

# MODUL I

## PENGENALAN MYSQL

### A. TUJUAN

- Mengetahui DBMS MySQL dan penggunaannya untuk pengelolaan basis data.
- Mampu memanfaatkan *command-line (client)* MySQL.
- Memahami operasi-operasi DDL dan DML di MySQL

### B. PETUNJUK

Awali setiap aktivitas dengan doa, semoga berkah dan mendapat kemudahan. Pahami tujuan, dasar teori, dan latihan-latihan praktikum dengan baik dan benar. Kerjakan tugas-tugas praktikum dengan baik, sabar, dan jujur. Tanyakan kepada asisten/dosen apabila ada hal-hal yang kurang jelas.

### C. DASAR TEORI

#### 1. MySQL

MySQL (dieja "My Ess Que Ell") merupakan perangkat lunak basis data yang dikembangkan oleh perusahaan bernama MySQL AB. Sebagai sebuah produk, ada beberapa point penting mengenai MySQL.

MySQL merupakan suatu *Database Management System (DBMS)*.

MySQL adalah suatu *Relational Database Management System (RDBMS)*

MySQL merupakan sebuah produk yang didistribusikan secara *open source*.

MySQL dapat berjalan baik di hampir semua *platform* sistem operasi.

#### 2. Server dan Client MySQL

Arsitektur DBMS MySQL dapat diklasifikasikan ke dalam dua jenis: *server* dan *client*. Secara garis besar, *client* digunakan untuk melakukan *request* informasi; *server* bertindak merespon permintaan dari *client*. Dengan demikian, lingkungan yang nantinya akan digunakan pada praktikum ini adalah *client*.

### 3. Objek Basis Data

Di MySQL, basis data direpresentasikan sebagai suatu direktori; di mana di dalamnya terdapat himpunan file-file tabel yang terdefinisi. Pembuatan basis data dilakukan menggunakan pernyataan `CREATE DATABASE` dan diikuti dengan nama basis data yang akan dibuat.

Bentuk sintaks pembuatan basis data:

```
CREATE DATABASE [IF NOT EXISTS] nama_database;
```

Oleh karena basis data merupakan suatu *schema*, maka penghapusannya dilakukan menggunakan perintah DDL `DROP DATABASE`.

Bentuk formal penghapusan basis data:

```
DROP DATABASE [IF EXISTS] nama_database
```



Untuk dapat bekerja dengan sebuah basis data yang terdefinisi, maka basis data tersebut harus dipilih terlebih dahulu. Langkah untuk mengindikasikan *current database* ini dilakukan melalui perintah `\u` atau `USE`.

### 4. Objek Tabel

MySQL mendukung penuh standar ANSI/ISO SQL. Dengan demikian, perintah pembuatan tabel di MySQL juga sama seperti DBMS lainnya.

Bentuk sintaks pembuatan tabel diperlihatkan sebagai berikut:

```
CREATE [TEMPORARY] TABLE [IF NOT EXISTS] nama_tabel  
[(create_definisi,...)]  
[table_options] [select_statement]
```

Untuk menghapus tabel yang terdefinisi, kita dapat memanfaatkan perintah `DROP`.

```
DROP TABLE [IF EXISTS] nama_tabel
```

Adapun untuk operasi-operasi modifikasi—baik modifikasi tabel maupun field—dapat dilakukan dengan memanfaatkan perintah `ALTER`.

### 5. Akses dan Manipulasi Data

Operasi-operasi akses dan manipulasi data—mencakup pengambilan atau seleksi, penambahan, perubahan, dan penghapusan—di MySQL juga didasarkan pada standar SQL. Dengan demikian, adanya standarisasi ini memungkinkan keseragaman operasi di lintas DBMS.

## D. LATIHAN

### 1. Mengetahui MySQL-Client

Secara default, paket instalasi MySQL menyertakan program `mysql-client`—bernama `mysql`—dalam bentuk *command-line*. Bergantung pada konfigurasi, `mysql-client` dapat diakses menggunakan atau tanpa *account* (nama dan password). Di praktikum ini, semua akses diharapkan melalui *account* yang telah disediakan.

Berikut langkah-langkah untuk masuk ke `mysql-client`.

1. Buka terminal (melalui menu **Applications > Terminal**).
2. Jalankan `mysql-client` dengan mengetikkan perintah berikut:

```
$ mysql -u userXXX -p
Enter Password: userXXX
```

Keterangan:

Sesuaikan notasi `xxx` dengan identitas user; misalnya untuk `user001` maka nama dan password-nya juga `user001`.

3. Jika *account* valid, akan terlihat tampilan prompt `mysql`.

```
Welcome to the MySQL monitor. Commands end with ; or \g.
Your MySQL connection id is 1
Server version: 5.0.67-community-nt MySQL Community Edition
(GPL)

Type 'help;' or '\h' for help. Type '\c' to clear the buffer.

mysql>
```

4. Untuk keluar dari prompt `mysql`, ketikkan `\q` atau **quit** atau **exit**.

```
mysql> \q
Bye
```

### 2. Operasi Database

Menciptakan Database

1. Buat database dengan nama sesuai NIM Anda dan awali dengan garis bawah “\_”.

```
mysql> CREATE DATABASE _207533408615;
Query OK, 1 row affected (0.00 sec)
```

2. Untuk memilih database sebagai database aktif, gunakan perintah **use**.

```
mysql> USE _207533408615
Database Changed
```

3. Apabila Anda ingin mendapatkan nama database yang aktif, gunakan perintah berikut:

```
mysql> SELECT DATABASE();
```

4. Untuk menampilkan daftar nama database di server, gunakan perintah SHOW.

```
mysql> SHOW DATABASES;
+-----+
| Database |
+-----+
| Information_schema |
| _207533408615 |
| test |
+-----+
3 row in set (0.00 sec)
```

5. Untuk menghapus database, gunakan perintah DROP yang diikuti nama database.

```
mysql> DROP DATABASE _207533408615;
Query OK, 1 row affected (0.00 sec)
```



Perhatikan, setiap pernyataan SQL harus diakhiri dengan titik koma (atau \g); sedangkan perintah mysql-client tidak perlu diakhiri titik koma.

### 3. Eksekusi File SQL

Untuk efisiensi, mysql-client mengizinkan eksekusi pernyataan-pernyataan SQL di suatu file.

1. Buka editor teks, kemudian ketikkan perintah berikut:

```
SHOW DATABASES;
```

2. Simpan file SQL, misalnya di lokasi **/home/latihan/show\_db.sql**.
3. Dari prompt MySQL, eksekusi file SQL dengan mengetikkan perintah berikut:

```
mysql> \. /home/latihan/show_db.sql
```

### 4. Operasi Tabel

#### Menciptakan Tabel

1. Buat file SQL baru, kemudian tuliskan pernyataan SQL berikut:

```
CREATE TABLE mahasiswa (
  nim CHAR(12) NOT NULL,
  nama VARCHAR(60) NOT NULL,
  jenis_kelamin CHAR(1) NOT NULL,
  alamat VARCHAR(120),
  PRIMARY KEY(nim)
);
```

2. Simpan dengan nama **create\_mahasiswa.sql**.
3. Eksekusi file SQL.

## Melihat Daftar dan Struktur Tabel

1. Untuk mendapatkan daftar nama tabel di database, gunakan perintah SHOW.

```
mysql> SHOW TABLES;
```

2. Apabila Anda ingin melihat struktur tabel, gunakan perintah DESC atau DESCRIBE.

```
mysql> DESCRIBE mahasiswa;
```

| Field         | Type         | Null | Key | Default | Extra |
|---------------|--------------|------|-----|---------|-------|
| nim           | int(12)      | NO   | PRI | NULL    |       |
| nama          | varchar(60)  | NO   |     | NULL    |       |
| jenis_kelamin | char(1)      | NO   |     | NULL    |       |
| alamat        | varchar(120) | YES  |     | NULL    |       |

```
4 rows in set (0.00 sec)
```

```
mysql>
```

## Modifikasi Tabel

Sintaks operasi modifikasi nama tabel diperlihatkan sebagai berikut:

```
ALTER TABLE nama_tabel_lama RENAME nama_tabel_baru
```

Contoh berikut akan mengubah tabel **mahasiswa** menjadi **mhs**.

```
ALTER TABLE mahasiswa RENAME mhs;
```

## 5. Mengelola Field

Berkaitan dengan eksistensi *field* (kolom), ada beberapa operasi yang bisa diberikan.

### Menambah Field Baru

Sintaks operasi penambahan field baru diperlihatkan sebagai berikut:

```
ALTER TABLE nama_tabel ADD nama_field tipe_data [FIRST | AFTER nama_field]
```

Contoh berikut akan menambah field **no\_tlp** dan diletakkan di akhir tabel.

```
ALTER TABLE mahasiswa ADD no_tlp INT(12);
```

### Mengubah Nama Field

Sintaks operasi modifikasi nama field diperlihatkan sebagai berikut:

```
ALTER TABLE nama_tabel CHANGE nama_field_lama nama_field_baru tipe_data
```

Contoh berikut akan mengubah field **no\_tlp** menjadi **no\_hp**.

```
ALTER TABLE mahasiswa CHANGE no_tlp no_hp INT(12);
```

### Menghapus Field

Sintaks operasi penghapusan field diperlihatkan sebagai berikut:

```
ALTER TABLE nama_tabel DROP COLUMN nama_field
```

Contoh berikut akan menghapus field **no\_hp** dari tabel.

```
ALTER TABLE mahasiswa DROP COLUMN no_hp;
```

## 6. Manipulasi Data

Operasi penambahan, pengubahan, dan penghapusan data di MySQL mengacu pada standar SQL. Penjelasan detail mengenai operasi-operasi ini akan dibahas di modul selanjutnya.

### E. TUGAS PRAKTIKUM

1. Buat tabel matakuliah yang memiliki struktur tabel terdiri dari field (kode matakuliah, nama matakuliah, dan sks). Lihat hasil struktur tabel yang terbentuk.
2. Tambahkan field 'js' pada tabel matakuliah yang telah anda buat. Lihat hasil struktur tabel yang terbentuk.
3. Tambahkan field 'nomer' pada tabel mahasiswa yang telah anda buat, dimana field "nomor" berada pada posisi field yang pertama. Lihat hasil struktur tabel yang terbentuk.
4. Tambahkan field 'tanggal\_lahir' pada tabel mahasiswa, dimana field "tanggal\_lahir" berada pada posisi setelah field nama. Lihat hasil struktur tabel yang terbentuk.