

# **MODUL PRAKTIKUM BASIS DATA**



**Disusun Oleh : Rahmi Eka Putri, MT**

**Program Studi Informatika  
Fakultas Teknologi Informasi  
Universitas Andalas  
2024**

## Modul Praktikum 4 - MySql

### Tujuan :

1. Mengetahui database MySql melalui MySql Console
2. Memahami query dalam MySql (CREATE, DROP, INSERT)
3. Memahami query untuk menampilkan data dari beberapa tabel

### 1. PENDAHULUAN

MySQL adalah Sebuah program database server yang mampu menerima dan mengirimkan datanya sangat cepat, multi user serta menggunakan perintah dasar SQL ( Structured Query Language ). MySQL merupakan dua bentuk lisensi, yaitu FreeSoftware dan Shareware. MySQL yang biasa digunakan adalah MySQL FreeSoftware yang berada dibawah Lisensi GNU/GPL ( General Public License ).

MySQL Merupakan sebuah database server yang free, artinya kita bebas menggunakan database ini untuk keperluan pribadi atau usaha tanpa harus membeli atau membayar lisensinya. MySQL pertama kali dirintis oleh seorang programmer database bernama Michael Widenius . Selain database server, MySQL juga merupakan program yang dapat mengakses suatu database MySQL yang berposisi sebagai Server, yang berarti program kita berposisi sebagai Client. Jadi MySQL adalah sebuah database yang dapat digunakan sebagai Client maupun server.

Database MySQL merupakan suatu perangkat lunak database yang berbentuk database relasional atau disebut Relational Database Management System ( RDBMS ) yang menggunakan suatu bahasa permintaan yang bernama SQL (Structured Query Language ).

### Tipe Data pada MySQL

Tipe data adalah suatu bentuk pemodelan data yang dideklarasikan pada saat melakukan pembuatan tabel. Tipe data ini akan mempengaruhi setiap data yang akan dimasukkan ke dalam sebuah tabel. Data yang akan dimasukkan harus sesuai dengan tipe data yang dideklarasikan.

**Tabel Tipe Data untuk Bilangan (Number)**

<b>Tipe Data</b>	<b>Keterangan</b>
TINYINT	Ukuran 1 byte. Bilangan bulat terkecil, dengan jangkauan untuk bilangan bertanda: -128 sampai dengan 127 dan untuk yang tidak bertanda : 0 s/d 255. Bilangan tak bertanda dengan kata UNSIGNED
SMALLINT	Ukuran 2 Byte. Bilangan bulat dengan jangkauan untuk bilangan bertanda : -32768 s/d 32767 dan untuk yang tidak bertanda : 0 s/d 65535
MEDIUMINT	Ukuran 3 byte. Bilangan bulat dengan jangkauan untuk bilangan bertanda : -8388608 s/d 8388607 dan untuk yang tidak bertanda : 0 s/d 16777215
INT	Ukuran 4 byte. Bilangan bulat dengan jangkauan untuk bilangan bertanda : -2147483648 s/d 2147483647 dan untuk yang tidak bertanda : 0 s/d 4294967295
INTEGER	Ukuran 4 byte. Sinonim dari int
BIGINT	Ukuran 8 byte. Bilangan bulat terbesar dengan jangkauan untuk bilangan bertanda : -9223372036854775808 s/d 9223372036854775807 dan untuk yang tidak bertanda : 0 s/d 1844674473709551615
FLOAT	Ukuran 4 byte. Bilangan pecahan
DOUBLE	Ukuran 8 byte. Bilangan pecahan
DOUBLEPRECISION	Ukuran 8 byte. Bilangan pecahan
REAL	Ukuran 8 byte. Sinonim dari DOUBLE
DECIMAL (M,D)	Ukuran M byte. Bilangan pecahan, misalnya DECIMAL(5,2) dapat digunakan untuk menyimpan bilangan -99,99 s/d 99,99
NUMERIC (M,D)	Ukuran M byte. Sinonim dari DECIMAL, misalnya NUMERIC(5,2) dapat digunakan untuk menyimpan bilangan -99,99 s/d 99,99

**Tabel Tipe Data untuk Tanggal dan Jam**

<b>Tipe Data</b>	<b>Keterangan</b>
DATETIME	Ukuran 8 byte. Kombinasi tanggal dan jam, dengan jangkauan dari '1000-01-01 00:00:00' s/d '9999-12-31 23:59:59'

Tipe Data	Keterangan
DATE	Ukuran 3 Byte. Tanggal dengan jangkauan dari '1000-01-01' s/d '9999-12-31'
TIMESTAMP	Ukuran 4 byte. Kombinasi tanggal dan jam, dengan jangkauan dari '1970-01-01 00:00:00' s/d '2037'
TIME	Ukuran 3 byte. Waktu dengan jangkauan dari '839:59:59' s/d '838:59:59'
YEAR	Ukuran 1 byte. Data tahun antara 1901 s/d 2155

Tabel Tipe Data untuk Karakter dan Lain-lain

Tipe Data	Keterangan
CHAR	Mampu menangani data hingga 255 karakter. Tipe data CHAR mengharuskan untuk memasukkan data yang telah ditentukan
VARCHAR	Mampu menangani data hingga 255 karakter. Tipe data VARCHAR tidak mengharuskan untuk memasukkan data yang telah ditentukan
TINYBLOB, TINYTEXT	Ukuran 255 byte. Mampu menangani data sampai $2^8-1$ data.
BLOB, TEXT	Ukuran 65535 byte. Tipe string yang mampu menangani data hingga $2^{16}-1$ (16M-1) data
MEDIUMBLOB, MEDIUMTEXT	Ukuran 16777215 byte. Mampu menyimpan data hingga $2^{24}-1$ (16M-1) Data
LOB, LONGTEXT	Ukuran 4294967295 byte. Mampu menyimpan data hingga berukuran GIGA BYTE. Tipe data ini memiliki batas penyimpanan hingga $2^{32}-1$ (4G-1) data
ENUM('nilai1','nilai2',..., 'nilaiN')	Ukuran 1 atau 2 byte. Tergantung jumlah nilai enumerasinya (maksimum 65535 nilai)
SET('nilai1','nilai2',..., 'nilaiN')	1,2,3,4 atau 8 byte, tergantung jumlah anggota himpunan (maksimum 64 anggota)

## 2. MENGENAL SQL ( STRUCTURED QUERY LANGUAGE )

SQL ( Structured Query Language ) adalah sebuah bahasa permintaan database yang terstruktur. Bahasa SQL ini dibuat sebagai bahasa yang dapat merelasikan beberapa tabel dalam database maupun merelasikan antar database.

SQL dibagi menjadi tiga bentuk Query, yaitu :

### a. *DDL ( Data Definition Language )*

DDL adalah sebuah metode Query SQL yang berguna untuk mendefinisikan data pada sebuah Database, Query yang dimiliki DDL adalah :

- CREATE : Digunakan untuk membuat Database dan Tabel
- DROP : Digunakan untuk menghapus Tabel dan Database
- ALTER : Digunakan untuk melakukan perubahan struktur tabel yang telah dibuat baik menambah Field (Add), mengganti nama Field ataupun menamakannya kembali (Rename), dan menghapus Field (Delete).

### b. *DML ( Data Manipulation Language )*

DML adalah sebuah metode Query yang dapat digunakan apabila DDL telah terjadi, sehingga fungsi dari Query DML ini untuk melakukan pemanipulasian database yang telah dibuat. Query yang dimiliki DML adalah :

- INSERT : Digunakan untuk memasukkan data pada Tabel Database
- UPDATE : Digunakan untuk pengubahan terhadap data yang ada pada Tabel Database
- DELETE : Digunakan untuk Penhapusan data pada tabel Database

### c. *DCL ( Data Control Language )*

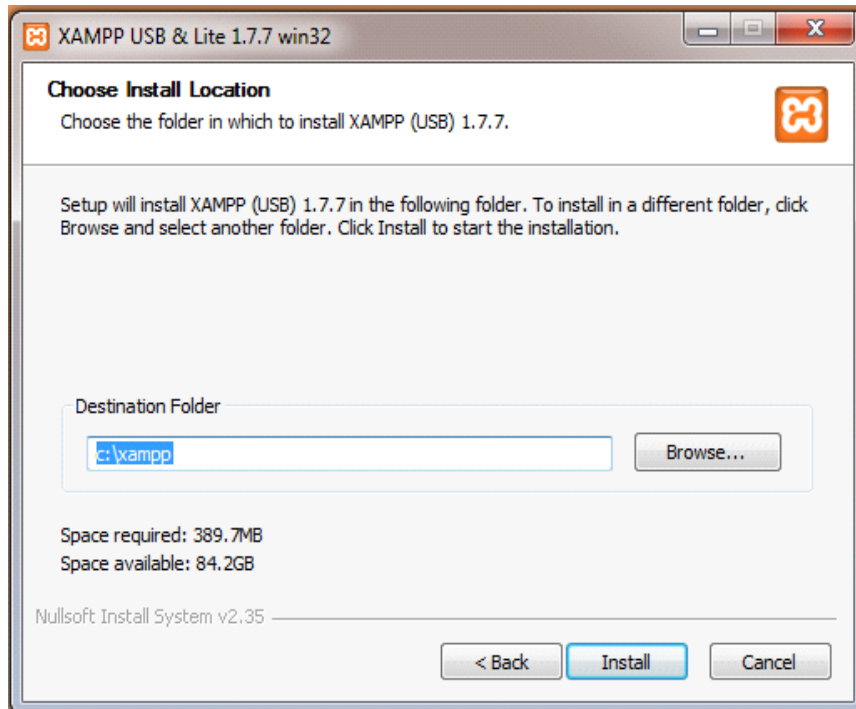
DCL adalah sebuah metode Query SQL yang digunakan untuk memberikan hak otorisasi mengakses Database, mengalokasikan space, pendefinisian space, dan pengauditan penggunaan database. Query yang dimiliki DCL adalah :

- GRANT : Untuk mengizinkan User mengakses Tabel dalam Database.
- REVOKE : Untuk membatalkan izin hak user, yang ditetapkan oleh perintah GRANT
- COMMIT : Menetapkan penyimpanan Database
- ROLLBACK : Membatalkan penyimpanan Database

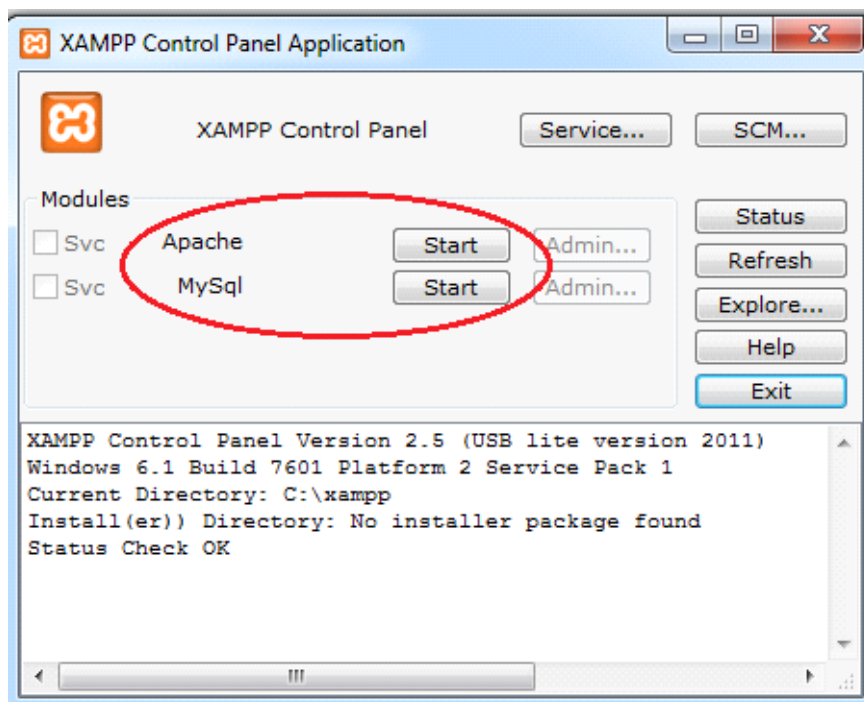
### 3. MEMULAI MYSQL

Langkah – langkah :

1. Install **xampp-win32-1.7.7-usb-lite.exe** ke drive C



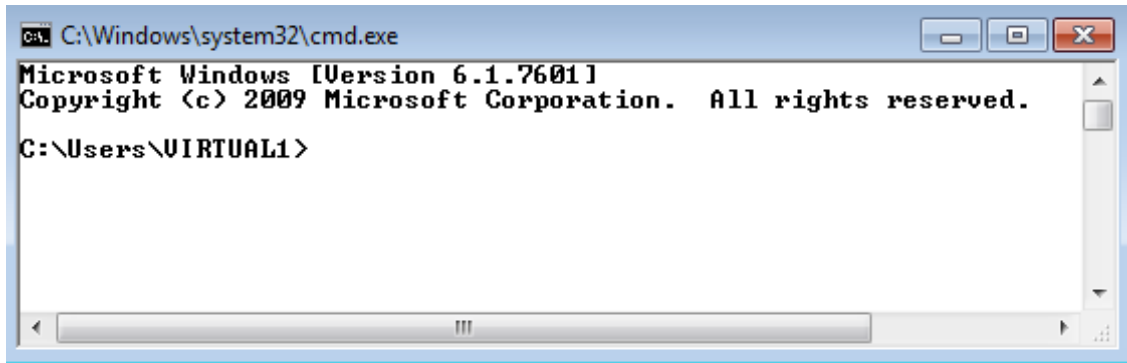
2. Cari folder **xampp** di drive C kemudian double klik **xampp-control.exe**



3. Aktifkan Apache dan MySql dengan cara klik tombol start

4. Masuk ke dalam Console MySql dengan cara :

Klik start , kemudian ketikkan perintah cmd



```
C:\Windows\system32\cmd.exe
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.
C:\Users\VIRTUAL1>
```

5. Masuk ke drive C dengan mengetikkan perintah :

cd\

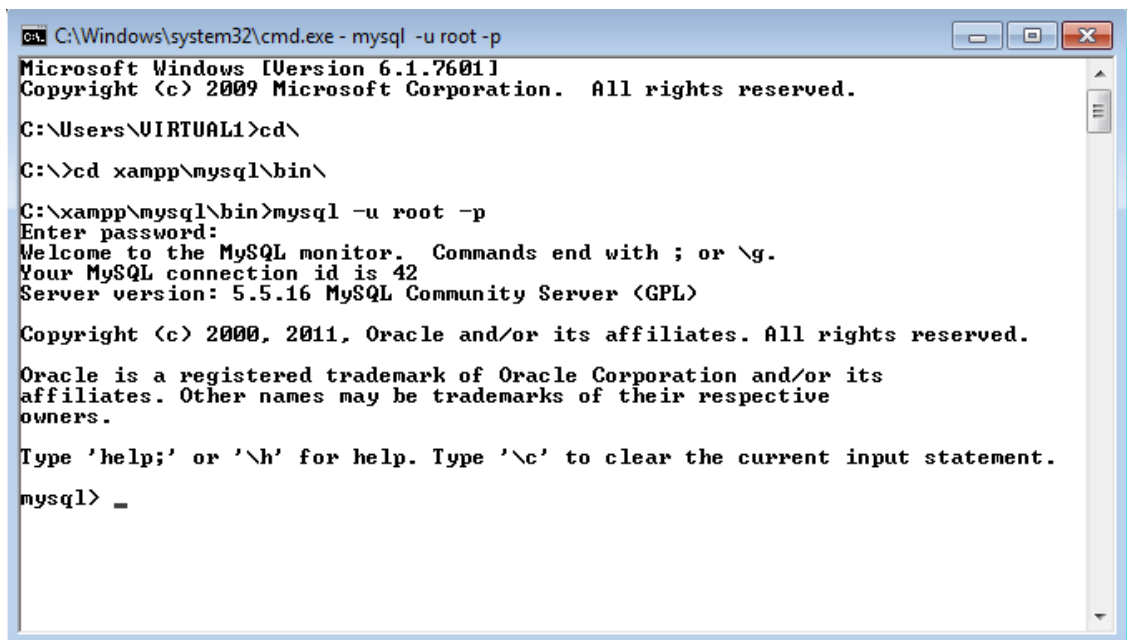
6. Masuk ke folder xampp\mysql\bin dengan mengetikkan perintah :

cd xampp\mysql\bin

7. Masuk ke server MySql dengan menggunakan user root dengan mengetikkan perintah :

mysql -u root -p

password:<enter>



```
C:\Windows\system32\cmd.exe - mysql -u root -p
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.
C:\Users\VIRTUAL1>cd\
C:\>cd xampp\mysql\bin\
C:\xampp\mysql\bin>mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 42
Server version: 5.5.16 MySQL Community Server (GPL)

Copyright (c) 2000, 2011, 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> _
```

#### 4. MEMBUAT DATABASE

Langkah-Langkah untuk membuat database :

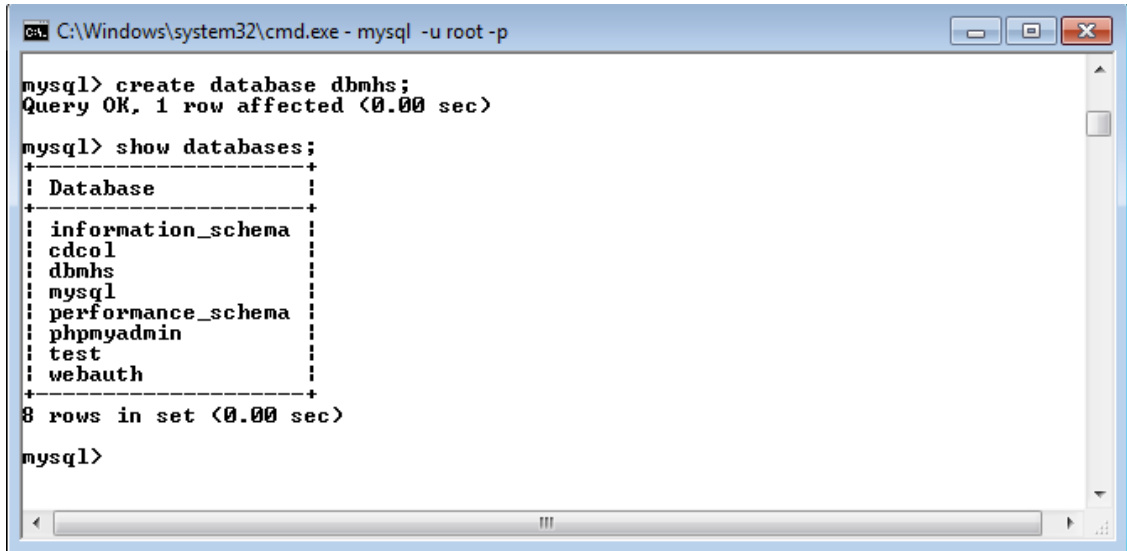
1. Ketikkan query :

**CREATE DATABASE** nama\_database;

Contoh : **CREATE DATABASE** dbmhs;

2. Untuk melihat apakah database dbmhs telah terbentuk, ketikkan query :

**SHOW** databases;



```

C:\Windows\system32\cmd.exe - mysql -u root -p

mysql> create database dbmhs;
Query OK, 1 row affected (0.00 sec)

mysql> show databases;
+-----+
| Database |
+-----+
| information_schema |
| cdcol          |
| dbmhs          |
| mysql          |
| performance_schema |
| phpmyadmin     |
| test           |
| webauth        |
+-----+
8 rows in set (0.00 sec)

mysql>

```

Untuk menghapus Database yang telah dibuat dapat menggunakan query SQL berikut :

**DROP DATABASE** nama\_database;

#### 5. MEMBUAT TABLE

Setelah database terbentuk, langkah berikutnya adalah membuat table-table di dalam database tersebut. Langkah –langkah membuat table adalah :

1. Aktifkan database dimana table akan dimasukkan dengan mengetikkan query :

**USE** nama\_database;

2. Untuk membuat table, ketikkan syntax :

**CREATE TABLE** nama\_tabel  
( field-1 type(length), field-2 type(length), field-3 type(length), ..... ..(.....));

3. Untuk melihat tabel yang ada pada database ketikkan query :

**SHOW TABLES;**

4. Untuk melihat struktur tabel ketikkan query:

**DESC** nama\_tabel;

atau

**DESCRIBE** nama\_tabel;



Untuk menghapus Tabel yang telah dibuat dapat menggunakan query SQL berikut :

**DROP TABLE** nama\_tabel;

**Contoh :**

Buatlah table berikut di dalam database dbmhs

**1. Table Mahasiswa**

No.	Field Name	Data Type
1.	Nobp	Varchar (10)
2.	Nama_mhs	Varchar (25)
3.	Alamat	Varchar (30)
4.	Prodi	Varchar (20)

**2. Table Matakuliah**

No.	Field Name	Data Type
1.	Kode_mtk	Varchar (10)
2.	Nama_mtk	Varchar (25)
3.	Sks	Int
4.	Semester	Int

**3. Table Nilai**

No.	Field Name	Data Type
1.	Nobp	Varchar (10)
2.	Kode_mtk	Varchar (10)
3.	Nilai	Char (2)

Ketikkan query SQL :

**USE** dbmhs;

**CREATE TABLE** Mahasiswa

(Nobp Varchar (10) NOT NULL Primary Key,

Nama\_mhs Varchar (25),

Alamat Varchar (30),

Prodi Varchar (20));

**CREATE TABLE** Matakuliah

(Kode\_mtk Varchar (10) NOT NULL Primary Key,

Nama\_mtk Varchar (25),

Sks Int,

Semester Int);

**CREATE TABLE** Nilai

(Nobp Varchar (10) NOT NULL References Mahasiswa (Nobp),

Kode\_mtk Varchar (10) NOT NULL References Matakuliah (Kode\_mtk),

Nilai Char (2));

```

C:\Windows\system32\cmd.exe - mysql -u root -p

mysql> use dbmhs;
Database changed
mysql> CREATE TABLE Mahasiswa
  -> (Nobp Uvarchar (10) NOT NULL Primary Key,
  -> Nama_mhs Uvarchar (25),
  -> Alamat Uvarchar (30),
  -> Prodi Uvarchar (20));
Query OK, 0 rows affected (0.52 sec)

mysql> CREATE TABLE Matakuliah
  -> (Kode_mtk Uvarchar (10) NOT NULL Primary Key,
  -> Nama_mtk Uvarchar (25),
  -> Sks Int,
  -> Semester Int);
Query OK, 0 rows affected (0.04 sec)

mysql> CREATE TABLE Nilai
  -> (Nobp Uvarchar (10) NOT NULL References Mahasiswa (Nobp),
  -> Kode_mtk Uvarchar (10) NOT NULL References Matakuliah (Kode_mtk),
  -> Nilai Char (2));
Query OK, 0 rows affected (0.03 sec)

mysql>

```

Untuk melihat table dan struktur table, ketikkan query :

**SHOW TABLES;**  
**DESC Mahasiswa;**  
**DESC Matakuliah;**  
**DESC Nilai;**

```

C:\Windows\system32\cmd.exe - mysql -u root -p

mysql> show tables;
+-----+
| Tables_in_dbmhs |
+-----+
| mahasiswa       |
| matakuliah      |
| nilai           |
+-----+
3 rows in set (0.00 sec)

mysql> desc mahasiswa;
+-----+-----+-----+-----+-----+-----+
| Field | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| Nobp  | varchar(10)   | NO   | PRI | NULL    |       |
| Nama_mhs | varchar(25)   | YES  |     | NULL    |       |
| Alamat | varchar(30)   | YES  |     | NULL    |       |
| Prodi  | varchar(20)   | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
4 rows in set (0.00 sec)

mysql> desc matakuliah;
+-----+-----+-----+-----+-----+-----+
| Field | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| Kode_mtk | varchar(10)   | NO   | PRI | NULL    |       |
| Nama_mtk | varchar(25)   | YES  |     | NULL    |       |
| Sks      | int(11)       | YES  |     | NULL    |       |
| Semester | int(11)       | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
4 rows in set (0.00 sec)

mysql> desc nilai;
+-----+-----+-----+-----+-----+-----+
| Field | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| Nobp  | varchar(10)   | NO   |     | NULL    |       |
| Kode_mtk | varchar(10)   | NO   |     | NULL    |       |
| Nilai | char(2)       | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
3 rows in set (0.01 sec)

mysql>

```

## 6. MEMASUKKAN DATA KE DALAM TABLE

Memasukkan data atau entry data, dalam semua program yang menggunakan query SQL sebagai standar permintaannya, digunakan perintah **INSERT**. Syarat untuk memasukkan data adalah telah terciptanya tabel pada sebuah database. Sintax yang digunakan adalah :

**INSERT INTO** nama\_tabel VALUES ('isi\_field1', 'isi\_field2', ..., 'isi\_fieldN');

### Contoh :

1. Masukkan 5 (lima) data kedalam Tabel Mahasiswa

Nobp	Nama_mhs	Alamat	Prodi

**INSERT INTO Mahasiswa VALUES (Nobp, Nama\_mhs, Alamat, Prodi);**

Setelah semua data mahasiswa dimasukkan ke dalam table mahasiswa, lakukan query untuk melihat data mahasiswa dengan menggunakan syntax berikut :

**Select \* From mahasiswa;**

2. Masukkan 8 (delapan) data kedalam Tabel Matakuliah

Kd_mtk	Nama_mtk	Sks	Semester

**INSERT INTO Matakuliah VALUES ('Kd\_mtk, Nama\_mtk, Sks, Semester);**

Setelah semua data matakuliah dimasukkan ke dalam table matakuliah, lakukan query untuk melihat data mahasiswa dengan menggunakan syntax berikut :

**Select \* From matakuliah;**

3. Masukkan 40 data nilai mahasiswa (untuk setiap matakuliah yang diambil oleh mahasiswa) kedalam Tabel Nilai

**INSERT INTO Nilai VALUES (Nobp, Kd\_mtk, Nilai);**

Setelah semua data nilai mahasiswa dimasukkan ke dalam table Nilai, lakukan query untuk melihat data mahasiswa dengan menggunakan syntax berikut :

**Select \* From Nilai;**

## 7. MENAMPILKAN DATA DARI BEBERAPA TABEL

Data yang terdapat dari tabel – tabel yang berelasi dalam sebuah database dapat ditampilkan dengan menggunakan query SELECT yang diikuti oleh oleh klausa WHERE. Sintax yang digunakan adalah :

**SELECT** Tabel\_alias1.nama\_field, Tabel\_alias2.nama\_field,..., Tabel\_aliasN.nama\_field  
**FROM** nama\_tabel1 Tabel\_alias1, nama\_tabel2 Tabel\_alias2,..., nama\_tabelN Tabel\_aliasN  
**WHERE** Tabel\_alias1.field\_key\_tabel1 = Tabel\_alias2.field\_key\_tabel2

### Contoh :

Tampilkan data Nobp, Nama\_mhs, Prodi, Kd\_mtk, Nama\_mtk, Sks, Semester dan Nilai dari Tabel Mahasiswa, Tabel Matakuliah, dan Tabel Nilai !

**SELECT** T1.Nobp, T1.Nama\_mhs, T1.Prodi, T2.Kode\_mtk, T2.Nama\_mtk, T2.Sks,  
 T2.Semester, T3.Nilai  
**FROM** Mahasiswa T1, Matakuliah T2, Nilai T3  
**WHERE** T1.Nobp = T3.Nobp and T2.Kode\_mtk = T3.Kode\_mtk;