

# Desain Pengelolaan Database

Sirojul Munir

[rojulman@nurulfikri.ac.id](mailto:rojulman@nurulfikri.ac.id) | @rojulman

# Pengelolaan Database

Manajemen User MySQL

[rojulman@nurulfikri.ac.id](mailto:rojulman@nurulfikri.ac.id) | @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 & Programmers



# 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/MariaDB	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 user,host,authentication_string from user;
```

user	host	authentication_string
root	localhost	
mysql.session	localhost	*THISISNOTAVALIDPASSWORDTHATCANBEUSEDHERE
mysql.sys	localhost	*THISISNOTAVALIDPASSWORDTHATCANBEUSEDHERE
rojul	localhost	*6BB4837EB74329105EE4568DDA7DC67ED2CA2AD9

# Update password user root

---

- ❑ Query update password user root :

```
mysql> UPDATE user  
      SET authentication_string=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';
```

```
mysql> FLUSH PRIVILEGES;
```

```
mysql> \q
```

- ❑ User rosa login ke database latihan

```
shell> mysql -u rosa latihan -p
```

# Membuat User (b)

---

- ❑ 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 (c)

---

- ❑ 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 (nik,nama,tmp_lahir,tgl_lahir),  
          UPDATE (gaji,jml_anak) ON dtu2016.pegawai  
          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
```

# Backup Database

---

- ❑ Perintah Backup database

```
mysqldump -u nama_user -p nama_database > data_backup.sql
```

- ❑ Mysqldump : command untuk backup database
- ❑ -u : username untuk akses ke database
- ❑ -p : password user
- ❑ -R : proses backup yang menyertakan ROUTINE( trigger, function, store procedure dsb)
- ❑ nama\_database : nama basis data
- ❑ Tanda > : output redirection ke file
- ❑ data\_backup.sql : nama file hasil backupnya

# Restore database

---

- ❑ Pastikan sudah memiliki file hasil backup (misal : data\_backup.sql)

- ❑ Buat database baru

```
mysql> CREATE DATABASE dtu2018_baru;
```

- ❑ Buat user pemilik database

```
mysql> GRANT ALL PRIVILEGES ON dbdtu2018_baru.* TO  
rosa@localhost IDENTIFIED BY 'rosa';
```

- ❑ Lakukan simpan ulang data erintah Backup database

```
mysql -u rosa -p dtu2018_baru < data_backup.sql
```

# LOCK TABLES (a)

---

- ❑ Berkenaan dengan penguncian sebuah table
- ❑ Contoh Kasus :

Sebuah database dapat digunakan oleh beberapa aplikasi, misalnya untuk internal aplikasi desktop sedangkan untuk eksternal berbasis web. Terkadang kedua aplikasi mengakses tabel secara bersamaan maka akan mungkin terjadi kesalahan data, contohnya klasik adalah tentang stok barang, aplikasi desktop sedang melakukan penambahan stok pada tabel produk, dan aplikasi web sedang melakukan penghitungan stok ketika proses penambahan belum selesai

# LOCK TABLES (b)

---

- ❑ Pada kasus stok barang ini bisa dihindari dengan menerapkan LOCK TABLES
- ❑ LOCK TABLES melakukan penguncian tabel ketika proses sedang berlangsung, kemudian akan membebaskan table (UNLOCK TABLES) ketika proses telah berakhir
- ❑ LOCK TABLES telah disupport oleh MySQL versi 5
- ❑ Ada dua jenis LOCK TABLES
  1. READ LOCK TABLES
  2. WRITE LOCK TABLES

# READ LOCK TABLES

---

- ❑ Penguncian table sehingga table hanya bisa dilakukan proses baca saja yang diterapkan pada semua session
- ❑ Jika terjadi manipulasi tabel seperti INSERT, UPDATE dan DELETE, truncate status nya akan selalu waiting hingga proses UNLOCK TABLES dilakukan
- ❑ Contoh :

```
mysql> LOCK TABLE produk READ;
```

```
mysql> update produk set stok=stok + 10 where idproduk=1;
```

```
ERROR 1099 (HY000): Table 'produk' was locked with a READ lock and  
can't be updated
```

# Contoh READ LOCK TABLES

---

- ❑ Contoh LOCK table produk dalam session yang sama :

```
mysql> LOCK TABLE produk READ;
```

```
mysql> update produk set stok=stok + 10 where idproduk=1;
```

```
ERROR 1099 (HY000): Table 'produk' was locked with a READ lock and can't  
be updated
```

- ❑ Jika dilakukan pada session lain maka perintah akan dieksekusi ketika sudah dijalankan UNLOCK TABLES;

```
mysql> update produk set stok=stok + 10 where idproduk=1;
```

**Proses diatas masih menunggu proses UNLOCK TABLES**

# WRITE LOCK TABLES

---

- ❑ Penguncian table sehingga table hanya bisa dilakukan proses SELECT,INSERT,UPDATE,DELETE pada session sendiri saja
- ❑ Jika terjadi manipulasi tabel seperti INSERT. UPDATE dan DELETE, truncate pada session lain status nya akan selalu waiting hingga proses UNLOCK TABLES dilakukan
- ❑ Session lain hanya bisa melakukan proses read atau SELECT saja



# Contoh WRITE LOCK TABLES

---

- Contoh LOCK table produk dalam session yang sama :

```
mysql> LOCK TABLE produk WRITE;
```

```
mysql> update produk set stok=stok + 10 where idproduk=1;
```

**Query update sukses dijalankan**

- Jika dilakukan pada session lain maka perintah akan dieksekusi ketika sudah dijalankan UNLOCK TABLES;

```
mysql> update produk set stok=stok + 10 where idproduk=1;
```

**Proses diatas masih menunggu proses UNLOCK TABLES**