

# **JOBSHEET**

## **PRAKTIKUM BASIS DATA**

Jurusan Teknologi Informasi

**POLITEKNIK NEGERI MALANG**



## **PERTEMUAN 10**

**Dasar MySQL**

### **Team Teaching:**

Dwi Puspitasari, S.Kom., M.Kom.

Yan Watequlis Syaifudin, ST., MMT., PhD.

Annisa Puspa Kirana, S. Kom, M.Kom

Yoppy Yunhasnawa, S.ST., M.Sc.

Candra Bella Vista, S.Kom., MT

Elok Nur Hamdana, S.T., M.T

Annisa Taufika Firdausi, ST., MT

Muhammad Shulhan Khairy, S.Kom, M.Kom

Farid Angga Pribadi, S.Kom.,M.Kom

Retno Damayanti, S.Pd., M.T.



Jurusan Teknologi Informasi - Politeknik Negeri Malang

Jobsheet-11: Dasar MySQL

Mata Kuliah : Basis Data

Pengampu : Tim Ajar Basis Data

Nama : Fitri Cahyaniati(11)

April 2024

---

## 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
  - Pengubahan 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/memanipulasi 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><code>C:\&gt;Program Files\xampp\mysql\bin&gt;mysql -u root -p</code> (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&gt; </pre>						
2	<p>Buatlah sebuah database dengan nama <b>db_polinema</b></p> <pre> mysql&gt; create database db_polinema; Query OK, 1 row affected (0.01 sec)  mysql&gt; show databases; +-----+   Database   +-----+   information_schema     db_polinema     mysql     performance_schema     sys   +-----+ 5 rows in set (0.00 sec) </pre>						
	<p>Sebelum melanjutkan langkah No. 3, jalankan perintah "use db_polinema"</p> <pre> mysql&gt; use db_polinema; Database changed mysql&gt; </pre>						
3	<table border="1" data-bbox="228 1608 1484 1733"> <thead> <tr> <th data-bbox="228 1608 584 1653">a. Tabel <b>pField</b></th><th data-bbox="584 1608 1484 1653">Type Data</th></tr> </thead> <tbody> <tr> <td data-bbox="228 1653 584 1688">kode_prodi</td><td data-bbox="584 1653 1484 1688">VARCHAR (6) PRIMARY KEY</td></tr> <tr> <td data-bbox="228 1688 584 1733">nama_prodi</td><td data-bbox="584 1688 1484 1733">VARCHAR (30)</td></tr> </tbody> </table> <pre> mysql&gt; create table prodi (kode_prodi char(6) primary key,nama_prodi char(30)); Query OK, 0 rows affected (0.02 sec) </pre>	a. Tabel <b>pField</b>	Type Data	kode_prodi	VARCHAR (6) PRIMARY KEY	nama_prodi	VARCHAR (30)
a. Tabel <b>pField</b>	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)
```

6	d. Tabel <i>ruang</i>		
	<i>Field</i>	<i>Type Data</i>	
	ruang_id	VARCHAR (3) PRIMARY KEY	
	nama_ruang	VARCHAR (20)	
	Kapasitas	INTEGER	
<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	e. Tabel <i>dosen</i>		
	<i>Field</i>	<i>Type Data</i>	
	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)	
	<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	<Soal> Tambahkan sebuah kolom <i>agama (varchar(10))</i> pada tabel mahasiswa sebagai kolom terakhir Catat : Buat Screenshot dari perintah yang anda ketikkan	
	9	<Soal> Tambahkan kolom <i>alamat(varchar(50))</i> pada tabel dosen sebagai kolom terakhir Catat : Buat Screenshot dari perintah yang anda ketikkan	
10	<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		
11	<Soal> Tampilkan semua tabel yang ada didalam database <i>db_polinema</i> Catat : Buat Screenshot dari perintah yang anda ketikkan		
12	<Soal> Tampilkan semua isi tabel yang ada didalam tabel mahasiswa Catat : Buat Screenshot dari perintah yang anda ketikkan		
13	<Soal> Tampilkan struktur(metadata) tabel mahasiswa		

	Catat : Buat Screenshot dari perintah yang anda ketikkan
14	<p>&lt;Soal&gt;</p> <p>hilangkan kolom asal_sma yang terdapat didalam tabel mahasiswa</p> <p>Catat : Buat Screenshot dari perintah yang anda ketikkan</p>

1.

```

Command Prompt - mysql -u
Microsoft Windows [Version 10.0.22621.3527]
(c) Microsoft Corporation. All rights reserved.

C:\Users\HP>cd C:\xampp\mysql\bin

C:\xampp\mysql\bin>mysql -u root -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 8
Server version: 10.4.32-MariaDB mariadb.org binary distribution

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> create database db_polinema;
Query OK, 1 row affected (0.001 sec)

MariaDB [(none)]> |

```

2.

```

MariaDB [(none)]> show databases;
+-----+
| Database |
+-----+
| db_polinema |
| information_schema |
| jual_boneka |
| mysql |
| penjualan_produk |
| performance_schema |
| phpmyadmin |
| test |
+-----+
8 rows in set (0.042 sec)

```

3.

```

MariaDB [(none)]> use db_polinema;
Database changed
MariaDB [db_polinema]> create table prodi (kode_prodi char(6) primary key, nama_prodi char(30));
Query OK, 0 rows affected (0.012 sec)

```

4.

```
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 (0.041 sec)
```

5.

```
MariaDB [db_polinema]> create table mata_kuiah (mk_id char(10) primary key,
nama_mk char(50), jumlah_jam float(4,2), sks integer);
Query OK, 0 rows affected (0.012 sec)
```

6.

```
MariaDB [db_polinema]> create table ruang (ruang_id char(3) primary key,
nama_ruang char(20), kapasitas integer);
Query OK, 0 rows affected (0.014 sec)
```

7.

```
MariaDB [db_polinema]> ALTER TABLE mahasiswa
-> ADD COLUMN agama varchar(10);
Query OK, 0 rows affected (0.010 sec)
Records: 0 Duplicates: 0 Warnings: 0
```

8. Tambahkan sebuah kolom *agama* (*varchar(10)*) pada tabel mahasiswa sebagai kolom terakhir  
 Catat : Buat Screenshot dari perintah yang anda ketikkan

```
MariaDB [db_polinema]> ALTER TABLE dosen
-> ADD COLUMN alamat varchar(50);
Query OK, 0 rows affected (0.009 sec)
Records: 0 Duplicates: 0 Warnings: 0
```

9. Tambahkan kolom *alamat* (*varchar(50)*) pada tabel dosen sebagai kolom terakhir  
 Catat : Buat Screenshot dari perintah yang anda ketikkan

```
MariaDB [db_polinema]> 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 (0.014 sec)
```

10.

Lakukan insert data ke dalam tabel-tabel yang ada pada database *db\_polinema* sesuai dengan *field*, tipe data dan panjang datanya

```
MariaDB [db_polinema]> INSERT INTO prodi(kode_prodi, nama_prodi)
-> VALUES
-> ('SIB11', 'SIB'),
-> ('SIB01', 'TI');
Query OK, 2 rows affected (0.003 sec)
Records: 2 Duplicates: 0 Warnings: 0
```

```
MariaDB [db_polinema]> INSERT INTO mahasiswa (nim, nama_mhs, jenis_kelamin, alamat, kota, asal_sma, no_hp, umur, agama)
-> VALUES
-> (2341760198, 'Fitri', 'P', 'Jl.pisang kipas', 'Namlea', 'SMAN3 BURU', '081240244083', 19, 'islam'),
-> (1243455199, 'Ahmad Hayyin B', 'L', 'Jl.Lowokdoro 3', 'Malang', 'SMK Telkom Malang', '081359131315', 20, 'Islam');
Query OK, 2 rows affected, 1 warning (0.003 sec)
Records: 2 Duplicates: 0 Warnings: 1
```

11. Tampilkan semua isi tabel yang ada didalam tabel mahasiswa **Catat : Buat Screenshot dari perintah yang anda ketikkan**

```
MariaDB [db_polinema]> show tables
-> ;
+-----+
| Tables_in_db_polinema |
+-----+
| dosen                  |
| mahasiswa              |
| mata_kuiah             |
| prodi                  |
| ruang                  |
+-----+
5 rows in set (0.001 sec)
```



12. Tampilkan semua isi tabel yang ada didalam tabel mahasiswa **Catat : Buat Screenshot dari perintah yang anda ketikkan**

```
MariaDB [db_polinema]> SELECT * FROM mahasiswa;
```

nim	nama_mhs	jenis_kelamin	alamat	kota	asal_sma
no_hp	umur	kode_prodi	agama		
1243455199	Ahmad Hayyin B	L	JL.Lowokdoro 3	Malang	SMK Telkom Malang
081359131315	20	NULL	Islam		
2147483647	Fitri	P	Jl.pisang kipas	Namlea	SMAN3 BURU
081240244083	19	NULL	islam		

13. Tampilkan struktur(metadata) tabel mahasiswa

```
MariaDB [db_polinema]> DESCRIBE mahasiswa;
```

Field	Type	Null	Key	Default	Extra
nim	int(8)	NO	PRI	NULL	
nama_mhs	varchar(50)	YES		NULL	
jenis_kelamin	enum('L','P')	YES		L	
alamat	varchar(50)	YES		NULL	
kota	varchar(20)	YES		Malang	
asal_sma	varchar(30)	YES		NULL	
no_hp	varchar(12)	YES		NULL	
umur	int(11)	YES		NULL	
kode_prodi	varchar(6)	YES	MUL	NULL	
agama	varchar(10)	YES		NULL	

10 rows in set (0.019 sec)

```
MariaDB [db_polinema]> SHOW COLUMNS FROM mahasiswa;
```

Field	Type	Null	Key	Default	Extra
nim	int(8)	NO	PRI	NULL	
nama_mhs	varchar(50)	YES		NULL	
jenis_kelamin	enum('L','P')	YES		L	
alamat	varchar(50)	YES		NULL	
kota	varchar(20)	YES		Malang	
asal_sma	varchar(30)	YES		NULL	
no_hp	varchar(12)	YES		NULL	
umur	int(11)	YES		NULL	
kode_prodi	varchar(6)	YES	MUL	NULL	
agama	varchar(10)	YES		NULL	

10 rows in set (0.020 sec)

```
MariaDB [db_polinema]>
```

14. hilangkan kolom asal\_sma yang terdapat didalam tabel mahasiswa **Catat : Buat Screenshot dari perintah yang anda ketikkan**

```
MariaDB [db_polinema]> ALTER TABLE mahasiswa
-> DROP COLUMN asal_sma;
Query OK, 0 rows affected (0.019 sec)
Records: 0 Duplicates: 0 Warnings: 0

MariaDB [db_polinema]> |
```

```
MariaDB [db_polinema]> DESCRIBE mahasiswa;
+-----+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| nim   | int(8) | NO | PRI | NULL | |
| nama_mhs | varchar(50) | YES | | NULL | |
| jenis_kelamin | enum('L','P') | YES | | L | |
| alamat | varchar(50) | YES | | NULL | |
| kota  | varchar(20) | YES | | Malang | |
| no_hp | varchar(12) | YES | | NULL | |
| umur  | int(11) | YES | | NULL | |
| kode_prodi | varchar(6) | YES | MUL | NULL | |
| agama | varchar(10) | YES | | NULL | |
+-----+-----+-----+-----+-----+-----+
9 rows in set (0.021 sec)
```

## 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

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

- Tabel Mahasiswa {No\_Mhs, Nama\_mhs}

```
MariaDB [(none)]> create database db_akademik;
Query OK, 1 row affected (0.001 sec)

MariaDB [(none)]> use db_akademik;
Database changed
MariaDB [db_akademik]> CREATE TABLE mahasiswa (
-> No_Mhs INT PRIMARY KEY,
-> Nama_mhs VARCHAR(255),
-> Jurusan VARCHAR(255)
-> );
Query OK, 0 rows affected (0.014 sec)
```

```
MariaDB [db_akademik]> INSERT INTO Mahasiswa (No_Mhs, Nama_mhs, Jurusan)
-> VALUES
-> (1921001, 'KIKI', 'SIB'),
-> (2341760198, 'CHYA', 'SIB');
Query OK, 2 rows affected, 1 warning (0.002 sec)
Records: 2 Duplicates: 0 Warnings: 1
```

- Tabel Mata\_Kuliah {Kd\_MK, Nama\_MK}

```
MariaDB [db_akademik]> CREATE TABLE mata_kuliah
-> ( kd_mk varchar(225) Primary Key,
-> nama_mk varchar(225),
-> kode_dosen varchar(225));
Query OK, 0 rows affected (0.012 sec)

MariaDB [db_akademik]> INSERT INTO mata_kuliah (Kd_MK, Nama_MK, Kode_Dos
en)
-> VALUES
-> ('TI350', 'Basis Data', 'B104'),
-> ('SIB201', 'Algoritma', 'C102');
Query OK, 2 rows affected (0.004 sec)
Records: 2 Duplicates: 0 Warnings: 0

MariaDB [db_akademik]> |
```

- Tabel nilai {No\_Mhs, Kode\_MK}

```
MariaDB [db_akademik]> CREATE TABLE nilai (
-> No_Mhs INT,
-> Kode_MK VARCHAR(255),
-> nilai INT,
-> FOREIGN KEY (No_Mhs) REFERENCES Mahasiswa(No_Mhs),
-> FOREIGN KEY (Kode_MK) REFERENCES Mata_Kuliah(Kd_MK)
-> );
Query OK, 0 rows affected (0.045 sec)
```

```
MariaDB [db_akademik]> INSERT INTO nilai (No_Mhs, Kode_MK, nilai)
-> VALUES
-> (1921001, 'TI350', 85),
-> (2341760198, 'SIB201', 90);
Query OK, 2 rows affected, 1 warning (0.003 sec)
Records: 2 Duplicates: 0 Warnings: 1
```

- tambahkan kolom Jurusan pada tabel Mahasiswa di kolom terakhir

```
MariaDB [db_akademik]> ALTER TABLE Mahasiswa
-> ADD COLUMN Prodi VARCHAR(255);
Query OK, 0 rows affected (0.007 sec)
Records: 0 Duplicates: 0 Warnings: 0
```

- tambahkan kolom Kode Dosen pada tabel Mata\_Kuliah

```
MariaDB [db_akademik]> ALTER TABLE Mata_Kuliah
-> ADD COLUMN Nama_Dosen VARCHAR(255);
Query OK, 0 rows affected (0.006 sec)
Records: 0 Duplicates: 0 Warnings: 0
```

- tambahkan kolom nilai pada tabel nilai serta berikanlah kunci *foreign key*

```
MariaDB [db_akademik]> ALTER TABLE nilai
-> ADD COLUMN Total_nilai INT;
Query OK, 0 rows affected (0.007 sec)
Records: 0 Duplicates: 0 Warnings: 0
```

d. tambahkan Tabel Dosen dengan atributnya Kd\_Dosen dan Nama Dosen

```
MariaDB [db_akademik]> CREATE TABLE Dosen (
-> Kd_Dosen VARCHAR(255) PRIMARY KEY,
-> Nama_Dosen VARCHAR(255)
-> );
Query OK, 0 rows affected (0.010 sec)
```

```
MariaDB [db_akademik]> INSERT INTO Dosen (Kd_Dosen, Nama_Dosen)
-> VALUES
-> ('B105', 'Cya'),
-> ('B104', 'dian');
Query OK, 2 rows affected (0.004 sec)
Records: 2 Duplicates: 0 Warnings: 0
```

e. tampilkan semua data yang ada pada tiap tabel

```
MariaDB [db_akademik]> SELECT * FROM Mahasiswa;
+-----+-----+-----+-----+
| No_Mhs | Nama_mhs | Jurusan | Prodi |
+-----+-----+-----+-----+
| 1921001 | KIKI     | SIB     | NULL  |
| 2147483647 | CHYA    | SIB     | NULL  |
+-----+-----+-----+-----+
2 rows in set (0.001 sec)

MariaDB [db_akademik]> SELECT * FROM Mata_Kuliah;
+-----+-----+-----+-----+
| kd_mk | nama_mk | kode_dosen | Nama_Dosen |
+-----+-----+-----+-----+
| SIB201 | Algoritma | C102      | NULL       |
| TI350  | Basis Data | B104      | NULL       |
+-----+-----+-----+-----+
2 rows in set (0.000 sec)

MariaDB [db_akademik]> SELECT * FROM nilai;
+-----+-----+-----+-----+
| No_Mhs | Kode_MK | nilai | Total_nilai |
+-----+-----+-----+-----+
| 1921001 | TI350   | 85    | NULL        |
| 2147483647 | SIB201 | 90    | NULL        |
+-----+-----+-----+-----+
2 rows in set (0.001 sec)

MariaDB [db_akademik]> SELECT * FROM Dosen;
+-----+-----+
| Kd_Dosen | Nama_Dosen |
+-----+-----+
| B104     | dian       |
| B105     | Cya        |
+-----+-----+
2 rows in set (0.000 sec)

MariaDB [db_akademik]>
```

**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

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

- Table Pegawai {Nopegawai, NamaPegawai}

```
MariaDB [(none)]> create database pegawai;
Query OK, 1 row affected (0.001 sec)
```

```
MariaDB [(none)]> USE pegawai;
Database changed
```

```
MariaDB [pegawai]> CREATE TABLE pegawai (
  -> Nopegawai VARCHAR(255) PRIMARY KEY,
  -> NamaPegawai VARCHAR(255),
  -> Golongan CHAR(1)
  -> );
Query OK, 0 rows affected (0.011 sec)
```



```

MariaDB [pegawai]> INSERT INTO Pegawai (Nopegawai, NamaPegawai, Golongan
)
-> VALUES
-> ('Peg01', 'Anton', 'A'),
-> ('Peg02', 'Paula', 'B'),
-> ('Peg06', 'Koko', 'C'),
-> ('Peg12', 'Sita', 'B'),
-> ('Peg14', 'Yusni', 'B'),
-> ('Peg03', 'Danial', 'C'),
-> ('Peg04', 'Lubis', 'C'),
-> ('Peg07', 'Keni', 'B'),
-> ('Peg08', 'Sofi', 'B'),
-> ('Peg15', 'Udin', 'D'),
-> ('Peg16', 'Didit', 'D'),
-> ('Peg17', 'Dani', 'D');
Query OK, 12 rows affected (0.004 sec)
Records: 12 Duplicates: 0 Warnings: 0

```

- Tabel Golongan  
{Golongan}

```

MariaDB [pegawai]> CREATE TABLE Golongan (
-> Golongan CHAR(1) PRIMARY KEY,
-> BesarGaji DECIMAL(10,2)
-> );
Query OK, 0 rows affected (0.010 sec)

MariaDB [pegawai]> INSERT INTO Golongan (Golongan, BesarGaji)
-> VALUES
-> ('A', 1000000),
-> ('B', 900000),
-> ('C', 750000),
-> ('D', 500000);
Query OK, 4 rows affected (0.004 sec)
Records: 4 Duplicates: 0 Warnings: 0

MariaDB [pegawai]>

```

- Tabel Proyek  
{Noprojek}

```
MariaDB [pegawai]> CREATE TABLE proyek (
  -> Noprojek VARCHAR(255) PRIMARY KEY,
  -> NamaProyek VARCHAR(255)
  -> );
Query OK, 0 rows affected (0.010 sec)

MariaDB [pegawai]> INSERT INTO Proyek (Noprojek, NamaProyek)
  -> VALUES
  -> ('NP001', 'BRR'),
  -> ('NP002', 'PEMDA'),
  -> ('NP003', 'CBR'),
  -> ('NP004', 'ASK'),
  -> ('NP005', 'OB');
Query OK, 5 rows affected (0.003 sec)
Records: 5 Duplicates: 0 Warnings: 0
```

- Tabel Proyekpegawai {Noprojek}

```
MariaDB [pegawai]> CREATE TABLE Proyekpegawai (
  -> Noprojek VARCHAR(255),
  -> Nopegawai VARCHAR(255),
  -> FOREIGN KEY (Noprojek) REFERENCES Proyek(Noprojek),
  -> FOREIGN KEY (Nopegawai) REFERENCES Pegawai(Nopegawai)
  -> );
Query OK, 0 rows affected (0.024 sec)

MariaDB [pegawai]> INSERT INTO Proyekpegawai (Noprojek, Nopegawai)
  -> VALUES
  -> ('NP001', 'Peg01'),
  -> ('NP001', 'Peg02'),
  -> ('NP001', 'Peg06'),
  -> ('NP002', 'Peg01'),
  -> ('NP002', 'Peg12'),
  -> ('NP002', 'Peg14'),
  -> ('NP003', 'Peg02'),
  -> ('NP003', 'Peg03'),
  -> ('NP003', 'Peg04'),
  -> ('NP004', 'Peg07'),
  -> ('NP004', 'Peg08'),
  -> ('NP004', 'Peg06'),
  -> ('NP005', 'Peg15'),
  -> ('NP005', 'Peg16'),
  -> ('NP005', 'Peg17');
Query OK, 15 rows affected (0.010 sec)
Records: 15 Duplicates: 0 Warnings: 0
```

- b. Tambahkan kolom Golongan pada tabel Pegawai di kolom terakhir

```
MariaDB [pegawai]> ALTER TABLE Pegawai
  -> ADD COLUMN Golongan CHAR(1) AFTER NamaPegawai;
ERROR 1060 (42S21): Duplicate column name 'Golongan'
MariaDB [pegawai]> |
```

- c. Tambahkan kolom Besar Gaji pada tabel Golongan di kolom terakhir

```
MariaDB [pegawai]> ALTER TABLE Golongan
-> ADD COLUMN BesarGaji DECIMAL(10,2) AFTER Golongan;
ERROR 1060 (42S21): Duplicate column name 'BesarGaji'
```

- d. Tambahkan kolom NamaProyek pada table Proyek

```
MariaDB [pegawai]> ALTER TABLE Proyek
-> ADD COLUMN NamaProyek VARCHAR(255) AFTER Noprojek;
ERROR 1060 (42S21): Duplicate column name 'NamaProyek'
```

- e. Tambahkan kolom NoPegawai pada table Proyekpegawai serta berikanlah kunci *foreign key*

```
MariaDB [pegawai]> ALTER TABLE Proyekpegawai
-> ADD COLUMN Nopegawai VARCHAR(255) AFTER Noprojek,
-> ADD FOREIGN KEY (Nopegawai) REFERENCES Pegawai(Nopegawai);
ERROR 1060 (42S21): Duplicate column name 'Nopegawai'
MariaDB [pegawai]>
```

- f. Tampilkan semua data yang ada pada tiap tabel

```
MariaDB [pegawai]> SELECT * FROM Pegawai;
```

Nopegawai	NamaPegawai	Golongan
Peg01	Anton	A
Peg02	Paula	B
Peg03	Daniar	C
Peg04	Lubis	C
Peg06	Koko	C
Peg07	Keni	B
Peg08	Sofi	B
Peg12	Sita	B
Peg14	Yusni	B
Peg15	Udin	D
Peg16	Didit	D
Peg17	Dani	D

12 rows in set (0.001 sec)

```
MariaDB [pegawai]> SELECT * FROM Golongan;
```

Golongan	BesarGaji
A	1000000.00
B	900000.00
C	750000.00
D	500000.00

4 rows in set (0.000 sec)

```
MariaDB [pegawai]> SELECT * FROM Proyek;
```

Noprojek	NamaProyek
NP001	BRR
NP002	PEMDA
NP003	CBR
NP004	ASK
NP005	OB

```
5 rows in set (0.000 sec)
```

```
MariaDB [pegawai]> SELECT * FROM Proyekpegawai;
```

Noprojek	Nopegawai
NP001	Peg01
NP001	Peg02
NP001	Peg06
NP002	Peg01
NP002	Peg12
NP002	Peg14
NP003	Peg02
NP003	Peg03
NP003	Peg04
NP004	Peg07
NP004	Peg08
NP004	Peg06
NP005	Peg15
NP005	Peg16
NP005	Peg17

```
15 rows in set (0.000 sec)
```

*~~ Selamat Mengerjakan ~~*

