

Pertermuan 10

DATABASE SECURITY

Pokok Bahasan / Materi

- ❑ Introduction
- ❑ Main Aspect of Database Security
 - Integrity
 - Confidentiality
 - Availability
- ❑ Access Control
 - Discretionary Access Control
 - Mandatory Access Control
- ❑ Conclusion
- ❑ References

Bagaimana cara berpikir tentang Ketidakamanan?

- ❑ Orang-orang adalah bagian dari masalah
- ❑ Orang jahat tidak mengikuti aturan.
- ❑ Perlu memahami jenis serangan apa yang mungkin untuk mengkompromikan suatu sistem
 - Prasyarat untuk memahami apa yang harus dilindungi dalam suatu sistem!

Penyebab Insiden Keamanan Perangkat Lunak

- ❑ Perangkat lunak Buggy dan konfigurasi yang salah
 - Bahasa program tidak aman
 - Program yang kompleks
- ❑ Kurangnya kesadaran dan pendidikan
 - Beberapa kursus dalam keamanan komputer
 - Pemrograman buku teks tidak menekankan keamanan
- ❑ Kegunaan yang buruk
 - Keamanan terkadang membuat hal-hal lebih sulit untuk digunakan
- ❑ Faktor-faktor ekonomi
 - Konsumen tidak peduli dengan keamanan
 - Keamanan sulit, mahal dan membutuhkan waktu
 - Sedikit audit keamanan
- ❑ Faktor manusia
 - Siapa penyerang?
 - Mengapa sistem diserang?

Ancaman Keamanan Terkadang

⦿ User errors

- Pengguna secara tidak sengaja meminta objek atau operasi yang dia tidak seharusnya diotorisasi.

⦿ Communications system errors

- Pengguna mengirim pesan yang harus dikirim ke pengguna lain.
- Sistem menghubungkan pengguna ke sesi milik pengguna lain dengan hak akses yang berbeda.

⦿ OS errors

- Secara tidak sengaja menimpa file dan menghancurkan bagian dari basis data.
- Mengambil file yang salah dan mengirimkannya ke pengguna.
- Gagal menghapus file yang harus dihapus.

Sumber-Ancaman Keamanan yang Disengaja

- ⦿ Pengguna secara sengaja mendapatkan akses tidak sah dan / atau melakukan operasi yang tidak sah pada database
- ⦿ Karyawan yang tidak puas yang terbiasa dengan sistem komputer organisasi berusaha membalas dendam
- ⦿ Mata-mata industri mencari informasi untuk pesaing

Metode-Ancaman Keamanan yang Disengaja

- ⦿ Penyadapan jalur komunikasi
- ⦿ Elektronik menguping mengambil sinyal elektronik
- ⦿ Membaca layar tampilan atau hasil cetak tanpa pengawasan
- ⦿ Meniru pengguna resmi atau pengguna dengan akses lebih besar
- ⦿ Menulis program untuk mem-bypass DBMS dan mengakses data database secara langsung
- ⦿ Menulis program aplikasi yang melakukan operasi tanpa izin
- ⦿ Memperoleh informasi tentang data tersembunyi dengan permintaan yang cerdas
- ⦿ Menghapus perangkat penyimpanan fisik dari fasilitas komputer
- ⦿ Membuat salinan file yang disimpan tanpa melalui DBMS
- ⦿ Menyuap, memeras, atau memengaruhi pengguna yang berwenang untuk mendapatkan informasi atau merusak basis data

Security Plan

- ⦿ Harus dimulai dengan langkah-langkah keamanan fisik untuk hambatan bangunan-fisik, mengontrol akses, memerlukan Id.Pengenal, masuk, dll.
- ⦿ Seharusnya memiliki lebih banyak keamanan fisik untuk fasilitas komputer, Misal pintu terkunci.
- ⦿ Kontrol keamanan tambahan untuk basis data

Otentikasi/Authentication

- ⦿ Otentikasi pengguna - memverifikasi identitas pengguna
- ⦿ Sistem operasi menggunakan profil pengguna, id pengguna, kata sandi, prosedur otentikasi, lencana, kunci, atau karakteristik fisik pengguna
- ⦿ Otentikasi tambahan dapat diperlukan untuk mengakses ID pengguna tambahan-basis data, Psw.

Profil Pengguna

- ⦿ Sistem memiliki profil pengguna untuk setiap id, memberikan informasi tentang pengguna
- ⦿ Profil yang disimpan harus dijaga keamanannya, mungkin dalam bentuk terenkripsi
- ⦿ Profil biasanya menyertakan kata sandi, yang diduga hanya diketahui oleh pengguna
- ⦿ Kata sandi harus dirahasiakan dan sering diubah
- ⦿ Sistem tidak boleh menampilkan kata sandi saat masuk

Prosedur Otentikasi Lainnya

- ⦿ Batasan kata sandi - pengguna menuliskannya, memilih kata-kata yang mudah ditebak, atau membaginya.
- ⦿ Dapat meminta pengguna untuk memasukkan lencana atau kunci untuk masuk ke workstation.
- ⦿ Suara, sidik jari, pemindaian retina, atau karakteristik fisik lainnya dapat digunakan.
- ⦿ Prosedur otentikasi dapat berupa serangkaian pertanyaan - membutuhkan waktu lebih lama dan lebih sulit untuk direproduksi daripada Psw.
- ⦿ Otentikasi dapat diminta lagi di basis data.
- ⦿ Pengguna harus diminta untuk menghasilkan PW tambahan untuk mengakses database.

Otorisasi / Authorization

- ⦿ DBMS yang dirancang untuk banyak pengguna memiliki subsistem keamanan
- ⦿ Menyediakan otorisasi-pengguna diberi hak untuk menggunakan objek database
- ⦿ Bahasa otorisasi - memungkinkan DBA untuk menulis aturan otorisasi yang menentukan pengguna mana yang memiliki jenis akses ke objek database.

Apa itu keamanan Database?

❑ Basis Data

- Ini adalah kumpulan informasi yang disimpan di komputer

❑ Keamanan

- Itu bebas dari bahaya

❑ Keamanan Basis Data

- Ini adalah mekanisme yang melindungi database terhadap ancaman yang disengaja atau tidak disengaja.

Atau

- Perlindungan dari upaya jahat untuk mencuri (melihat) atau mengubah data.

Apa itu Ancaman?

- ❑ Ancaman: Setiap situasi atau peristiwa, baik disengaja atau tidak, yang dapat mempengaruhi sistem dan akibatnya organisasi...
 - Computer Systems
 - Databases
- ⦿ Ancaman basis data:
 - Kehilangan integritas / *Loss of integrity*.
 - Kehilangan ketersediaan / *Loss of availability*.
 - Kehilangan kerahasiaan / *Loss of confidentiality*.

Ancaman (*Threats*)

Hardware

Kebakaran / Banjir / bom
Korupsi data karena kehilangan daya atau lonjakan. Kegagalan mekanisme keamanan memberikan akses yang lebih besar Pencurian peralatan
Kerusakan fisik peralatan.

DBMS and Application Software

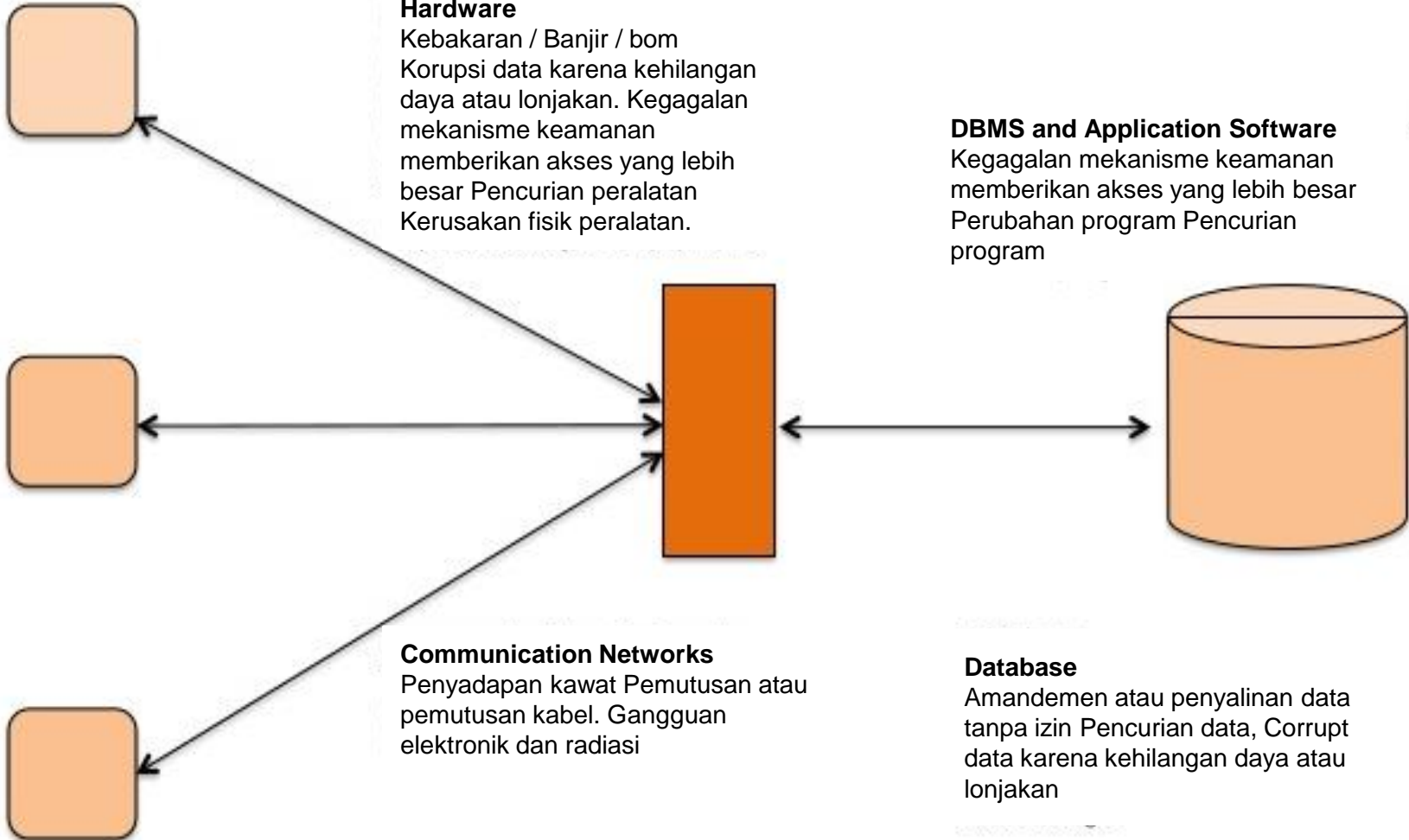
Kegagalan mekanisme keamanan memberikan akses yang lebih besar
Perubahan program Pencurian program

Communication Networks

Penyadapan kawat Pemutusan atau pemutusan kabel. Gangguan elektronik dan radiasi

Database

Amandemen atau penyalinan data tanpa izin Pencurian data, Corrupt data karena kehilangan daya atau lonjakan



Ancaman (*Threats*)

Hardware

Kebakaran / Banjir / bom
Korupsi data karena kehilangan

User

- Menggunakan cara akses orang lain
- Melihat dan mengungkapkan data yang tidak sah
- Pelatihan staf yang tidak memadai
- Entri ilegal oleh peretas
- Pemerasan
- Pengenalan virus

Programmers/ Operators

- Membuat pintu jebakan
- Perubahan program (seperti membuat perangkat lunak yang tidak aman)
- Pelatihan staf yang tidak memadai
- Kebijakan dan prosedur keamanan yang tidak memadai

Data/Database Administrator

- Keamanan yang tidak memadai
- Kebijakan dan prosedur

Database

Amandemen atau penyalinan data tanpa izin Pencurian data, Corrupt data karena kehilangan daya atau lonjakan

Definisi Database Security

- ❑ Database Security didefinisikan sebagai proses di mana "*Confidentiality, Integrity, dan Availability*" dari **Basis Data** dapat dilindungi.
- ❑ Berbagai tindakan balasan untuk ancaman yang terjadi dikaitkan dengan kontrol fisik sampai dengan prosedur administratif yang meliputi
 - Authorization
 - Authentication (Pembuktian keaslian)
 - Access control
 - Views
 - Backup and recovery
 - Journaling
 - Integrity
 - Encryption
 - RAID technology

⦿ **Authorization (Otorisasi)**

- Pemberian hak atau wewenang, yang menyebabkan subjek memiliki legitimasi untuk mengakses system atau objek-objek dalam system.

⦿ **Authentication (Pembuktian keaslian)**

- Suatu mekanisme yang menentukan apakah user yang mengakses benar-benar user yang dimaksud.

◎ **View**

- Merupakan hasil dinamis dari satu atau lebih operasi relasional yang dioperasikan pada relasi/table dasar untuk menghasilkan relasi/table lainnya. View merupakan relasi/table virtual yang tidak benar-benar ada dalam database, tetapi dihasilkan berdasarkan permintaan oleh user tertentu pada saat tertentu.

◎ **Back Up**

- Suatu proses yang secara periodik mengambil salinan database dan log file (dapat juga berupa program) untuk disimpan pada media penyimpanan offline.

⦿ **Journaling**

- Suatu proses pemeliharaan dan penyimpanan log file (jurnal) dari semua perubahan yang dilakukan terhadap database untuk kemudahan recovery bila terjadi kerusakan (failure).

⦿ **Integrity**

- Mencegah data dari ketidaksesuaian (invalid) dan mengakibatkan pemberian hasil yang salah.

⦿ **Encryption**

- Penyandian (*encoding*) data dengan menggunakan algoritma khusus yang membuat data tidak dapat dibaca oleh program tanpa kunci decryption.

◎ RAID (*Redundant Array of Independent Disks*) Technology

- Hardware dimana DBMS berjalan dengan fault-tolerant, yang berarti bahwa DBMS harus tetap melanjutkan operasi walaupun terdapat satu komponen hardware yang rusak (fail).
- Memberikan kesan memiliki komponen redundant (lebih) yang dapat diintegrasikan kedalam sistem kerja walaupun terdapat satu atau lebih kerusakan komponen.
- Komponen hardware utama yang harus memiliki fault-tolerant meliputi disk drives, disk controllers, CPU, Power supplies, cooling fans.
- Disk drives merupakan komponen yang paling mudah diserang dengan jarak yang dekat antar kerusakan dibandingkan dengan komponen hardware lainnya.
- Salah satu solusinya dengan menggunakan RAID technology, yaitu menyediakan serangkaian besar disk, yang terdiri dari susunan beberapa disk independen diatur untuk memperbaiki ketahanan (reliability) dan meningkatkan performa (*performance*).

- Performa (*performance*) meningkat melalui data striping, yaitu data disegmentasi (dibagi) menjadi beberapa bagian dengan ukuran yang sama (striping units), yang secara jelas didistribusikan melewati beberapa disk.
- Ketahanan (reliability) diperbaiki melalui penyimpanan informasi ganda (redundant) melewati disk dengan menggunakan skema parity atau skema error-correcting

Konsep Database security

⦿ Tiga Aspek Utama

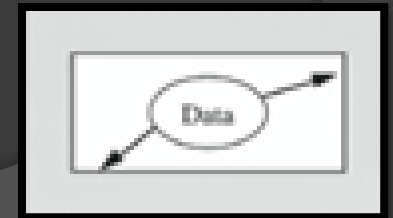
- Confidentiality
- Integrity
- Availability

⦿ Ancaman terhadap Basis Data:

- Loss of Integrity
- Loss of Availability
- Loss of Confidentiality

◎ Confidentiality

- Tidak ada yang bisa membaca data / komunikasi kami kecuali kami menginginkannya
- Ini melindungi database dari pengguna yang tidak sah.
- Memastikan bahwa pengguna diizinkan untuk melakukan hal-hal yang mereka coba lakukan.
- Sebagai contoh:
 - Karyawan tidak boleh melihat gaji manajer mereka.
- Kerahasiaan (*Confidentiality*) meliputi:
 - Privasi: perlindungan data pribadi,
 - Kerahasiaan: perlindungan data organisasi

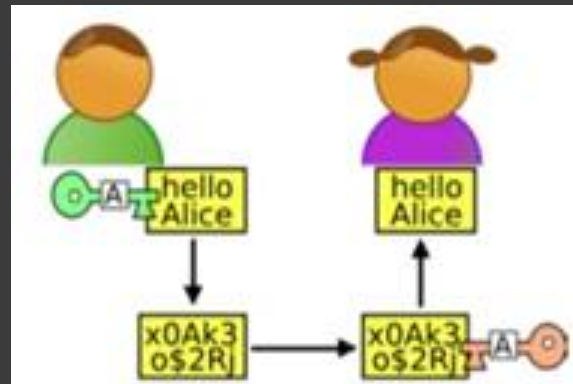


⦿ Confidentiality

- Contoh :

Bagaimana cara memastikan kerahasiaan data?

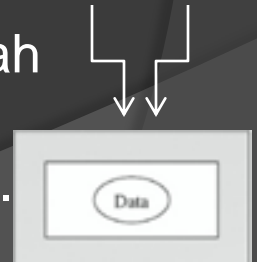
- Kriptografi



- Kontrol Akses Yang Kuat
- Membatasi jumlah tempat di mana data dapat muncul

◎ Integrity

- Tidak seorang pun dapat memanipulasi data / pemrosesan / komunikasi kami kecuali kami menginginkannya
- Melindungi database dari pengguna yang berwenang.
- Memastikan bahwa apa yang pengguna coba lakukan adalah benar
- Sebagai contoh:
 - Seorang karyawan harus dapat mengubah informasinya sendiri.
- "Memastikan bahwa semuanya seperti yang seharusnya. Mencegah penulisan atau modifikasi yang tidak sah



◎ **Integritas**

- Bagaimana integritas data dipertahankan?
 - Melalui Kendala integritas data
 - Kendala membatasi nilai data yang dapat dimasukkan atau diperbarui

◎ Column CHECK constraints

- Example :

```
CREATE TABLE test
```

```
(rollno number(2) check (rollno between 1 and 50),  
name varchar2(15));
```

```
INSERT INTO test values(45, ' Willy' );
```

1 row inserted

```
INSERT INTO test values(55, ' Hiess' );
```

ERROR-Check constraints violated

- ⦿ **Referential Integrity** adalah sebuah cara untuk menjaga konsistensi data antara tabel yang saling ber-Relasi.

Referential Integrity

Parent Key
Primary key of
referenced table

Referenced or Parent Table

Table DEPARTMENTS

DEPARTMENT_ID	DEPARTMENT_NAME	MANAGER_ID	LOCATION_ID
60	IT	103	1400
90	Executive	100	1700

Foreign Key
(values in dependent
table must match a
value in unique key
or primary key of
referenced table)

Dependent or Child Table

Table EMPLOYEES

EMPLOYEE_ID	LAST_NAME	EMAIL	HIRE_DATE	JOB_ID	MANAGER_ID	DEPARTMENT_ID
100	King	SKING	17-JUN-87	AD_PRES		90
101	Kochhar	NKOCHHAR	21-SEP-89	AD_VP	100	90
102	De Hann	LDEHANN	13-JAN-93	AD_VP	100	90
103	Hunold	AHUNOLD	03-JAN-90	IT_PROG	102	60

INSERT
INTO

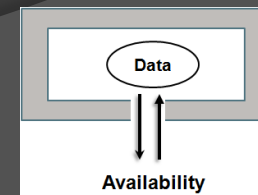
This row violates the referential
constraint because "99" is not
present in the referenced table's
primary key; therefore, the row
is not allowed in the table.

207	Ashdown	AASHDOWN	17-DEC-07	MK_MAN	100	99
208	Green	BGREEN	17-DEC-07	AC_MGR	101	

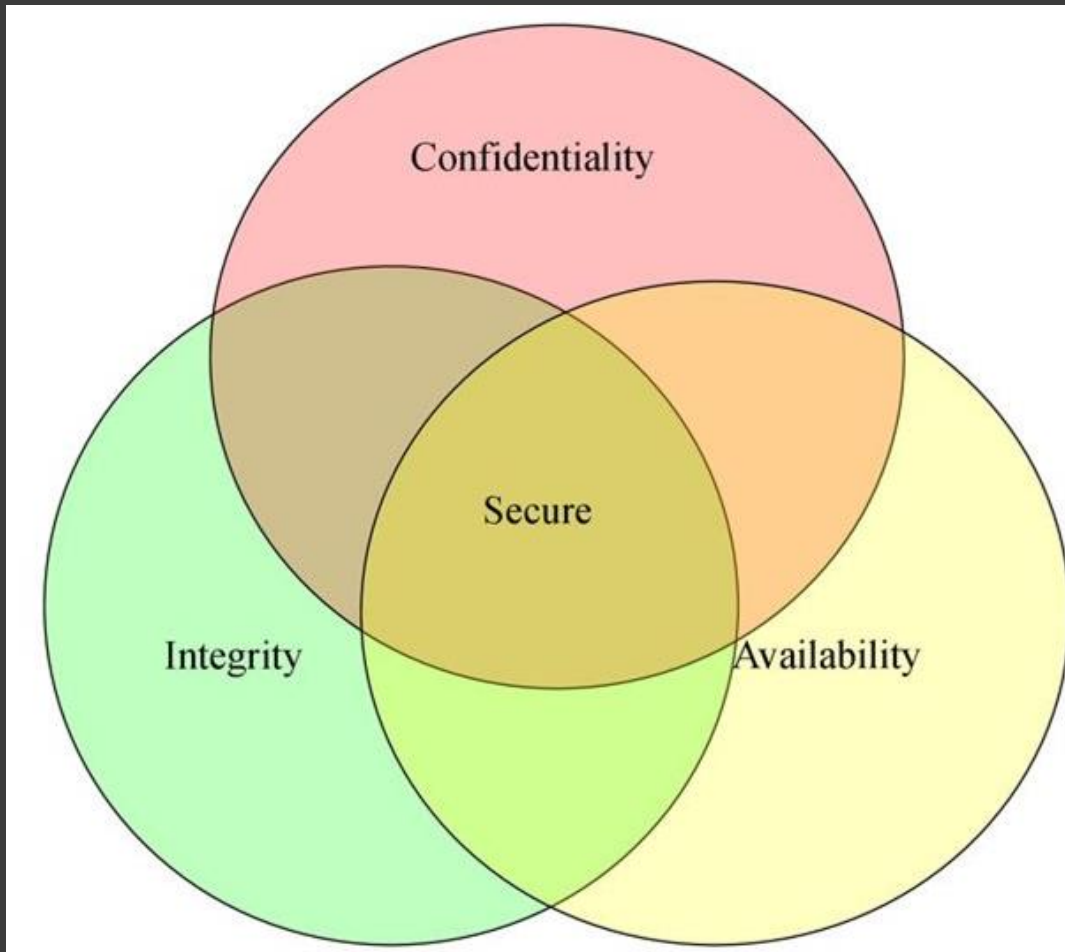
This row is allowed in the table
because a null value is entered
in the DEPARTMENT_ID column;
however, if a not null constraint
is also defined for this column,
this row is not allowed.

● Availability

- Kita dapat mengakses data kita / melakukan pemrosesan kita / menggunakan kemampuan komunikasi kita kapan saja kita mau
- Pengguna yang berwenang harus dapat mengakses data untuk Tujuan sebagaimana diperlukan
- Sebagai contoh:
 - Pesanan pembayaran terkait pajak harus dilakukan tepat waktu oleh undang-undang perpajakan.
- Layanan dapat diakses dan digunakan (tanpa penundaan) kapan pun dibutuhkan oleh entitas yang berwenang.



Relationship between Confidentiality Integrity and Availability



Access Control

- ⦿ Identitas memungkinkan akses ke sumber daya
- ⦿ Dalam keamanan komputer ini disebut
 - Access Control
 - Otorisasi
- ⦿ Kita bicara tentang:
 - Subjek (untuk siapa tindakan dilakukan)
 - Objek (berdasarkan tindakan yang dilakukan)
 - Operasi (jenis tindakan yang dilakukan)

Access Control Models

- DBMS menyediakan mekanisme kontrol akses untuk membantu menerapkan kebijakan keamanan.
- Dua jenis mekanisme yang saling melengkapi:

- 1. Discretionary access control (DAC)**

Model DAC menegakkan kontrol akses berdasarkan identitas pengguna, kepemilikan objek, dan delegasi izin. Pemilik objek dapat mendelegasikan izin objek ke pengguna lain.

- 2. Mandatory access control (MAC)**

Model MAC mengatur akses berdasarkan tingkat sensitivitas subjek dan objek. Subjek dapat membaca objek jika tingkat keamanan subjek lebih tinggi dari objek.

- 3. Role-Based Access Control (RBAC)**

Model MAC mengatur akses berdasarkan tingkat sensitivitas subjek dan objek. Subjek dapat membaca objek jika tingkat keamanan subjek lebih tinggi dari objek.

Access Control

◎ Discretionary Access Control (DAC)

- Ide Mencapai keamanan berdasarkan konsep hak akses:
 - ❖ Hak istimewa untuk objek (hak akses tertentu untuk tabel, kolom, dll.), Dan
 - ❖ Mekanisme untuk memberikan hak pengguna (dan mencabut hak istimewa)
 - Pengguna diberi hak istimewa untuk mengakses objek skema yang sesuai (tabel, tampilan).
 - Pengguna dapat memberikan hak istimewa kepada pengguna lain atas kebijakan mereka sendiri.
 - Implementasi: perintah GRANT dan REVOKE

- **GRANT** command: Berikan hak istimewa kepada pengguna untuk mendasarkan tabel dan tampilan.

GRANT privileges ON object TO users [WITH GRANT OPTIONS]

- **REVOKE** command: dimaksudkan untuk mencapai kebalikannya, untuk menarik hak istimewa yang diberikan dari pengguna.

REVOKE [GRANT OPTION FOR] privileges ON object FROM users {RESTRICT | CASCADE}

Granting/Revoking Privileges

```
GRANT SELECT ON database.* TO user@'localhost';
```

```
GRANT SELECT ON database.* TO user@'localhost' IDENTIFIED BY  
    'password';
```

Access Control

◎ Discretionary Access Control (DAC)

- Akses ke objek data (file, direktori, dll.) Diizinkan berdasarkan identitas pengguna.
- Aturan akses eksplisit yang menetapkan siapa yang bisa, atau tidak bisa, melakukan tindakan mana pada sumber daya mana.
- Kebijakan(*Discretionary*): pengguna dapat diberikan kemampuan untuk menyerahkan hak istimewa mereka kepada pengguna lain, di mana **pemberian** dan **pencabutan** hak istimewa diatur oleh kebijakan administratif.

Access Control

◎ Discretionary Access Control (DAC)

- DAC fleksibel dalam hal spesifikasi kebijakan
- Ini adalah bentuk kontrol akses yang diterapkan secara luas di platform multi-pengguna standar Unix, NT, Novell, dll.

Access Control

◎ Discretionary Access Control (DAC)

- Access control matrix
 - Menjelaskan kondisi perlindungan dengan tepat
 - Matriks yang menggambarkan hak-hak subyek
 - Status transisi mengubah elemen-elemen matriks
- Keadaan sistem perlindungan (State of protection system)
 - Menjelaskan pengaturan saat ini, nilai-nilai sistem yang relevan dengan perlindungan

Two problems with DAC :



- ⦿ Anda tidak dapat mengontrol jika seseorang yang Anda bagikan file tidak akan membagikan data yang terkandung di dalamnya
 - Cannot control “information flow”
- ⦿ Di banyak organisasi, pengguna tidak dapat memutuskan bagaimana tipe data tertentu dapat dibagikan
 - Biasanya pimpinan dapat mengamankan bagaimana berbagi berbagai jenis data sensitif
- ⦿ Mandatory Access Control (MAC) membantu mengatasi masalah ini

◎ **MAC: Mandatory Access Control**

- Kebijakan seluruh sistem menetapkan siapa yang diizinkan memiliki akses
- Mengandalkan sistem untuk mengontrol akses daripada individu
- Model ini digunakan di lingkungan yang sangat rahasia dan rahasia (mis. Militer)
- Contoh: Undang-undang mengizinkan pengadilan untuk mengakses catatan mengemudi tanpa izin pemilik

◎ Security Policy Model

- Model kebijakan keamanan adalah pernyataan singkat dari properti perlindungan yang harus dimiliki suatu sistem, atau jenis sistem generik
- Mekanisme MAC tradisional telah digabungkan dengan beberapa model keamanan
- Baru-baru ini, sistem mendukung model keamanan yang fleksibel (mis., ., SELinux, Trusted Solaris, TrustedBSD, dll.)

◎ Why MAC?

- Kebutuhan akan konsistensi kebijakan global yang tidak dapat dipenuhi oleh DAC
- Kontrol informasi mengalir satu objek ke objek lain, sehingga akses ke salinan tidak dimungkinkan jika pemilik dokumen asli tidak memberikan akses
- Kontrol untuk mencegah perangkat lunak berbahaya / cacat memodifikasi kebijakan sistem. DAC tidak dapat mencegah hal ini jika program dijalankan oleh akses pemilik.

◎ Multilevel Security

- Orang dan Informasi diklasifikasikan ke dalam tingkat kepercayaan dan sensitivitas yang berbeda



- Tingkat izin (Clearance level): Mengindikasikan tingkat tertinggi informasi rahasia yang akan disimpan atau ditangani oleh orang, perangkat, atau lokasi
- Tingkat klasifikasi (Classification level): Menunjukkan tingkat kerusakan yang dapat diderita negara jika informasi tersebut diungkapkan kepada musuh
- Tingkat keamanan adalah istilah umum untuk tingkat izin atau tingkat klasifikasi

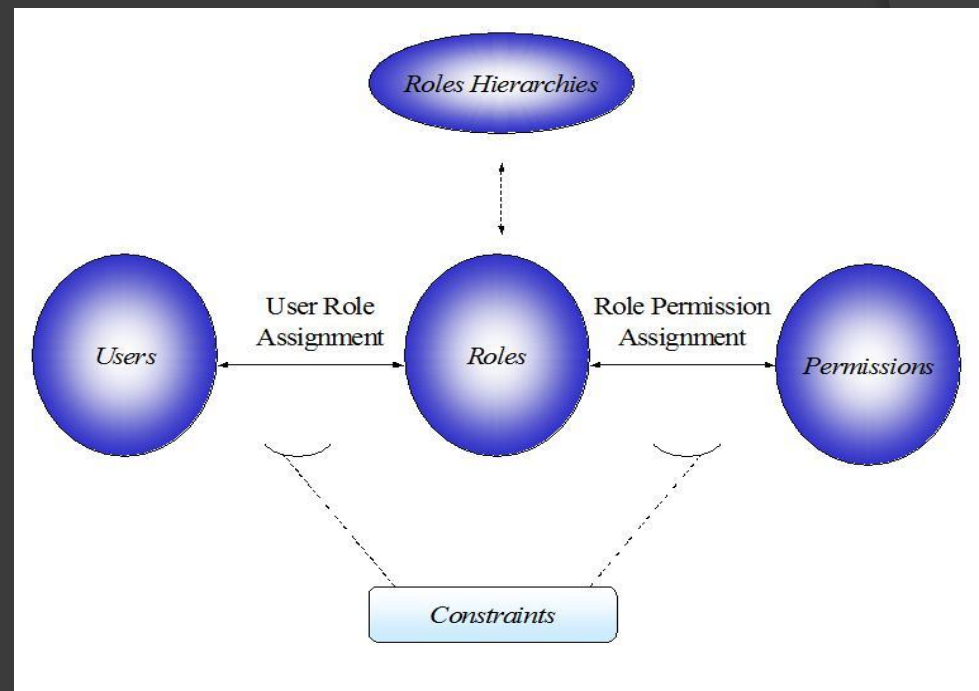
◎ **MAC: Mandatory Access Control**

Ringkasan :

- Memberikan perusahaan kemampuan untuk mengendalikan bagaimana berbagi informasi sensitif dapat dikendalikan
- Dapat mengatasi kerahasiaan dan integritas tetapi membutuhkan fungsionalitas tambahan dengan label
- Jaminan tingkat tinggi untuk sistem tepercaya memang menantang

● Role Based Access Control (RBAC)

- Kontrol akses dalam organisasi didasarkan pada "peran yang diambil pengguna individu sebagai bagian dari organisasi"
- Peran adalah "adalah kumpulan izin"



⦿ **Akses tergantung pada peran / fungsi, bukan identitas**

- Contoh: Allison adalah pemegang buku untuk Departemen Matematika. Ia memiliki akses ke catatan keuangan. Jika dia pergi dan Betty dipekerjakan sebagai pemegang buku baru, Betty sekarang memiliki akses ke catatan-catatan itu. Peran "pemegang buku" menentukan akses, bukan identitas individu.

◎ **Keuntungan dari RBAC**

- Memungkinkan Manajemen Keamanan yang Efisien
 - Peran administratif, Hirarki peran
- Prinsip privilege paling tidak memungkinkan meminimalkan kerusakan
- Pemisahan kendala Tugas untuk mencegah penipuan
- Mengizinkan pengelompokan objek
- Kebijakan-netral - Memberikan sifat umum
- Meliputi kebijakan DAC dan MAC

DBMSs and Web Security

◎ Penanggulangan

- Proxy servers
- Firewalls
- Secure Socket Layer atau SSL Yang digunakan secara luas untuk mengamankan e-commerce di Internet saat ini..

Proxy Servers

Definisi

Server proxy adalah komputer yang berada di antara browser Web dan server Web. Itu memotong semua permintaan untuk halaman web dan menyimpannya secara lokal untuk beberapa waktu. Server proxy menyediakan peningkatan kinerja dan permintaan filter.



Firewalls

- ◎ Firewall adalah sistem yang mencegah akses tidak sah ke atau dari jaringan pribadi. Diterapkan dalam perangkat lunak, perangkat keras atau keduanya.
 - Packet filter
 - Application gateway
 - Proxy server

Conclusion

- ⦿ Keamanan data sangat penting.
- ⦿ Memerlukan keamanan di tingkat yang berbeda.
- ⦿ Beberapa solusi teknis.
- ⦿ Tetapi pelatihan manusia sangat penting.



**TERIMA
KASIH
DAN
APA ADA
PERTANYAAN?**