**LAPORAN PRAKTIKUM**

Nama Matakuliah : DATABASE SYSTEMS

SKS : 4

Pertemuan ke : 2

**Nama Mahasiswa** **: Jeffry Heriyanto**

**NPM** **: 230210003**

1. **Materi / Topik /**
2. Pengenalan DBMS
3. Membuat database dengan mysql
4. **Sintak SQL dan Analisis**
5. Langkah-langkah **koneksi database** dengan **comand prompt**

* Buka **XAMPP** aktifkan **MySQL**
* Buka **Commad Prompt** (**CMD**)
* Pindah ke direkotri C dengan perintah

**cd C:\**

* Buka direktori Xampp yang ada di C dengan perintah:

**cd xampp/mysql/bin**

* Untuk membuka koneksi SQL dengan perintah :

**mysql** **-u** **root**

1. Membuat sebuah database
   1. **Create database kuliah**;

Merupakan perintah utuk membuat sebuah database dengan nama **kuliah**

* 1. **Show databases**;

Merupakan perintah untuk menampilkan database yang telah dibuat

* 1. **Use kuliah**;

Merupakan perintah untuk masuk kedalam database **kuliah**

1. Membuat tabel

**CREATE TABLE user**

(

id **int unsigned NOT NULL auto\_increment,**

username **varchar(100) NOT NULL,**

email **varchar(100) NOT NULL,**

**PRIMARY KEY (id)**

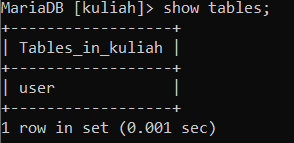
);

**Analisis koding:**

* **Create table** merupakan perintah untuk membuat sebuah **tabel** dengan nama **user**
* **id** **int unsigned NOT NULL auto\_increment** merupakan perintah yang digunakan untuk membuat field id dengan tipe data integer,
* **unsigned** menyatakan bahawa id tidak boleh bernilai negatif,
* **not null** merupakan perintah bahwa **id** tidak boleh kosong saat di input
* **auto\_increment** merupakan perintah id akan dibuat secara otomatis oleh sistem
* **varchar** merupakan sebuah perintah untuk mengubah tipe data username menjadi varchar (dapat menampung alfanumerik)
* **(100)** adalah jumlah bit data yang diperbolehkan atau jumlah data maksimal dalam sebuah field contohnya adalah, jika di set (100) maka **username hanya bisa** **menampung 100 karakter saja**
* **Primary key** berfungsi sebagai sebuah kunci untuk mengidentifikasi jumlah record yang ada dalam suatu tabel.

1. Menampilkan tabel yang telah dibuat

**Show tables;**

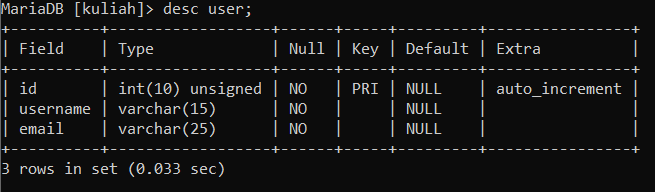


1. Untuk menampilkan struktur tabel

**Describe user;**

atau

**desc user;**

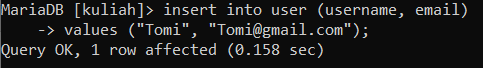


Menampilkan semua variable yang telah dibuat saat pertama kali dibuat, ini akan memperjelas jika terjadi kesalahan pengkonfigurasian dalam proses pemograman.

1. Memasukan data kedalam sebuah tabel

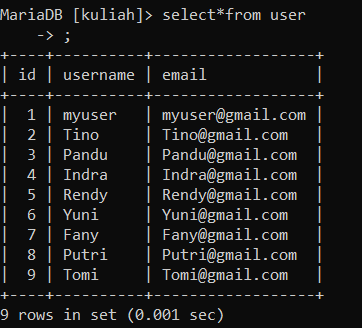
**Insert into user (username, email)**

**Values (“Tomi”, “Tomi@gmail.com”);**



1. Menampilkan isi data yang ada dalam tabel

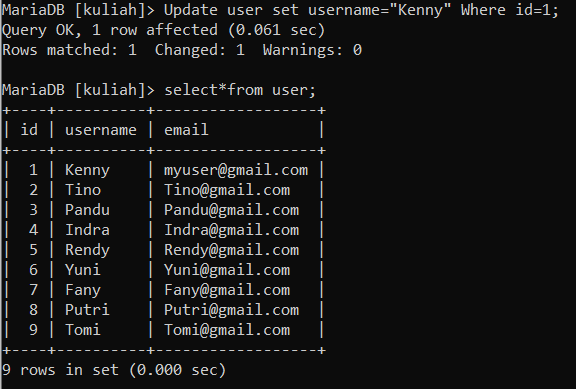
**Select\*from user;**



Tomi telah berhasil ditambahkan kedalam tabel **user**

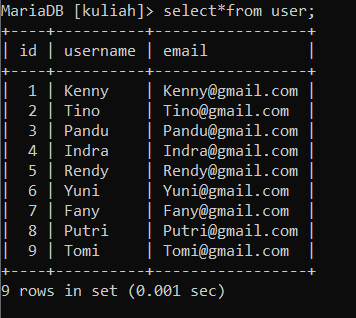
1. Update data tertentu yang ada di dalam tabel

**Update user set username=”Kenny” Where id=1;**



**Myuser** telah diganti menjadi **Kenny** dan email tetap **myuser@gmail.com** karena tidak ada perintah untuk mengupdate kolom tersebut.

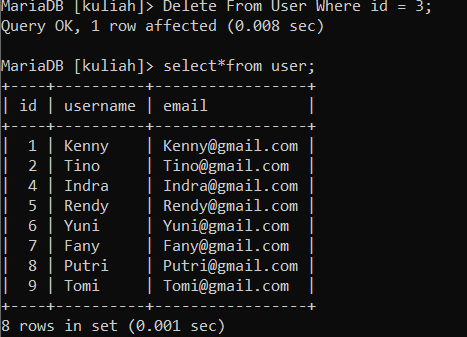
**Update user set email=”Kenny@gmail.com” Where id=1;**



Perintah ini akan berefek hanya kepada id “1” dan mengupdate data yang ada pada kolom email saja.

1. Menghapus data dalam tabel

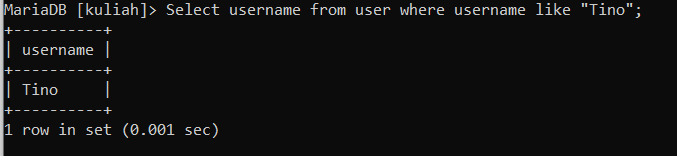
**Delete From User Where id = 3;**



Id “3” akan terhapus dalam tabel user, dan id yang lain tidak akan terupdate untuk mengisi id yang terdelete. Maka akan terjadi pelompatan penomoran dalam tabel user.

1. Untuk mencari data berdasarkan karakter tertentu

**Select username from user where username like “Tino”;**



Perintah ini akan menampilkan data apa saja yang ada dalam **field username** yang cocok dengan “**Tino**”.