



## Topik

Dasar MySQL dan Data Definition Language (DDL)

## Tujuan

Mahasiswa diharapkan dapat:

1. Memahami dasar-dasar MySQL
2. Membuat database dan tabel dengan menerapkan data definition language (DDL)
3. Memahami penggunaan perintah-perintah untuk mendefinisikan atribut-atribut database, tabel, field, maupun batasan-batasan terhadap suatu atribut dan hubungan antar tabel

## Pendahuluan

**Basis Data** : Himpunan kelompok data (arsip) yang saling berhubungan, yang diorganisasi sedemikian rupa, sehingga kelak dapat dimanfaatkan kembali dengan cepat.

### **Bahasa Basis Data**

DBMS merupakan perantara antara user dengan database. Cara komunikasi diatur dalam suatu bahasa khusus yang ditetapkan oleh DBMS. Misalnya SQL, dBase, QUEL, dsb.

Ada beberapa bentuk bahasa sql, namun yang biasa digunakan adalah 2 bentuk bahasa sql yaitu DDL dan DML

Data Definition Language (DDL) >> digunakan dalam membuat tabel baru, indeks, mengubah tabel, menentukan struktur tabel, dsb

Data Manipulation Language (DML) >> digunakan dalam memanipulasi dan pengambilan data pada database. Manipulasi data dapat mencakup :

- Pemanggilan data yang tersimpan dalam database (query)
- Penyisipan/penambahan data baru ke database
- Penghapusan data dari database
- Perubahan data pada database

### **Dasar-Dasar MySQL**

Dalam bahasa SQL pada umumnya informasi tersimpan dalam tabel-tabel yang secara logik merupakan struktur dua dimensi terdiri dari baris (row atau record) dan kolom (column atau field). Sedangkan dalam sebuah database dapat terdiri dari beberapa table. Beberapa tipe data dalam MySQL yang sering dipakai:

Tipe data	Keterangan
INT(M) [UNSIGNED]	Angka -2147483648 s/d 2147483647
FLOAT(M,D)	Angka pecahan
DATE	Tanggal Format : YYYY-MM-DD
DATETIME	Tanggal dan Waktu Format : YYYY-MM-DD HH:MM:SS
CHAR(M)	String dengan panjang tetap sesuai dengan yang ditentukan. Panjangnya 1-255 karakter
VARCHAR(M)	String dengan panjang yang berubah-ubah sesuai dengan yang disimpan saat itu. Panjangnya 1 – 255 karakter
BLOB	Teks dengan panjang maksimum 65535 karakter
LOB	Teks dengan panjang maksimum 4294967295 karakter

Data Definition Language (DDL) merupakan sekumpulan set perintah yang bertujuan untuk mendefinisikan atribut-atribut database, tabel, atribut kolom (field), maupun batasan-batasan terhadap suatu atribut dan relasi/ hubungan antar tabel. Yang termasuk dalam kelompok perintah DDL adalah :

- Create
- Alter
- Drop

**CREATE** merupakan perintah DDL yang digunakan untuk membuat database maupun tabel. Nama database maupun tabel tidak boleh mengandung spasi (space). Nama database tidak boleh sama antar database.

**ALTER** merupakan perintah DDL yang digunakan untuk mengubah nama/struktur tabel

**DROP** merupakan perintah DDL yang digunakan untuk menghapus database ataupun tabel

#### a. Database

Perintah untuk membuat database sebagai berikut :

- **CREATE DATABASE namadatabase;**

Contoh :

**CREATE DATABASE db\_polinema;**

Untuk menampilkan daftar nama database yang terdapat dalam database server pada MySQL menggunakan perintah :

- **SHOW DATABASES;**

Sebelum membuat suatu tabel yang akan digunakan untuk menyimpan data, terlebih dahulu harus memilih/mengaktifkan salah satu database sebagai database aktif yang akan digunakan untuk menyimpan beberapa tabel yang akan dibuat. Untuk memilih/mengaktifkan salah satu database menggunakan perintah berikut :

- **USE namadatabase;**

Sebagai contoh, database db\_polinema yang akan digunakan / diaktifkan maka perintahnya sebagai berikut :

**USE db\_polinema;**

Perintah yang berfungsi untuk menghapus database maupun tabel adalah perintah DROP. Untuk menggunakan perintah tersebut, berikut penulisan perintahnya :

- **DROP DATABASE namadatabase;**

Misalnya database db\_polinema yang akan dihapus maka perintahnya sebagai berikut :

**DROP DATABASE db\_polinema;**

#### b. Tabel

- Membuat Tabel

Nama tabel tidak boleh mengandung spasi (space). Ketika membuat tabel, ada beberapa yang harus dideklarasikan dalam pembuatannya yaitu antara lain meliputi : nama tabel, nama kolom (field),

tipe data dari field dan panjang data. Perintah yang digunakan untuk membuat tabel secara umum adalah sebagai berikut :

- **CREATE TABLE namatabel (field1 typedata1, field2 typedata2);**

Contoh berikut ini adalah syntax untuk membuat tabel mahasiswa :

```
CREATE TABLE mahasiswa (nim CHAR (20),nama_mhs CHAR(50),umur INT, PRIMARY KEY(nim));
```

- Menampilkan tabel

Untuk menampilkan daftar nama tabel yang terdapat dalam database yang sedang aktif/digunakan menggunakan perintah :

- **SHOW TABLES;**

- Menampilkan deskripsi atribut tabel

Untuk menampilkan deskripsi atribut-atribut yang terdapat pada suatu tabel dengan menggunakan perintah :

- **DESC namatabel;**

Contoh :

```
DESC mahasiswa;
```

- Menghapus Tabel

Untuk menghapus Tabel perintahnya sama dengan untuk menghapus database yaitu dengan menggunakan perintah DROP. Perintah yang digunakan adalah :

- **DROP TABLE namatabel;**

Misalnya yang akan dihapus adalah tabel mahasiswa :

```
DROP TABLE mahasiswa;
```

- Mengisi Data

Untuk memasukkan sebuah baris (record) kedalam tabel MySQL adalah sebagai berikut :

- **INSERT INTO table [(column1, column2,...) VALUES (value1, value2,...)];**

Contoh :

```
Insert into mahasiswa (nim,nama,umur) values ('001','imam',20);
```

- Mendefinisikan Null/Not Null

Null ataupun Not Null merupakan pernyataan yang digunakan untuk membuat kolom yang akan dibuat boleh kosong (Null) atau tidak boleh kosong (Not Null). Ketika pada kolom tabel tidak di set, maka secara default akan bernilai Null (boleh kosong). Untuk mendefinisikannya maka perintah yang akan digunakan adalah :

```
CREATE TABLE mahasiswa (nim CHAR (20) NOT NULL, nama_mhs CHAR (50) NOT NULL, umur INT, PRIMARY KEY (nim));
```

- Mendefinisikan Nilai Default

Nilai default merupakan nilai yang diberikan secara otomatis oleh sistem untuk suatu kolom ketika terjadi penambahan baris baru, sementara nilai pada kolom tersebut tidak diisi oleh pengguna.

Contoh :

```
CREATE TABLE mahasiswa (nim CHAR (20), nama-mhs CHAR (50), umur INT DEFAULT 0, PRIMARY KEY (nim));
```

- Mendefinisikan PRIMARY KEY pada Tabel

Suatu keharusan dalam suatu tabel adalah harus memiliki satu kolom yang dijadikan sebagai perwakilan dari tabel tersebut. Pembuatan perwakilan tabel ini berfungsi untuk melakukan

hubungan / relasional dengan tabel lain. Bentuk perwakilan ini dalam database disebut sebagai PRIMARY KEY yang aturan pembuatannya adalah sebagai berikut :

- Satu tabel bisa memiliki 2 primary key
- Nama kolom kunci tidak digunakan pada kolom lain satu tabel
- Nama kolom kunci tidak boleh sama dengan kolom kunci yang ada pada tabel lain
- Bentuk kolom kunci harus diset NOT NULL

Terdapat tiga cara untuk mendefinisikan primary key. Berikut ini perintah yang digunakan :

**CREATE TABLE mahasiswa (nim CHAR (20), nama\_mhs CHAR (50), umur INT, PRIMARY KEY (nim));**

Atau

**CREATE TABLE mahasiswa (nim CHAR (20) NOT NULL PRIMARY KEY, nama\_mhs CHAR (50), umur INT);**

Atau

**ALTER TABLE mahasiswa ADD CONSTRAINT namaconstraint PRIMARY KEY(namakolom);**

- Menghapus PRIMARY KEY pada tabel

Cara 1: Jika primary key dibuat menggunakan alter table:

**ALTER TABLE namatabel DROP CONSTRAINT namaconstraint;**

Cara 2: Jika primary key dibuat melalui create table:

**ALTER TABLE namatabel DROP PRIMARY KEY;**

- Menambah kolom baru pada tabel

Untuk menambah kolom baru pada suatu tabel bisa menggunakan perintah sebagai berikut :

**ALTER TABLE namatabel ADD fieldbaru typedata(lebar);**

Namatabel merupakan nama tabel yang akan ditambahkan kolomnya. Filedbaru merupakan nama kolom yang akan ditambahkan, typedata(lebar) merupakan type data dan lebar data yang akan ditambahkan. Contohnya menambahkan kolom telepon pada tabel mahasiswa setelah kolom umur:

**ALTER TABLE mahasiswa ADD COLUMN telepon CHAR (15) AFTER umur;**

- Mengubah Tipe Data atau Lebar Kolom pada Tabel

Perintah yang digunakan adalah :

**ALTER TABLE namatabel MODIFY COLUMN field type(lebar);**

Contoh :

**ALTER TABLE mahasiswa MODIFY COLUMN telepon(12);**

- Mengubah Nama Kolom(Field)

Perintah yang digunakan adalah :

**ALTER TABLE namatabel CHANGE COLUMN namakolomlama namakolombaru typedatabaru(lebarbaru);**

Contoh :

**ALTER TABLE mahasiswa CHANGE COLUMN telepon phone CHAR(25);**

- Menghapus Kolom pada Tabel

Perintah yang digunakan adalah :

**ALTER TABLE namatabel DROP COLUMN namakolom;**

Contoh :

**ALTER TABLE mahasiswa DROP COLUMN phone;**

- Mendefinisikan Foreign Key

Foreign Key adalah salah satu jenis constraint yang digunakan untuk merelasikan antar dua tabel atau lebih. Foreign Key digunakan pada tabel kedua (detail) yang mereferensi ke tabel utama yang mempunyai constraint primary key.



Pada gambar diatas, kolom **Cust\_ID** pada tabel **Orders** berperan sebagai foreign key yang mereferensi ke tabel **Customers** pada kolom **ID**.

**ALTER TABLE namatabel ADD FOREIGN KEY namafield REFERENCES namatabelreferensi (namafieldreferensi);**

## PRAKTIKUM

Langkah	Keterangan
1.	Buka prompt jalankan perintah berikut ini : C:\>Program Files\xampp\mysql\bin>mysql -u root -p (enter)
	<pre> Yushintias-MacBook-Pro:~ YushintiaPramitarini\$ mysql -u root -p Enter password: Welcome to the MySQL monitor.  Commands end with ; or \g. Your MySQL connection id is 17 Server version: 5.7.15 Homebrew  Copyright (c) 2000, 2016, Oracle and/or its affiliates. All rights reserved.  Oracle is a registered trademark of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.  Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.  mysql&gt; </pre>
2.	Buatlah sebuah database dengan nama db_polinema

```
mysql> create database db_polinema;
Query OK, 1 row affected (0.01 sec)

mysql> show databases;
+-----+
| Database |
+-----+
| information_schema |
| db_polinema |
| mysql |
| performance_schema |
| sys |
+-----+
5 rows in set (0.00 sec)
```

Buatlah beberapa tabel dalam database tersebut sesuai dengan kriteria berikut :

a. Tabel **Mahasiswa**

Field	Type Data
	Primary Key
mhs	50)
kelamin	('L','P') DEFAULT 'L'
t	ar (50)
	ar (20) DEFAULT 'MALANG'
ma	30)
	ar (12)
prodi	6) foreign key fk0 (kode_prodi) references prodi (kode_prodi)

```
mysql> create table mahasiswa (nim int(8) primary key, nama_mhs char(50), jenis_kelamin enum('L','P') default 'L', alamat varchar(50), kota varchar(20) default 'malang', asal_sma char(30), no_hp varchar(12), umur integer, kode_prodi char(6), foreign key fk0 (kode_prodi) references prodi(kode_prodi));
Query OK, 0 rows affected (0.01 sec)
```

b. Tabel **Prodi**

Field	Type Data
prodi	6) Primary Key
prodi	30)

```
mysql> create table prodi (kode_prodi char(6) primary key, nama_prodi char(30))
Query OK, 0 rows affected (0.02 sec)
```

c. Tabel **mata\_kuliah**

Field	Type Data
	10) Primary Key
_mk	50)
_jam	4,2)
	r

	<pre>mysql&gt; create table mata_kuliah (mk_id char(10) primary key, nama_mk char(50), jumlah_jam float(4,2), sks integer); Query OK, 0 rows affected (0.02 sec)</pre>												
6.	<p>d. Tabel ruang</p> <table> <tr> <th>Field</th><th>Type Data</th></tr> <tr> <td>id</td><td>3) Primary Key</td></tr> <tr> <td>ruang</td><td>20)</td></tr> <tr> <td>tas</td><td>r</td></tr> </table>	Field	Type Data	id	3) Primary Key	ruang	20)	tas	r				
Field	Type Data												
id	3) Primary Key												
ruang	20)												
tas	r												
	<pre>mysql&gt; create table ruang (ruang_id char(3) primary key, nama_ruang char(20), kapasitas integer); Query OK, 0 rows affected (0.02 sec)</pre>												
7.	<p>e. Tabel dosen</p> <table> <tr> <th>Field</th><th>Type Data</th></tr> <tr> <td></td><td>r (20) Primary Key</td></tr> <tr> <td>dosen</td><td>50)</td></tr> <tr> <td></td><td>('PNS','KONTRAK') Default 'PNS'</td></tr> <tr> <td>kelamin</td><td>('L','P') Default 'L'</td></tr> <tr> <td></td><td>ar (15)</td></tr> </table>	Field	Type Data		r (20) Primary Key	dosen	50)		('PNS','KONTRAK') Default 'PNS'	kelamin	('L','P') Default 'L'		ar (15)
Field	Type Data												
	r (20) Primary Key												
dosen	50)												
	('PNS','KONTRAK') Default 'PNS'												
kelamin	('L','P') Default 'L'												
	ar (15)												
	<pre>mysql&gt; create table dosen (nidn integer(20) primary key, nama _dosen char(50), status enum ('PNS','KONTRAK') default 'PNS', jenis_kelamin enum ('L','P') default 'L', no_hp varchar(15)); Query OK, 0 rows affected (0.02 sec)</pre>												
8.	<p>&lt;Soal&gt;  Tambahkan sebuah kolom agama (varchar(10)) pada tabel mahasiswa sebagai kolom terakhir  <b>Catat : Buat Screenshot dari perintah yang anda ketikkan</b></p>												
9.	<p>&lt;Soal&gt;  Tambahkan kolom alamat(varchar(50)) pada tabel dosen sebagai kolom terakhir  <b>Catat : Buat Screenshot dari perintah yang anda ketikkan</b></p>												
10.	<p>&lt;Soal&gt;  Lakukan insert data ke dalam tabel-tabel yang ada pada database db_polinema sesuai dengan field, tipe data dan panjang datanya  <b>Catat : Buat Screenshot dari perintah yang anda ketikkan</b></p>												
11.	<p>&lt;Soal&gt;  Tampilkan semua tabel yang ada didalam database db_polinema  <b>Catat : Buat Screenshot dari perintah yang anda ketikkan</b></p>												
12.	<p>&lt;Soal&gt;  Tampilkan semua isi tabel yang ada didalam tabel mahasiswa</p>												

	Catat : Buat Screenshot dari perintah yang anda ketikkan
13.	<b>&lt;Soal&gt;</b> Tampilkan struktur(metadata) tabel mahasiswa Catat : Buat Screenshot dari perintah yang anda ketikkan
14.	<b>&lt;Soal&gt;</b> hilangkan kolom asal_sma yang terdapat didalam tabel mahasiswa Catat : Buat Screenshot dari perintah yang anda ketikkan

## Tugas

### 1. Buatlah basis data akademik dengan data sebagai berikut :

Hari	Jam Ke	Kode MK	Nama MK	Kode Dosen	Nama Dosen	Kls	Ruang
Senin	1-3	EL230	Fisika 1	105	Prof. Bajuri	A, B, C	AMP1
Senin	4-5	EL230	Fisika 2	105	Prof. Bajuri	C, D	AMP1
Selasa	1-3	EL440	Pemrograman	102	Susilowati	A, B	R1
Selasa	4-6	EL440	Pemrograman	105	Prof. Bajuri	C, D	R2
Selasa	1-3	EL450	Pancasila	109	Timbul, PhD.	E, F	R1

#### a. deskripsikan struktur data dari table-table berikut serta isikan datanya:

Tabel mata\_kuliah {kode\_mk, nama\_mk}

Tabel dosen {kd\_dosen,nama\_dosen}

Tabel jadual {kode\_mk,hari,jam\_ke,kelas,ruang}

#### b. tambahkan kolom alamat\_dosen pada tabel dosen di kolom terakhir

#### c. tambahkan kolom sks pada tabel mata\_kuliah

#### d. tambahkan kolom kd\_dosen dari tabel dosen kedalam tabel\_mata\_kuliah serta berikanlah kunci foreign key

#### e. tampilkan semua data yang ada pada tiap tabel

### 2. Buatlah basis data penjualan yang terdiri dari tabel sebagai berikut :



Nama Mobil	Tipe Mobil	Harga Mobil	Jumlah Penjualan	Jenis Mobil
Toyota Alphard	2.4 G	Rp906.000.000	2	Mobil Keluarga
Toyota Alphard	3.5 G	Rp1.112.000.000	78	Mobil Keluarga
Toyota Avanza	1.3 G A/T	Rp158.050.000	22	Mobil Keluarga
Toyota Camry	2.4 G A/T LUX	Rp462.200.000	21	Sedan
Toyota Camry	3.5 Q A/T	Rp647.000.000	43	Sedan
Toyota Corolla Altis	1.8 G A/T	Rp346.700.000	10	Sedan
Toyota Corolla Altis	2.0 V A/T	Rp375.500.000	14	Sedan
Toyota Dyna	130 HT	Rp220.000.000	31	Mini Truk
Toyota Fortuner	2.5 G M/T	Rp363.700.000	11	SUV
Toyota Fortuner	2.7 G Lux A/T	Rp421.300.000	54	SUV
Toyota Fortuner	2.7 V A/T	Rp470.300.000	5	SUV
Toyota Hilux	D Cab E	Rp302.500.000	4	Pick Up
Toyota Kijang Innova	E M/T Gasoline	Rp210.000.000	21	Mobil Keluarga
Toyota Kijang Innova	G M/T Luxury Gasoline	Rp230.600.000	11	Mobil Keluarga
Toyota Kijang Innova	J M/T Gasoline	Rp181.400.000	21	Mobil Keluarga

a. Deskripsikan struktur data dari table-table berikut serta isikan datanya:

Transaksi {notransaksi\_kode\_mobil\_jml\_jual}

Mobil {kode\_mobil,nama\_mobil,tipe\_mobil,harga,jenis\_mobil}

b. Tambahkan kolom tgl\_transaksi pada tabel transaksi di kolom terakhir

c. Tambahkan kolom warna\_mobil pada tabel mobil di kolom terakhir

d. Tampilkan semua data yang ada pada tiap tabel

-- Selamat Mengerjakan --