

# TEKNOLOGI BASIS DATA

## Praktikum MySQL pada CMD

Diajukan untuk memenuhi salah satu Tugas praktikum mata Kuliah Basis Data



Disusun oleh :

311710187 - Izvio Prijaya Mu'min

Kelas - TI.17.D2

Program Studi Teknik Informatika

Sekolah Tinggi Teknik

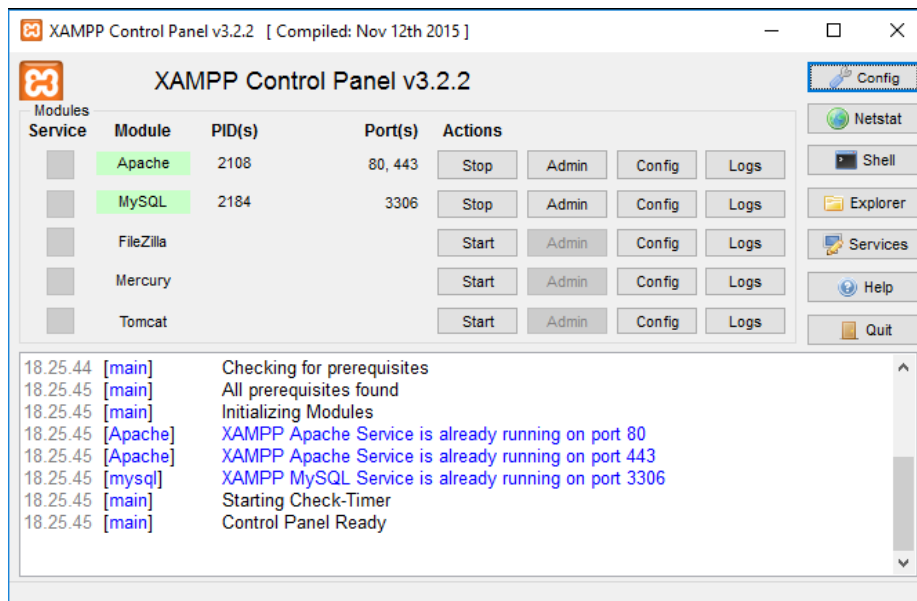
Pelita Bangsa

2019

# TUGAS PRAKTIKUM

## 1. Buat sebuah database dengan nama latihan1

Sebelum membuat sebuah database, terlebih dahulu kita harus mengaktifkan fitur MySQL pada XAMPP (Apabila kita menggunakan Aplikasi XAMPP).



Pada gambar diatas saya telah menjalankan fitur MySQL pada XAMPP setelah kita menjalankan fitur tersebut langkah selanjutnya kita harus login ke sistem MySQL guna membuat databasenya caranya seperti gambar di bawah ini

```
C:\> Administrator: Command Prompt - mysql -u root
Microsoft Windows [Version 10.0.14393]
(c) 2016 Microsoft Corporation. All rights reserved.

C:\Windows\system32>cd c:/xampp/mysql/bin

c:\xampp\mysql\bin>mysql -u root
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 2
Server version: 10.1.37-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.
```

Pada perintah **mysql -u root** berfungsi untuk login kedalam sistem MySQL, maksud -u adalah username dengan root sebagai usernya, karena secara *default* MySQL pada XAMPP tidak memiliki password jadi kita tidak perlu mengisi passwordnya. Setelah kita berhasil login kita dapat membuat Database latihan1 dengan perintah:

- CREATE DATABASE [nama\_database];

```
MariaDB [(none)]> create database latihan1;
Query OK, 1 row affected (0.00 sec)
```

2. Buat sebuah tabel dengan nama Biodata dengan field nama dan alamat pada database latihan1

Untuk membuat table pada database yang kita inginkan, kita terlebih dahulu harus menggunakan database tersebut dengan perintah :

- USE [nama\_database]; contoh:

```
MariaDB [(none)]> use latihan1;
Database changed
MariaDB [latihan1]>
```

Setelah kita menggunakan Database yang kita inginkan kita dapat membuat table dengan perintah:

- CREATE TABLE [nama\_table] ([field1] [type\_data]([ukuran]), [field2] [type\_data]([ukuran])); contoh:

```
MariaDB [latihan1]> create table biodata (
-> nama varchar(15),
-> alamat varchar(15)
-> );
Query OK, 0 rows affected (0.26 sec)
```

3. Tambahkan sebuah Kolom keterangan (varchar 15), sebagai kolom terakhir

Untuk menambahkan kolom / field baru pada field terakhir setelah kita membuat table kita dapat menggunakan perintah:

- ALTER TABLE [nama\_table] ADD [field] [type\_data]([ukuran]); contoh:

```
MariaDB [latihan1]> alter table biodata add keterangan varchar(15);
Query OK, 0 rows affected (0.52 sec)
Records: 0 Duplicates: 0 Warnings: 0
```

Maka secara otomatis field tersebut akan menjadi field terakhir

4. Tambahkan kolom **id** (int 11) di awal (sebagai kolom pertama)!

Sama seperti diatas namun untuk menambahkan field baru pada urutan field yang pertama kita hanya perlu menambahkan kata first pada kalimat terakhir diatas

- ALTER TABLE [nama\_table] ADD [field] [type\_data]([ukuran]) first; contoh:

```
MariaDB [latihan1]> alter table biodata add id int(11) first;
Query OK, 0 rows affected (0.40 sec)
Records: 0 Duplicates: 0 Warnings: 0

MariaDB [latihan1]> desc biodata;
+-----+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| id    | int(11) | YES |     | NULL    |       |
| nama  | varchar(15) | YES |     | NULL    |       |
| alamat | varchar(15) | YES |     | NULL    |       |
| keterangan | varchar(15) | YES |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
4 rows in set (0.01 sec)
```

5. Sisipkan sebuah kolom dengan nama **phone** (varchar 15) setelah kolom alamat!

Masih menggunakan perintah yang sama tetapi untuk kasus ini kita sisipkan kata AFTER | BEFORE [field] tergantung kita ingin menggunakan BEFORE atau AFTER di field yang kita inginkan

- ALTER TABLE [nama\_table] ADD [field] [type\_data]([ukuran]) AFTER [field];

```
MariaDB [latihan1]> alter table biodata add phone varchar(15) after alamat;
Query OK, 0 rows affected (0.33 sec)
Records: 0 Duplicates: 0 Warnings: 0

MariaDB [latihan1]> desc biodata;
+-----+-----+-----+-----+-----+-----+
| Field      | Type      | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| id         | int(11)   | YES  |     | NULL    |       |
| nama       | varchar(15) | YES  |     | NULL    |       |
| alamat     | varchar(15) | YES  |     | NULL    |       |
| phone      | varchar(15) | YES  |     | NULL    |       |
| keterangan | varchar(15) | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
5 rows in set (0.01 sec)
```

6. Ubah kolom **id** menjadi char(11)!

Untuk mengubah field yang kita inginkan kita dapat menggunakan perintah :

- ALTER TABLE [nama\_table] CHANGE [field\_lama] [field\_baru] [type\_data]([ukuran]);

```
MariaDB [latihan1]> alter table biodata change id id char(11);
Query OK, 0 rows affected (0.66 sec)
Records: 0 Duplicates: 0 Warnings: 0

MariaDB [latihan1]> desc biodata;
+-----+-----+-----+-----+-----+-----+
| Field      | Type      | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| id         | char(11)  | YES  |     | NULL    |       |
| nama       | varchar(15) | YES  |     | NULL    |       |
| alamat     | varchar(15) | YES  |     | NULL    |       |
| phone      | varchar(15) | YES  |     | NULL    |       |
| keterangan | varchar(15) | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
5 rows in set (0.01 sec)
```

7. Ubah nama kolom **phone** menjadi **hp** (varchar 20)!

Sama seperti diatas kita dapat mengubahnya menggunakan perintah:

- ALTER TABLE [nama\_table] CHANGE [field\_lama] [field\_baru] [type\_data]([ukuran]);

```
MariaDB [latihan1]> alter table biodata change phone hp varchar(20);
Query OK, 0 rows affected (0.58 sec)
Records: 0 Duplicates: 0 Warnings: 0

MariaDB [latihan1]> desc biodata;
+-----+-----+-----+-----+-----+-----+
| Field | Type      | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| id    | char(11)  | YES  |     | NULL    |       |
| nama  | varchar(15)| YES  |     | NULL    |       |
| alamat| varchar(15)| YES  |     | NULL    |       |
| hp    | varchar(20)| YES  |     | NULL    |       |
| keterangan| varchar(15)| YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
5 rows in set (0.05 sec)
```

8. Hapus kolom **keterangan** dari tabel!

Untuk menghapus field kita dapat menggunakan perintah:

- ALTER TABLE [nama\_table] drop [field\_yang\_ingin\_dihapus];

```
MariaDB [latihan1]> alter table biodata drop keterangan;
Query OK, 0 rows affected (0.30 sec)
Records: 0 Duplicates: 0 Warnings: 0

MariaDB [latihan1]> desc biodata;
+-----+-----+-----+-----+-----+-----+
| Field | Type      | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| id    | char(11)  | YES  |     | NULL    |       |
| nama  | varchar(15)| YES  |     | NULL    |       |
| alamat| varchar(15)| YES  |     | NULL    |       |
| hp    | varchar(20)| YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
4 rows in set (0.02 sec)
```

9. Ganti nama tabel menjadi **data\_mahasiswa**!

Untuk mengganti nama table kita masih menggunakan perintah alter table dengan kondisi:

- ALTER TABLE [nama\_table] RENAME TO [nama\_tabel\_baru];

```
MariaDB [latihan1]> alter table biodata rename to data_mahasiswa;
Query OK, 0 rows affected (0.32 sec)

MariaDB [latihan1]> desc data_mahasiswa;
+-----+-----+-----+-----+-----+-----+
| Field | Type      | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| id    | char(11)  | YES  |     | NULL    |       |
| nama  | varchar(15)| YES  |     | NULL    |       |
| alamat| varchar(15)| YES  |     | NULL    |       |
| hp    | varchar(20)| YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
4 rows in set (0.13 sec)
```

## 10. Ganti nama **field** id menjadi **nim**!

Disini kita menggunakan perintah yang sama saat mengganti type data id yaitu menggunakan perintah alter table dengan kondisi:

- ALTER TABLE [nama\_table] CHANGE [field\_lama] [field\_baru] [type\_data]([ukuran]);

```
MariaDB [latihan1]> alter table data_mahasiswa change id nim char(11);
Query OK, 0 rows affected (0.09 sec)
Records: 0 Duplicates: 0 Warnings: 0

MariaDB [latihan1]> desc data_mahasiswa;
+-----+-----+-----+-----+-----+-----+
| Field | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| nim   | char(11)      | YES  |     | NULL    |       |
| nama  | varchar(15)   | YES  |     | NULL    |       |
| alamat | varchar(15)   | YES  |     | NULL    |       |
| hp     | varchar(20)   | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
4 rows in set (0.02 sec)
```

## 11. Jadikan **nim** sebagai primary key!

Dengan menjadikan nim sebagai primary key berarti kita telah menyetujui bahwa nim adalah kunci utama atau field unik yang datanya tidak boleh ada yang sama. Cara menambahkan primary key dapat menggunakan perintah berikut:

- ALTER TABLE [nama\_table] ADD PRIMARY KEY (nama\_field);

```
MariaDB [latihan1]> alter table data_mahasiswa add primary key (nim);
Query OK, 0 rows affected (0.59 sec)
Records: 0 Duplicates: 0 Warnings: 0

MariaDB [latihan1]> desc data_mahasiswa;
+-----+-----+-----+-----+-----+-----+
| Field | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| nim   | char(11)      | NO   | PRI | NULL    |       |
| nama  | varchar(15)   | YES  |     | NULL    |       |
| alamat | varchar(15)   | YES  |     | NULL    |       |
| hp     | varchar(20)   | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
4 rows in set (0.03 sec)
```

## EVALUASI DAN PERTANYAAN

1. Tulis semua perintah-perintah SQL percobaan di atas beserta outputnya!

Perintah-perintah SQL dan Outputnya sudah kita bahas di atas

2. Apa maksud dari int (11)?

Maksud dari int(11) adalah sebuah type data yang digunakan untuk menyimpan data bilangan bulat positif dan negatif, sedangkan angka 11 yang ada di dalam kurung menandakan length atau ukuran maksimal sebuah data tersebut.

3. Ketika kita melihat struktur tabel dengan perintah desc, ada kolom **Null** yang berisi **Yes** dan **No**. Apa maksudnya ?

Maksudnya adalah apabila Kolom Null berisi **Yes** maka data yang ada di kolom tersebut atau field tersebut dapat kosong atau tidak ada isinya, sedangkan apabila berisi **No** maka data yang ada di field/kolom tersebut tidak boleh kosong.

4. Buat laporan praktikum beserta langkah-langkah yang dilakukan dalam bentuk tutorial beserta screenshot outputnya (point 1)

Langkah-langkah atau tutorial dalam bentuk video sudah saya upload di channel youtube pribadi saya, kalian dapat menontonnya dengan klik link berikut:

[Kutt.it/viotutormysql1](https://www.youtube.com/channel/UCkuttitvotutormysql1)

5. Upload hasilnya pada calssroom.