

4. SQL SERVER SECURITY AND ROLES

Objektif :

Setelah menyelesaikan materi ini, mahasiswa dapat melakukan hal sebagai berikut :

1. Membuat SQL Server Login yang baru beserta user
 2. Menggunakan Statement GRANT, DENY dan REVOKE untuk memberikan dan mencabut hak akses dari suatu objek
 3. Membuat sebuah ROLE dan memberikan hak akses kepada sebuah ROLE
-

4.1. Creating a New SQL Server Login

Dengan SQL Server security, user membuat LOGIN ID yang benar – benar terpisah dari informasi network login User. Terdapat empat cara untuk membuat login dalam SQL Server, tetapi hanya satu cara yang akan di bahas di sini :

1. Dengan menggunakan Transact – SQL (T-SQL)
2. Dengan menggunakan fitur SQL Management Studio
3. Dengan menggunakan SQL Distributed Management Objects (DMO)
4. Dengan menggunakan Windows Management Instrumentation (WMI)

4.1.1. Dengan menggunakan Transact – SQL

Dalam pembuatan login id baru di dalam sql server 2008. Login id baru secara default sudah mempunyai hak akses untuk melakukan login, tetapi tidak mempunyai hak untuk melakukan aksi khusus kepada object di dalam database. Untuk membuat login id, yang pertama kali diperhatikan adalah harus login sebagai sa (dba / database administrator).

Syntax membuat login id :

```
CREATE LOGIN <loginName> WITH PASSWORD = '<enterStrongPasswordHere>'
[MUST_CHANGE]
```

*Jika memakai MUST_CHANGE, SQL Server meminta pengguna untuk mendapatkan password baru pertama kali login baru digunakan

Contoh : membuat login baru dengan login name “james” dan password “12345”

Gunakan Login Sa

```
CREATE LOGIN james WITH PASSWORD = '12345';
GO
```



- Memodifikasi Login ID

Ketika login telah terbuat, login id masih membutuhkan mekanisme untuk merubah informasi password ,login name, atau bahkan mengubah status hak untuk melakukan login.

Syntaxnya adalah :

```
ALTER LOGIN login_name [status_option] WITH <set_option> = <new_condition>
```

Contoh :

Gunakan Login Sa

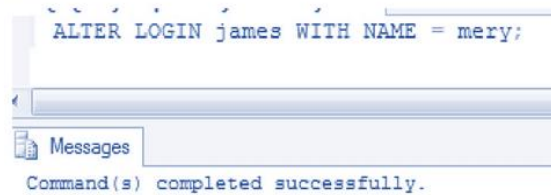
A. Mengubah password login

```
ALTER LOGIN james WITH PASSWORD = '123456';
```



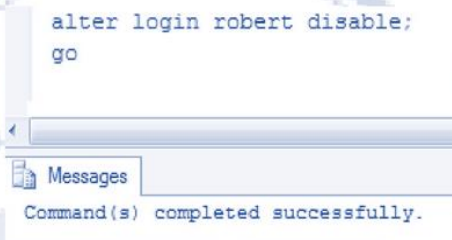
B. Mengubah login name

```
ALTER LOGIN james WITH NAME = mery;
```



C. Mengubah hak akses login

```
alter login robert disable;  
go
```



4.2. Membuat User

Setelah selesai membuat login ID langkah selanjutnya membuat user, user berfungsi untuk menerima hak atau izin untuk mengoperasikan database.

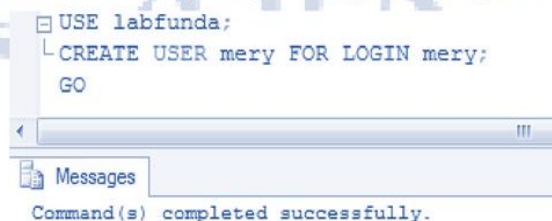
Syntaxnya :

```
USE database_name;  
CREATE USER user_name  
[  
    { FOR | FROM } LOGIN login_name  
]  
[;]
```

Contoh :

Gunakan login Sa. Membuat user “mery” dengan menggunakan login id “mery”

```
USE labfunda;  
CREATE USER mery FOR LOGIN mery;  
GO
```



- Memodifikasi User

Syntaxnya :

USE database_name;

ALTER USER userName WITH <set_item> [,...n] [;]

Contoh :

Mengubah user “mery” menjadi “meryl”

```
USE labfunda;  
ALTER USER mery WITH NAME = meryl;  
GO
```

Messages
Command(s) completed successfully.

4.3. Hak User

Definisinya adalah “Apa yang dapat dan tidak dapat dilakukan”. Hak User terbagi dalam 2 kategori :

1. Hak untuk melakukan aksi khusus kepada system di dalam database (System Privilege).
2. Hak untuk melakukan aksi khusus kepada object dalam database (Object Privilege).

4.3.1. Memberikan Izin Operasi System dalam Database

Tabel 4.1. System Privilege

System Privilege	Operasi yang diijinkan
CREATE DATABASE	Membuat database dalam shcema user
CREATE TABLE	Membuat tabel dalam shcema user
CREATE ROLE	Membuat role dalam schema user
CREATE VIEW	Membuat view dalam schema user

4.3.2. Memberikan Izin Operasi Object dalam Database

Hak user dalam objek terbagi dalam 6 tipe, keenam tipe tersebut dapat anda lihat pada tabel di bawah ini :

Tabel 4.2. Object Privilege

Object Privilege	Operasi yang diijinkan
SELECT	Memperbolehkan user untuk melihat data
INSERT	Memperbolehkan user untuk membuat data baru
UPDATE	Memperbolehkan user untuk merubah data yang telah ada
DELETE	Memperbolehkan user untuk menghapus data
REFERENCES	Memperbolehkan user untuk mengisi baris yang tabelnya memiliki foreign key yang berhubungan dengan tabel lain yang USER tidak boleh SELECT
EXECUTE	Memperbolehkan user untuk menjalankan Procedure yang telah tersimpan

4.4. GRANT

Syntax :


```
GRANT { ALL [ PRIVILEGES ] }  
    permission [ ( column [ ,...n ] ) ] [ ,...n ]  
    [ ON [ class :: ] securable ] TO principal [ ,...n ]  
    [ WITH GRANT OPTION ] [ AS principal ]
```

- ALL mengindikasikan bahwa user ingin memberikan semua hak yang tersedia untuk objek tersebut.
- ON mengindikasikan object yang ingin user berikan izin / hak.
- TO mengindikasikan kepada siapa user ingin memberikan akses tersebut, dapat berupa login ID atau nama role
- WITH GRANT OPTION memperoleh hak yang user berikan hak kepadanya untuk dapat juga memberikan hak kepada user lain

Contoh :

Memberikan hak akses untuk membuat tabel dan view pada mery

```
GRANT CREATE TABLE,CREATE VIEW to mery1;
GO
```



Messages
Command(s) completed successfully.

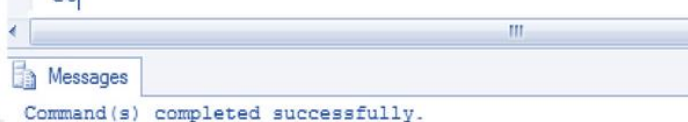
4.4.1. Menggunakan WITH GRANT OPTION

Hak akses yang diberikan menggunakan WITH GRANT OPTION dapat diteruskan ke user dan *role* lainnya.

Contoh :

Meryl dapat memberikan hak akses SELECT, INSERT, UPDATE, DELETE pada table EMP kepada user lain.

```
USE labfunda;
GO
GRANT SELECT, UPDATE, INSERT, DELETE on dbo.EMP
to mery1 with grant option;
GO
```



Messages
Command(s) completed successfully.

4.5. DENY

Syntax :

```
DENY { ALL [ PRIVILEGES ] }
      permission [ ( column [ ,...n ] ) ] [ ,...n ]
      [ ON [ class :: ] securable ] TO principal [ ,...n ]
      [ CASCADE ] [ AS principal ]
```

Jika user tidak menginginkan atau melarang sebuah user untuk mendapatkan akses, maka user dapat memasukkan perintah DENY (tentunya login dulu sebagai sa).

Contoh:

```
USE labfunda;
GO
DENY SELECT, UPDATE, INSERT, DELETE on dbo.EMP to mery1 CASCADE;
GO
```

Messages
Command(s) completed successfully.

Ket: CASCADE digunakan apabila user telah memiliki object.

4.6. REVOKE

Digunakan untuk menghilangkan efek dari perintah GRANT atau DENY yang telah ada sebelumnya.

Syntax :

```
REVOKE [ GRANT OPTION FOR ]
{
    [ ALL [ PRIVILEGES ] ]
    permission [ ( column [ ,...n ] ) ] [ ,...n ]
}
[ ON [ class :: ] securable ]
{ TO | FROM } principal [ ,...n ]
[ CASCADE ] [ AS principal ]
```

Contoh :

```
USE labfunda;
GO
REVOKE SELECT, UPDATE, INSERT, DELETE on dbo.EMP to mery1;
GO
```

Messages
Command(s) completed successfully.

4.7. ROLE

Merupakan koleksi dari hak akses yang data diberikan kepada user dengan mudah dengan mendaftarkan user kepada role tersebut.

- User – Defined Database Roles

Untuk membuat role user sendiri, dapat menggunakan CREATE ROLE yang syntaxnya :

```
CREATE ROLE role_name [ AUTHORIZATION owner_name ]
```

Contoh membuat role user sendiri (menggunakan login sa) :

```
CREATE ROLE Manager
```

Messages

Command(s) completed successfully.

Kemudian beri hak akses ke pada role manager :

```
USE labfunda;  
GRANT SELECT on dbo.emp to manager;  
GO
```

Messages

Command(s) completed successfully.

Dengan demikian maka setiap user yang berada pada role user akan mempunyai akses SELECT ke tabel emp (kecuali user tersebut memiliki DENY dari tempat lain dalam security informationnya).

- Menambahkan dan Menghapus User ke dalam Role

Syntaxnya :

```
execute sp_addrolemember RoleName, UserName
```

```
execute sp_droprolemember RoleName, UserName
```

Contoh :

```
exec sp_addrolemember manager,mery1;  
GO
```

Messages

Command(s) completed successfully.


```
exec sp_droprolemember manager, meryl;  
Go
```

Messages

Command(s) completed successfully.

- Dropping Role (menghapus role yang telah user buat)

Syntaxnya :

DROP ROLE role_name

```
Use labfunda;  
DROP ROLE MANAGER;  
Go
```

Messages

Command(s) completed successfully.