



Jurusan Teknologi Informasi Politeknik Negeri Malang
Jobsheet-10: MySQL (Data Definition Language (DDL))
Mata Kuliah Basis Data

Pengampu: Tim Ajar Basis Data

April 2021

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
LONGBLOB	Teks dengan panjang maksimum 4294967295 karakter

Data Definition Language (DDL)

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** → 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

Penerapan DDL pada :

A. DATABASE

- **CREATE DATABASE nama_database;**
Merupakan perintah untuk membuat database baru, contoh:
CREATE DATABASE db_polinema;
- **SHOW DATABASES;**
Merupakan perintah untuk menampilkan daftar nama database yang terdapat dalam database server.
- **USE nama_database;**
Merupakan perintah untuk masuk kedalam database yang akan digunakan. Sebelum membuat tabel yang digunakan untuk menyimpan data, terlebih dahulu harus memilih/mengaktifkan satu database sebagai database aktif. Contoh untuk dapat menggunakan/manipulasi database *db_polinema*, maka gunakan perintah berikut:
USE db_polinema;
- **DROP DATABASE namadatabase;**
Merupakan perintah yang berfungsi untuk menghapus database. Contoh kita ingin menghapus database *db_polinema*, maka perintahnya adalah:
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 VARCHAR(20), nama_mhs VARCHAR(50), umur INT);
```

➤ Menampilkan tabel

Untuk menampilkan daftar nama-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 nama_tabel;
```

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 / Input Data ke dalam tabel

Untuk memasukkan sebuah baris (*tuple/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 VARCHAR(20) NOT NULL, nama_mhs VARCHAR(50) NOT NULL, umur INT);
```

➤ 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 VARCHAR(20) NOT NULL, nama_mhs VARCHAR(50) NOT NULL, umur INT DEFAULT 0);
```

➤ 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 :

1. **Primary key** dibuat saat **create table**, dan didefinisikan di akhir

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

2. **Primary key** dibuat saat **create table**, dan didefinisikan bersamaan dengan kolom/field

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

3. **Primary key** dibuat menggunakan **alter table**

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

```
ALTER TABLE mahasiswa ADD CONSTRAINT PK_Mahasiswa PRIMARY KEY(nim);
```

➤ Menghapus **PRIMARY KEY** pada tabel

» Cara 1: Jika **primary key** dibuat menggunakan **create table**:

```
ALTER TABLE nama_table DROP PRIMARY KEY;
```

Contoh

```
ALTER TABLE mahasiswa DROP PRIMARY KEY;
```

» Cara 2: Jika **primary key** dibuat melalui **alter table**:

```
ALTER TABLE namatabel DROP CONSTRAINT namaconstraint;
```

Contoh

```
ALTER TABLE mahasiswa DROP CONSTRAINT PK_Mahasiswa;
```

- Menambah kolom baru pada tabel

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

```
ALTER TABLE nama_tabel ADD field_baru typedata(Lebar);
```

nama_tabel merupakan nama tabel yang akan ditambahkan kolomnya. *field_baru* 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 VARCHAR(15) AFTER umur;
```

- Mengubah Tipe Data atau Lebar Kolom pada Tabel

Perintah yang digunakan adalah :

```
ALTER TABLE nama_tabel MODIFY COLUMN field type(Lebar);
```

Contoh :

```
ALTER TABLE mahasiswa MODIFY COLUMN telepon VARCHAR(12);
```

- Mengubah Nama Kolom(Field)

Perintah yang digunakan adalah :

```
ALTER TABLE nama_tabel CHANGE COLUMN nama_kolom_Lama nama_kolom_baru typedatabaru(Lebarbaru);
```

Contoh :

```
ALTER TABLE mahasiswa CHANGE COLUMN telepon phone VARCHAR(25);
```

- Menghapus Kolom pada Tabel

Perintah yang digunakan adalah :

```
ALTER TABLE nama_tabel DROP COLUMN nama_kolom;
```

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 tersebut kolom **Cust_ID** pada tabel **Orders** berperan sebagai *foreign key* yang mereferensi ke tabel **Customers** pada kolom **ID**.

ALTER TABLE nama_tabel ADD FOREIGN KEY nama_kolom REFERENCES nama_tabel_referensi (nama_kolom_referensi);

PRAKTIKUM

No	Keterangan						
1	<p>Buka prompt jalankan perintah berikut ini :</p> <p>C:\>Program Files\xampp\mysql\bin>mysql -u root -p (enter)</p> <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> </pre>						
2	<p>Buatlah sebuah database dengan nama db_polinema</p> <pre> 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) </pre>						
3	<p>a. Tabel prodi</p> <table border="1"> <thead> <tr> <th>Field</th><th>Type Data</th></tr> </thead> <tbody> <tr> <td>kode_prodi</td><td>VARCHAR (6) PRIMARY KEY</td></tr> <tr> <td>nama_prodi</td><td>VARCHAR (30)</td></tr> </tbody> </table> <pre> mysql> create table prodi (kode_prodi char(6) primary key,nama_prodi char(30)); Query OK, 0 rows affected (0.02 sec) </pre>	Field	Type Data	kode_prodi	VARCHAR (6) PRIMARY KEY	nama_prodi	VARCHAR (30)
Field	Type Data						
kode_prodi	VARCHAR (6) PRIMARY KEY						
nama_prodi	VARCHAR (30)						

Buatlah beberapa tabel dalam database tersebut sesuai dengan kriteria berikut :

b. Tabel *mahasiswa*

Field	Type Data
nim	INT (8) PRIMARY KEY
nama_mhs	VARCHAR (50)
jenis_kelamin	ENUM ('L','P') DEFAULT 'L'
alamat	VARCHAR (50)
kota	VARCHAR (20) DEFAULT 'MALANG'
asal_sma	VARCHAR (30)
no_hp	VARCHAR (12)
umur	INT
kode_prodi	VARCHAR (6) FOREIGN KEY fk0 (kode_prodi) REFERENCES prodi (kode_prodi)

4

Untuk DBMS MySQL

```
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)
```

Untuk DBMS MariaDB

```
MariaDB [db_polinema]> create table mahasiswa (nim int(8) primary key, nama_mhs varchar(50), jenis_kelamin enum('L','P') default 'L', alamat varchar(50), kota varchar(20) default 'Malang', asal_sma varchar(30), no_hp varchar(12), umur integer, kode_prodi varchar(6), foreign key(kode_prodi) references prodi(kode_prodi));
Query OK, 0 rows affected (0.046 sec)
```

c. Tabel *mata_kuliah*

Field	Type Data
mk_id	VARCHAR (10) PRIMARY KEY
nama_mk	VARCHAR (50)
jumlah_jam	FLOAT (4,2)
sks	INTEGER

5

```
mysql> 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)
```

d. Tabel *ruang*

Field	Type Data
ruang_id	VARCHAR (3) PRIMARY KEY
nama_ruang	VARCHAR (20)
Kapasitas	INTEGER

6

	<pre>mysql> 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 <i>dosen</i></p> <table border="1"> <thead> <tr> <th>Field</th><th>Type Data</th></tr> </thead> <tbody> <tr> <td>nidn</td><td>INTEGER (20) PRIMARY KEY</td></tr> <tr> <td>nama_dosen</td><td>VARCHAR (50)</td></tr> <tr> <td>status</td><td>ENUM ('PNS','KONTRAK') DEFAULT 'PNS'</td></tr> <tr> <td>jenis_kelamin</td><td>ENUM ('L','P') DEFAULT 'L'</td></tr> <tr> <td>no_hp</td><td>VARCHAR (15)</td></tr> </tbody> </table> <pre>mysql> 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>	Field	Type Data	nidn	INTEGER (20) PRIMARY KEY	nama_dosen	VARCHAR (50)	status	ENUM ('PNS','KONTRAK') DEFAULT 'PNS'	jenis_kelamin	ENUM ('L','P') DEFAULT 'L'	no_hp	VARCHAR (15)
Field	Type Data												
nidn	INTEGER (20) PRIMARY KEY												
nama_dosen	VARCHAR (50)												
status	ENUM ('PNS','KONTRAK') DEFAULT 'PNS'												
jenis_kelamin	ENUM ('L','P') DEFAULT 'L'												
no_hp	VARCHAR (15)												
8	<p><Soal> Tambahkan sebuah kolom <i>agama (varchar(10))</i> pada tabel mahasiswa sebagai kolom terakhir Catat : Buat Screenshot dari perintah yang anda ketikkan</p>												
9	<p><Soal> Tambahkan kolom <i>alamat(varchar(50))</i> pada tabel dosen sebagai kolom terakhir Catat : Buat Screenshot dari perintah yang anda ketikkan</p>												
10	<p><Soal> Lakukan insert data ke dalam tabel-tabel yang ada pada database <i>db_polinema</i> sesuai dengan <i>field</i>, tipe data dan panjang datanya Catat : Buat Screenshot dari perintah yang anda ketikkan</p>												
11	<p><Soal> Tampilkan semua tabel yang ada didalam database <i>db_polinema</i> Catat : Buat Screenshot dari perintah yang anda ketikkan</p>												
12	<p><Soal> Tampilkan semua isi tabel yang ada didalam tabel mahasiswa Catat : Buat Screenshot dari perintah yang anda ketikkan</p>												
13	<p><Soal> Tampilkan struktur(metadata) tabel mahasiswa Catat : Buat Screenshot dari perintah yang anda ketikkan</p>												
14	<p><Soal> hilangkan kolom asal_sma yang terdapat didalam tabel mahasiswa Catat : Buat Screenshot dari perintah yang anda ketikkan</p>												

TUGAS

1. Buatlah basis data Akademik dengan data sebagai berikut :

No_Mhs	Nama_mhs	Jurusan	Kd_MK	Nama_mk	Kd_Dosen	Nm_Dosen	nilai
1921001	Aminah	MI	MI350	Basis Data	B104	Ati	85
1921001	Budiman	MI	MI465	Pemrograman	B105	Dita	87
1921002	Carina	MI	MI465	Pemrograman	B105	Dita	85
1921003	Della	TI	TI201	Mobile	C102	Leo	78
1921004	Firda	TI	TI201	Mobile	C102	Leo	80

- deskripsikan struktur data dari table-table berikut serta isikan datanya:
Tabel Mahasiswa {No_Mhs, Nama_mhs}
Tabel Mata_Kuliah {Kd_MK, Nama_MK}
Tabel nilai {No_Mhs, Kode_MK}
tambahkan kolom Jurusan pada tabel Mahasiswa di kolom terakhir
- tambahkan kolom Kode Dosen pada tabel Mata_Kuliah
- tambahkan kolom nilai pada tabel nilai serta berikanlah kunci *foreign key*
- tambahkan Tabel Dosen dengan atributnya Kd_Dosen dan Nama Dosen
- tampilkan semua data yang ada pada tiap tabel

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

Noproyek	NamaProyek	Nopegawai	NamaPegawai	Golongan	BesarGaji
NP001	BRR	Peg01	Anton	A	1.000.000
NP001	BRR	Peg02	Paula	B	900.000
NP001	BRR	Peg06	Koko	C	750.000
NP002	PEMDA	Peg01	Anton	A	1.000.000
NP002	PEMDA	Peg12	Sita	B	900.000
NP002	PEMDA	Peg14	Yusni	B	900.000
NP003	CBR	Peg02	Paula	B	900.000
NP003	CBR	Peg03	Daniar	C	750.000
NP003	CBR	Peg04	Lubis	C	750.000
NP004	ASK	Peg07	Keni	B	900.000
NP004	ASK	Peg08	Sofi	B	900.000
NP004	ASK	Peg06	Yuni	C	750.000
NP005	OB	Peg15	Udin	D	500.000
NP005	OB	Peg16	Didit	D	500.000
NP005	OB	Peg17	Dani	D	500.000

- Deskripsikan struktur data dari table-table berikut serta isikan datanya:
Table Pegawai {Nopegawai, NamaPegawai}
Tabel Golongan {Golongan}
Tabel Proyek {Noproyek}
Tabel Proyekpegawai {Noproyek}
- Tambahkan kolom Golongan pada tabel Pegawai di kolom terakhir
- Tambahkan kolom BesarGaji pada tabel Golongan di kolom terakhir
- Tambahkan kolom NamaProyek pada table Proyek
- Tambahkan kolom NoPegawai pada table Proyekpegawai serta berikanlah kunci *foreign key*
- Tampilkan semua data yang ada pada tiap tabel

~~ Selamat Mengerjakan ~~