BAB 11
SECURITY (KEAMANAN)

11.1. *Level Security*

Security adalah merupakan tindakan untuk memproteksi kejahatan untuk mencuri atau memodifikasi data dalam sistem database. Macam – macam level dalam melakukan security terhadap data adalah sebagai berikut :

1. Database system level.
   Merupakan mekanisme autentikasi dan otorisasi untuk mengijinkan pemakai tertentu melakukan akses data yang diperlukan saja.
2. Operating system level.
   Operating system super-user dapat melakukan apapun terhadap database. Kemanan sistem operasi yang handal dan bagus diperlukan dalam hal ini.
3. Network level.
   Pada level ini proses kemanan harus menggunakan enkripsi untuk menjaga :
   - Eavesdropping (pembacaan yang tidak terotorisasi terhadap pesan – pesan tertentu.
   - Masquerading (berpura – pura menjadi pemakai yang sah atau mengirimkan pesan yang seolah berasal dari pemakai yang sah).
4. Physical Level.
   Yaitu melakukan akses fisik terhadap komputer memungkinkan terjadinya perusakan data, kemanan dengan menggunakan kunci yang diperlukan. Komputer juga harus diamankan dari kebanjiran, kebakaran dan lainnya.
5. Human Level.
   Pemakai harus disaring dahulu untuk memastikan bahwa pemakai yang sah tidak memperbolehkan memberikan hak akses kepada orang lain (penyusup). Pemakai harus dilatih dalam pemilihan password dan menjaga kerahasiaannya.

11.2. Otorisasi

Bentuk otorisasi yang diperbolehkan kepada pemakai (user) dalam suatu database dalam suatu perusahaan adalah sebagai berikut :

1. Read authorization.
   Merupakan hak akses yang diperuntukkan user untuk diijinkan melakukan pembacaan data tetapi tidak diberikan ijin untuk melakukan modifikasi data yang ada.
2. Insert authorization.
   Merupakan hak akses yang diperuntukkan user untuk diijinkan melakukan penyisipan data baru tetapi tidak diberikan ijin untuk melakukan modifikasi data yang ada.
3. Update authorization.

Merupakan hak akses yang diperuntukkan user untuk diijinkan melakukan modifikasi data tetapi tidak diberikan ijin untuk melakukan penghapusan data yang ada.

Bentuk otorisasi yang diperbolehkan kepada pemakai (user) dalam memodifikasi skema database dalam suatu perusahaan adalah sebagai berikut :

1. Indeks authorization.
   Merupakan hak akses yang diperuntukkan user untuk diijinkan melakukan pembuatan dan penghapusan indeks.
2. Resources authorization.
   Merupakan hak akses yang diperuntukkan user untuk diijinkan melakukan pembuatan relasi (hubungan) baru dalam database.
3. Alteration authorization.
   Merupakan hak akses yang diperuntukkan user untuk diijinkan melakukan penambahan dan penghapusan atribut baru dalam relasi (hubungan) dalam database.
4. Drop authorization.
   Merupakan hak akses yang diperuntukkan user untuk diijinkan melakukan penghapusan relasi dalam database.

11.3. Otorisasai dan View

Pemakai (user) dapat diberikan hak otorisasi pada view, tanpa diberikan otorisasi apaun terhadap relasi (hubungan) yang digunakan dalam definisi view. Kemampuan view untuk menyembutikan data memberikan kesederhanaan dalam penggunaan sistem dan untuk meningkatkan keamanan dengan mengijinkan pemakai hanya mengakses data yang diperlukan untuk pekerjaannya masing – masing. Kombinasi dari keamanan data pada tingkat relational dan tingkat view dapat digunakan untuk membatasi akses pemakai dengan tepat terhadap data yang dibutuhkan saja.

1. Contoh View

Misalkan seorang pegawai Bank ingin perlu mengetahui nama nasabah setiap cabang, tetapi tidak diperkenankan melihat informasi mengenai pinjaman.

- Pendekatan : tiadakan akses langsung terhadap relasi (hubungan) loan, tetapi diberikan hak akses terhadap view cust_loan, yang berisikan nama – nama nasabah dan cabang dimana mereka memiliki pinjaman.
- View cust_loan didefinisikan dalam SQL sebagai berikut :

```
Create view cust_loan as
        Select branchname, customer_name
        From borrower, loan
        Where borrower.loan_number = loan.loan_number
```

- Pegawai Bank tersebut diberikan otorisasi untuk melihat hasil dari query berikut ini :

```
Select *
From cust_loan
```

- Jika query prosessor menterjemahkan hasilnya kedalam sebuah query pada relasi aktual yang ada dalam database, maka kita akan memperoleh sebuah query pada borrower dan loan.
- Otorisasi harus diuji pada query pegawai Bank tersebut sebelum pemrosesan query dimulai.

2. Otorisasi dalam View

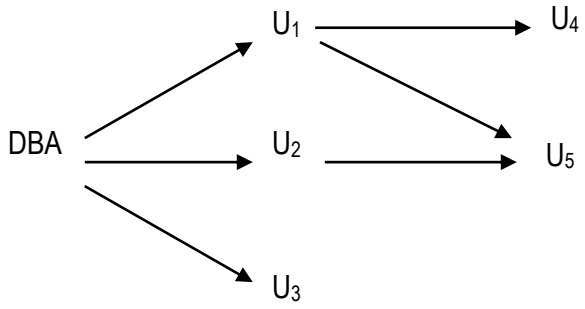
Pembuatan view tidak membutuhkan otorisasi terhadap sumberdaya yang diperlukan, dikarenakan tidak ada relasi nyata yang dibuat. Pembuat view memperoleh hak hanya sebagaimana yang telah dia miliki sebelumnya, tidak lebih.
Misalkan : jika pembuat view cust_loan hanya memiliki otorisasi read pada borrower dan loan maka dia hanya memperoleh otorisasi read pada cust_loan.

11.4. Pemberian Hak

Pemberian otorisasi dari satu pemakai pada pemakai yang lain dapat ditunjukkan dengan (graph) grafik otorisasi. Titik (node) dari graph adalah pemakai (user). Akar (root) dari graph adalah administrasi database.

Berikut ini contoh graph untuk otorisasi update pada loan :

- Sisi (edge) $U_i \rightarrow U_j$ menunjukkan bahwa pemakai U_i telah memberikan otorisasi update pada loan untuk U_j.



Gambar 11.1. Pemberian hak

11.5. Spesifikasi Security dalam SQL

Spesifikasi sistem keamanan (security system) SQL didalam database adalah ada beberapa hal berikut ini :

- Pernyataan Grant digunakan untuk memberikan otoritas, dengan statement berikut ini :

  Grant <daftar hak>
  On <nama relasi atau view> **to** <daftar pemakai>
- <daftar pemakai adalah :
  - Identitas pemakai.
  - Public, yang mengijinkan semua pemakai yang valid akan memperoleh hak akses.
  - Role.

- Pemberian hak pada view tidak berarti memberikan hak kepada relasi yang mendasarinya, pemberian hak harus memiliki hak tersebut terlebih dahulu (atau menjadi administrator database).

1. Hak dalam SQL.

Berikut ini adalah beberapa hak yang diberikan kepada user dalam mengakses database :

- ***Select***, mengijinkan akses read pada relasi, atau query menggunakan view.
  Contoh : berikan pemakai U_1, U_2 dan U_3 untuk otorisasi ***select*** pada relasi branch.
  Grant select on branch ***to*** U_1, U_2 , U_3
- ***Insert***, kemampuan untuk menambahkan tupple dalam relasi didalam suatu database.
- ***Update***, kemampuan untuk melakukan perubahan tupple dalam relasi didalam suatu database.
- ***Delete***, kemampuan untuk melakukan penghapusan tupple dalam relasi didalam suatu database.
- ***References***, kemampuan untuk mendeklarasikan foreign key pada saat membuat relasi didalam suatu database.
- ***Usage***, dalam SQL-92 yaitu kemampuan untuk memberikan otorisasi pemakai untuk mempergunakan domain tertentu.
- ***All Privileges***, kemampuan untuk melakukan pemakaian hal seluruhnya didalam suatu database.

2. Hak untuk memberikan Hak

Dengan pilihan grant, memungkinkan pemakai yang mempunyai hak tersebut memberikan hak akses kepada pemakai yang lainnya didalam suatu database. Contoh berikut ini adalah hak untuk memberikan hak :

Grant select on branch ***to*** U_1 ***with grant option***

Berikan user U_1 hak melakukan **select** pada relasi branch dan ijinkan user U_1untuk memberikan hak tersebut kepada pemakai yang lainnya.

3. Role

Role memungkinkan hak yang sama diberikan kepada sekelompok pemakai sekali saja dengan membuat role yang sesuai. Haknya dapat diberikan atau diambil dari role, seperti pada pemakai. Role dapat diberikan kepada pemakai atau role yang lainnya. SQL-1999 mendukung penggunaan role, sebagai contoh berikut ini :

Create role teller
Create role manager

Grant select on branch **to** teller
Grant update (balance) **on** account **to** teller
Grant all privileges on account **to** manager

Grant teller **to** manager
Grant teller **to** alice, bob
Grant manager **to** avi

4. Pencabutan otorisasi dalam SQL

Dalam pencabutan otorisasi didalam database kepada penggunakan maka digunakan pernyataan **revoke**, dengan statement berikut ini :

Revoke<daftar hak>
On<nama relasi atau view>**from**<daftar pemakai>
[restrict|cascade]

Contoh berikut ini adalah pencabutan hak otorisasi dalam SQL :

Revoke select on branch **from** U_1, U_2, U_3 **cascade**

Yaitu melakukan pencabutan hak dari seorang pemakai dapat menyebabkan pemakai lain juga kehilangan hak tersebut (cascade). Hal tersebut dapat dicegah dengan menentukan restrict :

Revoke select on branch **from** U_1, U_2, U_3 **restrict**

Pertanyaan Soal

1. Jelaskan pengertian physical level dan operating system level dalam security ?.
2. Jelaskan pengertian read dan update authorization dalam database ?.
3. Berikan suatu contoh graph untuk otorisasi update pada suati relasi dalam suatu database ?.
4. Jelaskan dan berikan contoh pencabutan otorisasi dalam SQL dengan menggunakan pernyataan revoke ?.