

Basis Data II

STT Terpadu NurulFikri - 2014

Sirojul Munir S.Si M.Kom
@rojulman



Manajemen User Database

**Basis Data II, Teknik Informatika / Sistem Informasi
STT Terpadu NurulFikri - 2013**

**Sirojul Munir S.Si M.Kom
@rojulman**

SQL: Structured Query Language

- ❑ **Data Definition Language (DDL)** – Perintah SQL untuk mendefinisikan skema database atau table : CREATE, DROP, ALTER
- ❑ **Data Manipulation Language (DML)** – Perintah SQL untuk manipulasi data dan pencarian data : INSERT, UPDATE, DELETE, SELECT
- ❑ **Data Control Language (DCL)** – Perintah SQL untuk mengontrol database atau table : GRANT, REVOKE

Kategori User Database

- **Database administrators**
- **Database designers**
- **End User**
- **System Analyst & Programmer**



Database Administrators



- ❑ **Memberikan otorisasi akses database**
 - Create Database
 - Create Roles, Privileges
 - Create User Account with Profiles
- ❑ **Mengkoordinasikan & Memonitoring**
 - Mengkoordinasikan penggunaan database
 - Memonitoring penggunaan sumber daya database yang dibutuhkan

Database Designers

- ❑ **Identifikasi data yang akan disimpan**
- ❑ **Memilih struktur data & menyimpan data**
- ❑ **Mendesain user yang akan mengakses database sesuai requirements**
- ❑ **Mendefinisikan views database sesuai requirements**
(akses user / group user)



End Users



- ❑ **Casual End Users** – occasionally access, may need different information each time. Use query language to specify requests.
- ❑ **Naïve or parametric end users** – main job is to query and update the database using standard queries and updates. These canned transactions have been carefully programmed and tested. Examples?
- ❑ **Sophisticated end users** – engineers, scientists, analysts who implement applications to meet their requirements.
- ❑ **Stand alone users** – maintain personal databases using ready made packages.

System Analyst dan Application Programmers

- ❑ Mendefinisikan End User akses database sesuai dengan requirement
- ❑ Mengembangkan spesifikasi transaksi yang akan digunakan sesuai dengan requirement
- ❑ Programmer aplikasi implementasi spesifikasi program, testing dan debugging



User Admin

#	Vendor Database	Admin User (default)
---	-----------------	----------------------

1	MySQL	root
---	-------	------

2	PostgreSQL	postgres
---	------------	----------

3	SQL Server	sa
---	------------	----

4	Oracle	Sys : user installer
---	--------	----------------------

		System : DBA
--	--	--------------



Administrasi User MySQL



User MySQL

- ❑ Secara default user admin dari server MySQL adalah *root*
- ❑ Informasi user tersimpan dalam database *mysql*

```
mysql> show databases
-> ;
+-----+
| Database |
+-----+
| information_schema |
| cdcol      |
| latihan1    |
| latihvien1db |
| mysql      | ←
| phpmyadmin  |
| test       |
| webauth    |
+-----+
8 rows in set (0.04 sec)

mysql> _
```

Table user MySQL

- Data user server mysql ada pada table user

```
mysql> use mysql ;
Database changed
mysql> show tables
-> ;
+-----+
| Tables_in_mysql |
+-----+
| columns_priv    |
| db              |
| func            |
| help_category   |
| help_keyword    |
| help_relation   |
| help_topic      |
| host            |
| proc            |
| procs_priv      |
| tables_priv     |
| time_zone       |
| time_zone_leap_second |
| time_zone_name  |
| time_zone_transition |
| time_zone_transition_type |
| user            |
+-----+
17 rows in set (0.01 sec)

mysql> _
```

Berisi data
User :
Host,
UserName,
Password

GRANT TABLES :

Lima Table izin akses user (1)

- **User** : berisi data user yang mendapatkan izin akses MySQL, asal koneksi dan izin akses kepada user

Tingkatan akses : global

- **Db** : Mengatur database apa saja yang dapat diakses seorang user dan jenis izin aksesnya

Tingkatan akses : Database

- **Host** : Mengatur asal host yang diperkenankan bagi user untuk mengakses MySQL, jika lebih dari satu host

Tingkatan akses : Database

GRANT TABLES :

Lima Table izin akses user (2)

- **Table priv** : Mengatur tabel apa saja yang dapat diakses oleh seorang user dan jenis izin aksesnya

Tingkatan akses : Tabel

- **columns_priv** : Mengatur kolom (field) apa saja yang dapat diakses oleh seorang user dan jenis izin aksesnya

Tingkatan akses : Kolom - field

User Privileges

- ❑ Izin akses user terbagi atas tiga bagian :
 1. Tingkatan akses user biasa
 2. Tingkatan akses administrator – global administrative
 3. Tingkatan akses khusus – special privileges

Reguler User Privileges

- Izin akses user untuk Tingkatan akses user biasa adalah :

**ALTER
CREATE
DELETE
DROP
INDEX
INSERT
SELECT
UPDATE
REFERENCES**

Administrator User Privileges

- ❑ Izin akses administrator – global administrative, hanya digunakan oleh user setingkat root atau administrator dan tidak diberikan kepada user biasa, yaitu :
 - ❑ **FILE**
 - PROCESS**
 - RELOAD**
 - SHUTDOWN**
 - CREATE TEMPORARY TABLE**
 - EXECUTE**
 - LOCK TABLES**
 - REPLICATION CLIENT**
 - REPLICATION SLAVE**
 - SHOW DATABASES**
 - SUPER**

Special Privileges

- Izin akses yang dapat diterapkan pada setiap user dengan izin akses sebagai berikut :

**ALL
USAGE**

Update password user root

□ Query melihat user

```
mysql> select host,user,password from user;
```

host	user	password
localhost	root	*2470C0C06DEE42FD1618BB99005ADCA2EC9D1E19
127.0.0.1	root	*2470C0C06DEE42FD1618BB99005ADCA2EC9D1E19
localhost	debian-sys-maint	*63453BC3CB9C8B242152B05400D73B1ED6575683
localhost	siswa	*0D52ED4E5F79A1C47E30BCF89A87A04D87985448
localhost	rojul	*4C0321E784283A27B260C38EF993A5BD0EC31577
localhost	phpmyadmin	*2470C0C06DEE42FD1618BB99005ADCA2EC9D1E19

□ Query update password user root :

```
mysql> UPDATE user  
      SET password=password('r4h4s14')  
      where user='root';
```

□ Lanjutkan dengan perintah berikut :

```
mysql> FLUSH PRIVILEGES;
```

- Fungsi Flush : mysql membaca grant tables hanya satu kali pada saat server pertama kali dijalankan, perintah flush akan memerintahkan kepada sistem untuk membaca ulang kelima grant tables tanpa harus merestart server mysql

Manajemen User

- Gunakan perintah GRANT untuk membuat user

Sintaksnya :

```
GRANT jenis_akses (nama_kolom) ON  
nama_database TO nama_user@alamat_komputer  
IDENTIFIED BY 'password_user' [With GRANT  
OPTION]
```

- Gunakan perintah REVOKE untuk menghapus user

```
REVOKE jenis_akses ON nama_database  
FROM nama_user
```

Membuat User (a)

❑ **Buat database latihan**

```
mysql> CREATE DATABASE latihan;
```

❑ **Buat user rosa pemilik database latihan, rosa hanya bisa akses database langsung di komputer server MySQL**

```
mysql> GRANT ALL PRIVILEGES ON latihan.* TO  
      rosa@localhost IDENTIFIED BY 'rosa';
```

❑ **Buat user faiz agar bisa mengakses dari komputer nomor IP 192.168.1.10 dengan privileges SELECT,INSERT dan UPDATE saja**

```
mysql> GRANT SELECT,INSERT,UPDATE ON latihan.* TO  
      faiz@192.168.1.10 IDENTIFIED BY 'faiz';
```

Membuat User (a)

❑ **Buat database latihan**

```
mysql> CREATE DATABASE latihan;
```

❑ **Buat user rosa pemilik database latihan, rosa hanya bisa akses database langsung di komputer server MySQL**

```
mysql> GRANT ALL PRIVILEGES ON latihan.* TO  
      rosa@localhost IDENTIFIED BY 'rosa';
```

❑ **Buat user faiz agar bisa mengakses dari komputer nomor IP 192.168.1.10 dengan privileges SELECT,INSERT dan UPDATE saja**

```
mysql> GRANT SELECT,INSERT,UPDATE ON latihan.* TO  
      faiz@192.168.1.10 IDENTIFIED BY 'faiz';
```

Membuat User (b)

- **Buat user alissa dan hanya boleh melakukan perintah SELECT pada table produk untuk field kode,nama,stok,harga pada dan hanya boleh melakukan UPDATE pada field stok dan harga saja dimana table produk terdapat pada database penjualan.**

```
mysql> GRANT SELECT  
      (kode,nama,stok,harga),INSERT (stok,harga) ON  
      penjualan.produk TO alissa@localhost  
      IDENTIFIED BY 'alissa';
```

Login User

- ❑ User rosa login ke database latihan langsung di server MySQL

```
$ mysql -u rosa latihan -p
```

- ❑ User faiz login ke database server MySQL (IP Server MySQL misalnya 192.168.1.1)

```
$ mysql -u faiz latihan -h 192.168.1.1 -p
```