan orang lain, kata sandi MySQL Anda mungkin terlihat oleh pengguna lain di komputer yang sama yang juga dapat menjalankan skrip melalui server Web yang sama. Bergantung pada cara sistem Anda diatur, hal ini mungkin tidak dapat dihindari. Hal ini dapat dihindari dengan menyiapkan server Web untuk menjalankan skrip sebagai pengguna individual, atau dengan meminta setiap pengguna menjalankan instans server Web miliknya sendiri. Jika Anda bukan administrator untuk server Web Anda (seperti yang mungkin terjadi jika Anda berbagi server), mungkin ada baiknya mendiskusikan hal ini dengan administrator Anda dan mengeksplorasi opsi keamanan.

Mengapa Anda Menyimpan Nomor Kartu Kredit?

Setelah membahas penyimpanan aman untuk data sensitif, satu jenis data sensitif perlu disebutkan secara khusus. Pengguna internet paranoid tentang nomor kartu kredit mereka. Jika Anda akan menyimpannya, Anda harus sangat berhati-hati. Anda juga perlu bertanya pada diri sendiri mengapa Anda melakukannya, dan apakah itu benar-benar diperlukan.

Apa yang akan Anda lakukan dengan nomor kartu? Jika Anda memiliki transaksi satu kali untuk diproses dan pemrosesan kartu secara real-time, Anda akan lebih baik menerima nomor kartu dari pelanggan Anda dan mengirimkannya langsung ke gerbang pemrosesan transaksi Anda tanpa menyimpannya sama sekali.

Jika Anda memiliki tagihan berkala yang harus dibuat, seperti kewenangan untuk membebankan biaya bulanan ke kartu yang sama untuk langganan yang sedang berlangsung, ini mungkin bukan pilihan. Dalam kasus ini, Anda harus mempertimbangkan untuk menyimpan nomor tersebut di tempat lain selain server Web.

Jika Anda akan menyimpan sejumlah besar detail kartu pelanggan Anda, pastikan Anda memiliki administrator sistem yang terampil dan agak paranoid yang memiliki cukup waktu untuk memeriksa sumber informasi keamanan terkini untuk sistem operasi dan produk lain yang Anda gunakan.

Menggunakan Enkripsi dalam PHP

Tugas sederhana, tetapi bermanfaat, yang dapat kita gunakan untuk menunjukkan enkripsi adalah mengirim email terenkripsi. Standar de facto untuk email terenkripsi selama bertahun-tahun adalah PGP, yang merupakan singkatan dari Pretty Good Privacy. Philip R. Zimmermann menulis PGP secara khusus untuk menambahkan privasi ke email.

Versi perangkat lunak gratis PGP tersedia, tetapi Anda harus memperhatikan bahwa ini bukan Perangkat Lunak Bebas. Versi perangkat lunak gratis hanya dapat digunakan secara legal untuk penggunaan nonkomersial.

Anda dapat menggunakan kedua produk tersebut bersama-sama, membuat pesan terenkripsi menggunakan GPG untuk seseorang yang menggunakan PGP (selama itu adalah versi terbaru) untuk mendekripsi. Karena pembuatan pesan di server Web yang kami minati, kami akan memberikan contoh di sini menggunakan GPG. Menggunakan PGP tidak akan memerlukan banyak perubahan.

Selain persyaratan umum untuk contoh dalam buku ini, Anda perlu memiliki GPG agar kode ini berfungsi. GPG mungkin sudah terinstal di sistem Anda. Jika belum, jangan khawatir: Prosedur instalasinya sangat mudah, tetapi pengaturannya bisa sedikit rumit.

Menginstal GPG

Untuk menambahkan GPG ke mesin Linux kami, kami mengunduh berkas arsip yang sesuai dari www.gnupg.org, dan menggunakan gunzip dan tar untuk mengekstrak berkas dari arsip.

Untuk mengompilasi dan menginstal program, gunakan perintah yang sama seperti untuk sebagian besar program Linux:

configure (atau ./configure tergantung pada sistem Anda) make install

Jika Anda bukan pengguna root, Anda perlu menjalankan skrip configure dengan opsi --prefix sebagai berikut:

```
./configure --prefix=/path/to/your/directory
```

Hal ini karena pengguna non-root tidak akan memiliki akses ke direktori default untuk GPG.

Jika semuanya berjalan lancar, GPG akan dikompilasi dan file yang dapat dieksekusi akan disalin ke /usr/local/bin/gpg atau direktori yang Anda tentukan. Anda dapat mengubah banyak opsi. Lihat dokumentasi GPG untuk detailnya.

Untuk server Windows, prosesnya sama mudahnya. Unduh file zip, ekstrak, dan tempatkan gpg.exe di suatu tempat di PATH Anda. (C:\Windows\ atau yang serupa akan baik-baik saja). Buat direktori di C:\gnupg. Buka prompt perintah dan ketik gpg.

Anda juga perlu menginstal GPG atau PGP dan membuat pasangan kunci pada sistem yang Anda rencanakan untuk memeriksa email.

Di server Web, hanya ada sedikit perbedaan antara versi baris perintah GPG dan PGP, jadi sebaiknya kita menggunakan GPG karena gratis. Pada mesin tempat Anda membaca email, Anda mungkin lebih suka membeli versi komersial PGP agar memiliki plug-in antarmuka pengguna grafis yang bagus untuk pembaca email Anda.

Jika Anda belum memilikinya, buat pasangan kunci pada mesin pembaca email Anda. Ingatlah bahwa pasangan kunci terdiri dari Kunci Publik yang digunakan orang lain (dan skrip PHP Anda) untuk mengenkripsi email sebelum mengirimkannya kepada Anda, dan Kunci Pribadi, yang Anda gunakan untuk mendekripsi pesan yang diterima atau menandatangani email keluar. Penting untuk membuat kunci di mesin pembaca email Anda, bukan di server Web Anda, karena kunci pribadi Anda tidak boleh disimpan di server Web.

Jika Anda menggunakan GPG versi baris perintah untuk membuat kunci, masukkan perintah berikut:

```
gpg --gen-key
```

Anda akan ditanya sejumlah pertanyaan. Sebagian besar pertanyaan memiliki jawaban default yang dapat diterima. Anda akan diminta memberikan nama dan alamat email, yang akan digunakan untuk memberi nama kunci. Kunci saya bernama 'Luke Welling <luke@tangledweb.com.au>'. Saya yakin Anda dapat melihat polanya.

Untuk mengekspor kunci publik dari pasangan kunci baru Anda, Anda dapat menggunakan perintah:

```
gpg --export > filename
```

Ini akan memberi Anda file biner yang cocok untuk diimpor ke gantungan kunci GPG atau PGP di mesin lain. Jika Anda ingin mengirim kunci ini melalui email kepada orang lain, sehingga mereka dapat mengimpornya ke dalam gantungan kunci mereka, Anda dapat membuat versi ASCII seperti ini:

```
gpg --export -a > namafile
```

Setelah mengekstrak kunci publik, Anda dapat mengunggah file ke akun Anda di server Web. Anda dapat melakukannya dengan FTP.

Perintah berikut mengasumsikan bahwa Anda menggunakan UNIX. Langkah-langkahnya sama untuk Windows, tetapi nama direktori dan perintah sistem akan berbeda. Masuk ke akun Anda di server Web dan ubah izin pada berkas tersebut sehingga pengguna lain dapat membacanya. Ketik

```
chmod 644 namaberkas
```

Anda perlu membuat keyring sehingga pengguna yang menjalankan skrip PHP Anda dapat menggunakan GPG. Pengguna ini bergantung pada cara server Anda diatur. Sering kali pengguna tersebut adalah "nobody", tetapi bisa juga pengguna lain.

Ubah menjadi pengguna server Web. Anda perlu memiliki akses root ke server untuk melakukan ini. Pada banyak sistem, server Web berjalan sebagai nobody. Contoh berikut

mengasumsikan hal ini. (Anda dapat mengubahnya ke pengguna yang sesuai pada sistem Anda.) Jika demikian halnya pada sistem Anda, ketik

```
su root
su nobody
```

Buat direktori untuk nobody guna menyimpan key ring dan informasi konfigurasi GPG lainnya. Direktori ini harus berada di direktori home nobody.

Direktori home untuk setiap pengguna ditentukan dalam /etc/passwd. Pada banyak sistem Linux, direktori home nobody secara default adalah /, yang tidak akan memiliki izin untuk ditulis oleh nobody. Pada banyak sistem BSD, direktori home nobody secara default adalah /nonexistent, yang, karena tidak ada, tidak dapat ditulisi. Pada sistem kami, nobody telah diberi direktori home /tmp. Anda perlu memastikan pengguna server Web Anda memiliki direktori home yang dapat mereka tulisi.
Ketik

```
cd ~
mkdir .gnupg
```

Pengguna nobody akan memerlukan kunci penandatanganan mereka sendiri. Untuk membuatnya, jalankan perintah ini lagi:

```
gpg --gen-key
```

Karena pengguna nobody Anda mungkin menerima sangat sedikit email pribadi, Anda dapat membuat kunci penandatanganan saja untuk mereka. Tujuan utama kunci ini adalah untuk memungkinkan kita memercayai kunci publik yang kita ekstrak sebelumnya.
Untuk mengimpor kunci publik yang kita ekspor sebelumnya, gunakan yang berikut:

```
gpg --import filename
```

Untuk memberi tahu GPG bahwa kita ingin memercayai kunci ini, kita perlu mengedit properti kunci menggunakan

```
gpg --edit-key ‘Luke Welling <luke@tangledweb.com.au>’
```

Pada baris ini, teks dalam tanda kutip adalah nama kunci. Jelas, nama kunci Anda tidak akan menjadi ‘Luke Welling <luke@tangledweb.com.au>’, tetapi kombinasi nama, komentar, dan alamat email yang Anda berikan saat membuatnya.

Opsi dalam program ini mencakup bantuan, yang akan menjelaskan perintah yang tersedia—percaya, tanda tangani, dan simpan.

Ketik percaya dan beri tahu GPG bahwa Anda memercayai kunci Anda sepenuhnya. Ketik tanda tangani untuk menandatangani kunci publik ini menggunakan kunci pribadi milik siapa pun. Terakhir, ketik simpan untuk keluar dari program ini, dan menyimpan perubahan Anda.

Menguji GPG

GPG sekarang seharusnya sudah disiapkan dan siap digunakan. Membuat berkas yang berisi beberapa teks dan menyimpannya sebagai test.txt akan memungkinkan kita untuk mengujinya. Mengetik perintah berikut

```
gpg -a --recipient 'Luke Welling <luke@tangledweb.com.au>' --encrypt
test.txt
```

(dimodifikasi untuk menggunakan nama kunci Anda) akan memberi Anda peringatan

Peringatan: menggunakan memori yang tidak aman! dan buat berkas bernama test.txt.asc.

Jika Anda membuka test.txt.asc, Anda akan melihat pesan terenkripsi seperti ini:

```
-----BEGIN PGP MESSAGE-----
Version: GnuPG v1.0.3 (GNU/Linux)
Comment: For info see http://www.gnupg.org

hQEOA0DU7hVGgdtnEAQAhr4HgR7xpIBsK9CiELQw85+k1QdQ+p/FzqL8tICrQ+B3
0GJTEehPUDErwqUw/uQLTds0r1oPSrIAZ7c6GVkh0YEVBj2MskT81IIBvdo95OyH
K9PUCvg/rLxJ1kxe4Vp8QFET5E3FdII/ly8VP5gSTE7gAgm0SbFf3S91PqwMyTkD
/2oJEvL6e3cP384s0i8lrBbDbOUAAhCjjXt2DX/uX9q6P18QW56UICUOn4DPaW1G
/gnNZCkcVDgLcKfBjbkB/TCWWhpA7o7kX4CIcIh7KlIMHY4RKdnCWQf271oE+8i9
cJRSCMsFIoI6MMNRCQHY6p9bfxL2uE39IRJrQbe6xoEe0nkB0uTYxiL0TG+FrNrE
tvBVMS0nsHu7HJey+oY4Z833pk5+MeVwYumJwlvHjdZxZmV6wz46GO2XGT17b28V
wSBnWOoBHSZsPvkQXHTOq65EixP8y+YJvBN3z4pzdH0Xa+NpqbH7q3+xXmd30hDR
+u7t6MxTLDbgC+NR
=gfQu
-----END PGP MESSAGE-----
```

Anda seharusnya dapat mentransfer berkas ini ke sistem tempat Anda membuat kunci awalnya dan menjalankan:

```
gpg -d test.txt.asc
```

untuk melihat teks asli Anda lagi.

Listing 9.1 private_mail.php—Formulir HTML Kami untuk Mengirim Email Terenkripsi

```
<html>
<body>
<h1>Send Me Private Mail</h1>
```

```
<?
  // you might need to change this line, if you do not use
  // the default ports, 80 for normal traffic and 443 for SSL
  if($HTTP_SERVER_VARS[“SERVER_PORT”]!=443)
    echo “<p><font color = red>
              WARNING: you have not connected to this page using SSL.
              Your message could be read by others.</font></p>”;
?>

<form method = post action = send_private_mail.php><br>
Your email address:<br>
<input type = text name = from size = 38><br>
Subject:<br>
<input type = text name = title size = 38><br>
Your message:<br>
<textarea name = body cols = 30 rows = 10>
</textarea><br>
<input type = submit value = “Send!”>
</form>
</body>
</html>
```

Untuk menempatkan teks dalam berkas, daripada menampilkannya di layar, Anda dapat menggunakan tanda -o dan menentukan berkas keluaran seperti ini:

```
gpg -do test.out test.txt.asc
```

Jika Anda telah mengatur GPG sehingga pengguna yang menjalankan skrip PHP Anda dapat menggunakannya dari baris perintah, berarti Anda sudah hampir berhasil. Jika ini tidak berhasil, hubungi administrator sistem Anda atau dokumentasi GPG.

Listing 9.1 dan 9.2 memungkinkan orang untuk mengirim email terenkripsi dengan menggunakan PHP untuk memanggil GPG.

Listing 9.2 `send_private_mail.php`—Skrip PHP Kita untuk Memanggil GPG dan Mengirim Email Terenkripsi

```
<?
  $to_email = “luke@localhost”;

  // Tell gpg where to find the key ring
  //  On  this  system,  user  nobody’s  home  directory  is  /tmp/
  putenv(“GNUPGHOME=/tmp/.gnupg”);
  //create a unique file name
  $infile = tempnam(“”, “pgp”);
  $outfile = $infile.”.asc”;
```

```
    //write the user's text to the file
    $fp = fopen($infile, "w");
    fwrite($fp, $body);
    fclose($fp);

    //set up our command
    $command =  "/usr/local/bin/gpg -a \\
                --recipient 'Luke Welling <luke@tangledweb.com.au>' \\
                --encrypt -o $outfile $infile";

    // execute our gpg command
    system($command, $result);

    //delete the unencrypted temp file
    unlink($infile);

  if($result==0)
    {
      $fp = fopen($outfile, "r");
      if(!$fp||filesize ($outfile)==0)
      {
        $result = -1;
      }
      else
      {
        //read the encrypted file
        $contents = fread ($fp, filesize ($outfile));
        //delete the encrypted temp file unlink($outfile);

        mail($to_email, $title, $contents, "From: $from\n");
        echo "<h1>Message Sent</h1>
             <p>Your message was encrypted and sent.
             <p>Thank you.";
      }
    }

    if($result!=0)
    {
      echo "<h1>Error:</h1>
           <p>Your message could not be encrypted, so has not been sent.
           <p>Sorry.";
    }
?>
```

Agar kode ini berfungsi untuk Anda, Anda perlu mengubah beberapa hal. Email akan dikirim ke alamat di `$to_email`.

Baris

```
putenv("GNUPGHOME=/tmp/.gnupg");
```

perlu diubah untuk mencerminkan lokasi keyring GPG Anda. Pada sistem kami, server Web berjalan sebagai pengguna nobody, dan memiliki direktori home/tmp/.

Kami menggunakan fungsi tempnam() untuk membuat nama berkas sementara yang unik. Anda dapat menentukan direktori dan awalan nama berkas. Kami akan membuat dan menghapus berkas-berkas ini dalam waktu sekitar satu detik, jadi tidak terlalu penting apa yang kami sebut. Kami menentukan awalan 'pgp', tetapi membiarkan PHP menggunakan direktori sementara sistem.

Pernyataan

```
$command =  "/usr/local/bin/gpg -a ".
            "--recipient 'Luke Welling <luke@tangledweb.com.au>' ".
            "--encrypt -o $outfile $infile";
```

menyiapkan perintah dan parameter yang akan digunakan untuk memanggil gpg. Perintah dan parameter tersebut perlu dimodifikasi agar sesuai dengan Anda. Seperti saat kita menggunakannya pada baris perintah, Anda perlu memberi tahu GPG kunci mana yang akan digunakan untuk mengenkripsi pesan.

Pernyataan

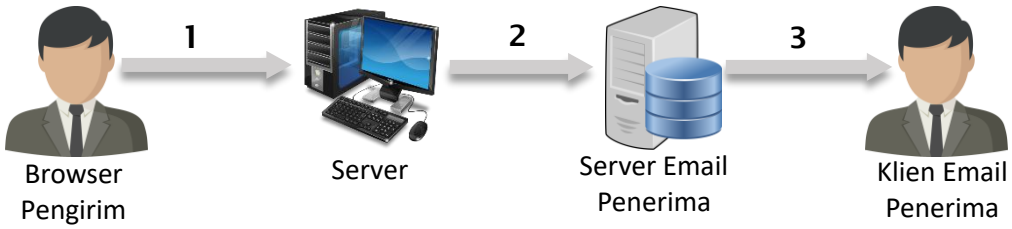
```
system($command, $result);
```

mengeksekusi instruksi yang disimpan dalam $command dan menyimpan nilai yang dikembalikan dalam $result.

Kita dapat mengabaikan nilai yang dikembalikan, tetapi pernyataan tersebut memungkinkan kita memiliki pernyataan if dan memberi tahu pengguna bahwa ada yang salah.

Setelah selesai dengan file sementara yang kita gunakan, kita menghapusnya menggunakan fungsi unlink(). Ini berarti bahwa email pengguna kita yang tidak terenkripsi disimpan di server untuk waktu yang singkat. Bahkan mungkin saja jika server gagal selama eksekusi, file tersebut dapat tertinggal di server.

Saat kita memikirkan keamanan skrip kita, penting untuk mempertimbangkan semua aliran informasi dalam sistem kita. GPG akan mengenkripsi email kita dan mengizinkan penerima untuk mendekripsinya, tetapi bagaimana informasi tersebut awalnya berasal dari pengirim? Jika kita menyediakan antarmuka Web untuk mengirim email terenkripsi GPG, aliran informasi akan terlihat seperti Gambar 9.5.



Gambar 9.5 Dalam aplikasi email terenkripsi kami, pesan dikirim melalui Internet sebanyak tiga kali.

Dalam gambar ini, setiap anak panah menunjukkan pesan kami dikirim dari satu mesin ke mesin lain. Setiap kali pesan dikirim, pesan tersebut melewati Internet dan mungkin melewati sejumlah jaringan dan mesin perantara.

Skrip yang kita lihat di sini ada di mesin yang diberi label Server Web dalam diagram. Di server Web, pesan akan dienkripsi menggunakan kunci publik penerima. Pesan tersebut kemudian akan dikirim melalui SMTP ke server email penerima. Penerima akan terhubung ke server emailnya, mungkin menggunakan POP atau IMAP, dan mengunduh pesan menggunakan pembaca email. Di sini, ia akan mendekripsi pesan menggunakan kunci pribadinya.

Transfer data pada Gambar 9.5 diberi label 1, 2, dan 3. Untuk tahap 2 dan 3, informasi yang dikirimkan adalah pesan terenkripsi GPG dan tidak terlalu berharga bagi siapa pun yang tidak memiliki kunci pribadi. Untuk transfer 1, pesan yang dikirimkan adalah teks yang dimasukkan pengirim dalam formulir.

Jika informasi kita cukup penting sehingga kita perlu mengenkripsinya untuk tahap kedua dan ketiga perjalanannya, agak konyol untuk mengirimkannya tanpa enkripsi untuk tahap pertama. Oleh karena itu, skrip ini berada di server yang menggunakan SSL.

Jika kita terhubung ke skrip kita menggunakan port selain 443, skrip akan memberikan peringatan. Ini adalah port default untuk SSL. Jika server Anda menggunakan port non-default untuk SSL, Anda mungkin perlu mengubah kode ini.

Daripada memberikan pesan kesalahan, kita dapat menangani situasi ini dengan cara lain. Kita dapat mengarahkan pengguna ke URL yang sama melalui koneksi SSL. Kita juga dapat memilih untuk mengabaikannya karena biasanya tidak penting jika formulir dikirimkan menggunakan koneksi aman. Yang biasanya penting adalah detail yang diketik pengguna ke dalam formulir dikirimkan kepada kita dengan aman. Kita dapat memberikan URL lengkap sebagai tindakan formulir kita.

Saat ini, tag formulir terbuka kita terlihat seperti ini:

```
<form method = post action = send_private_mail.php>
```

Kita dapat mengubahnya untuk mengirim data melalui SSL bahkan jika pengguna terhubung tanpa SSL seperti ini:

```
<form method = post action = “https://webserver/send_private_mail.php”>
```

Jika kita mengodekan URL lengkap seperti ini, kita dapat yakin bahwa data pengunjung akan dikirim menggunakan SSL, tetapi kita perlu mengubah kode setiap kali kita menggunakannya di server lain atau bahkan di direktori lain.

Meskipun dalam kasus ini, dan banyak kasus lainnya, tidaklah penting bahwa formulir kosong dikirim ke pengguna melalui SSL, biasanya merupakan ide yang baik untuk melakukannya. Melihat simbol gembok kecil di bilah status browser mereka meyakinkan orang bahwa informasi mereka akan dikirim dengan aman. Mereka seharusnya tidak perlu melihat sumber HTML Anda dan melihat apa atribut tindakan formulir tersebut.

DAFTAR PUSTAKA

Abdul Kadir. *Mudah Menpelajari Database MySQL*. ANDI: Yogyakarta, 2010.

Arief Ramadhan. *Pemrograman Web Database Dengan PHP Dan MySQL*. Elex Media Komputindo: Jakarta, 2006.

Arief, M. Rudyanto. *Pemrograman Web Dinamis menggunakan PHP dan MySQL*. ANDI: Yogyakarta, 2011.

Betha Sidik, Ir., Husni Iskandar Pohan. *Pemrograman Web dengan HTML*. Informatika: Bandung, 2007.

Budi Raharjo. *Belajar Otodidak Membuat Database Menggunakan MySQL*. Informatika Bandung, 2011.

Bunafit Nugroho. *PHP dan MySQL dengan editor Dreamweaver MX*. ANDI: Yogyakarta, 2004.

Bunafit Nugroho. *Rekayasa Perangkat Lunak Menggunakan UML dan Java*. ANDI: Yogyakarta, 2009.

Dennis, A., Wixom, B. H., & Tegarden, D. *System Analysis and Design with UML*. John Wiley & Sons Inc., 2010.

Fathansyah. *Basis Data*. Informatika Bandung, 2012.

Fatima, Siti. *Perancangan Sistem Informasi Penjualan Mebel Online pada UD. Melindo Jaya*. AMIK Royal Kisaran, 2013.

Fawaidus. *Jaringan Komputer LABKOM STIKOM Surabaya*, 2012.

Gregorius, Agung. *Buku Pintar HTML5 + CSS3 + DreamWeaver CS6*. Jubilee Enterprise: Yogyakarta, 2012.

Hakim, D. K., & Fauzan, A. *Aplikasi Tracer Study Teknik Informatika Universitas Muhammadiyah Purwokerto*, 2015.

Hakim, Lukmanul. *Membangun Web Berbasis PHP dengan Framework Codeigniter*. Lokomedia: Yogyakarta, 2010.

Hidayat, R. *Cara Praktis Membangun Website Gratis*. Elex Media Komputindo, 2010.

Husni. *Pemrograman Database Berbasis Web*. Graha Ilmu: Yogyakarta, 2007.

Indra Warman, M. & Zahni, A. *Rekayasa Web Untuk Pemesanan Handphone*. Jurnal Momentum, 2013.

Indra Yatini. *Aplikasi Pengolahan Citra Berbasis Web*. STMIK AKAKOM Yogyakarta, 2014.

Jogiyanto, Hartono. *Analisis dan Desain Sistem Informasi*, Edisi III. ANDI: Yogyakarta, 2005.

Kadir, Abdul. *Mudah Menjadi Programmer PHP*. ANDI: Yogyakarta, 2009.

Kurniawan, Y. *Aplikasi Web Database Dengan PHP Dan MySQL*. Elex Media Komputindo: Jakarta, 2002.

Kustiyahningsih, Yeni. *Pemrograman Basis Data berbasis web menggunakan PHP dan MySQL*. Graha Ilmu: Yogyakarta, 2011.

Madcom. *Pemrograman PHP dan MySQL Untuk Pemula*. ANDI: Yogyakarta, 2016.

Madcoms. *Menguasai XHTML, CSS, PHP dan MySQL melalui Dreamweaver*. ANDI, 2009.

Nugroho, Adi. *Rekayasa Perangkat Lunak Berorientasi Objek dengan Metode USDP*. ANDI: Yogyakarta, 2010.

Nugroho, Adi. *Rekayasa Perangkat Lunak Menggunakan UML dan Java*. ANDI: Yogyakarta, 2009.

Pooley, Rob, Pauline Wilcox. *Applying UML*. Butterworth-Heinemann: United Kingdom, 2003.

Prasetyo, Joko. *Implementasi MySQL pada Sistem Informasi*. 2020.

Pressman, R. S. *Software Engineering: A Practitioner's Approach*. Palgrave Macmillan, 2005.

R. Ramakrishnan and J. Gehrke. *Database Management System*. McGraw Hill Higher Education, USA, 2007.

Saputra, A. *Panduan Praktis Menguasai Database Server MySQL*. Elex Media Komputindo: Jakarta, 2011.

Saputra, Agus. *Webtrik: PHP, HTML5, dan CSS3*. Jakarta, 2012.

Simarmata, J. *Rekayasa Web*. Penerbit Andi: Yogyakarta, 2010.

Solichin, A. *Pemrograman Web Dengan PHP dan MySQL*. Budi Luhur: Jakarta, 2016.

Subagia, A. *Membuat Web dengan PHP 7 dan Database PDO MySQLi*, 2016.

Sunarfrihantono, Bimo. *PHP dan MySQL untuk Web*. ANDI OFFSET: Yogyakarta, 2003.

Sutabri, Tata. *Teknologi Informasi*. ANDI: Yogyakarta.

Sutarman. *Membangun Aplikasi Web dengan PHP dan MySQL*. Graha Ilmu: Yogyakarta, 2007.

Suyanto, M. *Strategi Periklanan pada e-commerce Perusahaan Top Dunia*. ANDI: Yogyakarta, 2007.

Syafrizal, Melwin. *Pengantar Jaringan Komputer (Ed.I)*. ANDI: Yogyakarta, 2005.

Tan, Anton. *Becoming The Best Salespeople*. PT. Elex Media Komputindo: Jakarta, 2010.

Tim EMS. *Teori dan Praktik PHP-MySQL untuk Pemula*. PT Elex Media Komputindo: Jakarta, 2014.

Widodo, P. P. *Menggunakan UML*. Informatika: Bandung, 2011.

Wijayanti, Eva Kurnia. *Rancangan Bangun Website penyewaan penjualan kamera*. Universitas Muhammadiyah Ponorogo, 2014.

Wulandari, Maya. *Pengembangan Aplikasi Web dengan PHP dan MySQL*. 2021.

PENGEMBANGAN WEB

MySQL

(My Structured Query Language)

Dr. Budi Raharjo, S.Kom, M.Kom, MM.

BIODATA PENULIS

Dr. Budi Raharjo, S.Kom, M.Kom, MM lahir di Semarang, tanggal 22 Februari 1985. Beliau adalah Alumni dari Universitas Bina Nusantara (BINUS University) Jakarta dan juga alumni Universitas Kristen Satya wacana (UKSW) Salatiga. Dr. Budi Raharjo telah menjadi Dosen pada Universitas STEKOM pada mata kuliah Kepemimpinan (Leadership), mata kuliah Pengantar Akuntansi, Manajemen Proses, Manajemen Akuntansi dan Manajemen Resiko Bisnis. Selain sebagai dosen Universitas STEKOM, Dr. Budi Raharjo, M.Kom, MM juga mempunyai bisnis sendiri dalam bidang perhotelan dan juga sebagai wirausaha dalam bidang pemasok unggas (ayam) beku, ke berbagai kota besar, khususnya Jakarta dan sekitarnya.

Pengalaman beliau berwirausaha menjadi bekal utama dalam penulisan buku ajar yang diterbitkan oleh Yayasan Prima Agus Teknik (YPAT) Semarang. Oleh sebab itu bukunya berisi langkah langkah praktis yang mudah diikuti oleh para mahasiswa, saat mahasiswa mengikuti proses perkuliahan pada Universitas Sains dan Teknologi Komputer (Universitas STEKOM). Memiliki Jabatan Akademik Lektor 300 dan Menjabat sebagai Wakil Rektor 1 bidang (Akademik) di kampus Universitas STEKOM Semarang.

YAYASAN PRIMA AGUS TEKNIK

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

ISBN 978-634-7227-22-5 (PDF)
9 786347 227225