Modul Praktikum

SISTEM BASIS DATA



Dosen Pengampu:

Ahmad Fashiha Hastawan, S.T., M.Eng 198802102018031001

Program Studi Pendidikan Teknik Informatika dan Komputer Jurusan Teknik Elektro, Fakultas Teknik Universitas Negeri Semarang

TUTORIAL MYSQL (Part 1)

Database dan Tabel dengan Command Prompt

I. Tujuan Instruksial Khusus

Setelah melakukan Praktikum ini mahasiswa dapat:

- 1. Melakukan instalasi software Ms Visio untuk membuat ER-Diagram
- 2. Menggunakan software Ms Visio untuk membuat ER-Diagram
- 3. Memahami dan mempraktikkan pembuatan ER-D dengan software Ms Visio

II. Dasar Teori

a) MySQL

MySQL merupakan database *server open source* yang cukup popular keberadaannya. Dengan berbagai keunggulan yang dimiliki, membuat software database ini banyak digunakan oleh para praktisi untuk membangun suatu project. Adanya fasilitas API (Aplication Programming Interface) yang dimiliki oleh MySQL, memungkinkan bermacam-macam aplikasi komputer yang ditulis berbagai bahasa pemrograman dapat mengakses basis data MySQL.

b) Database

Database adalah seluruh struktur yang umumnya terbagi dalam dua hal, yaitu database flat dan database relasional. Database relasional lebih mudah dipahami daripada database flat karena database relasional mempunyai bentuk yang sederhana serta mudah dilakukan operasi data. Database yang memiliki struktur relasional terdapat tabel-tabel untuk menyimpan data. Pada setiap tabel terdiri dari kolom dan baris serta sebuah kolom untuk mendefenisikan jenis informasi apa yang harus disimpan.

Database akan menjadi sangat berguna saat Anda perlu menyimpan informasi yang dikategorikan secara logis. Contoh, jika Anda ingin menyimpan informasi tentang PT. Electric E11 dengan database, Anda bisa mengelompokkan berbagai hal dalam bisnis menajdi beberapa tabel

c) Database Relational

Ketika Anda menggunakan software sistem manajemen database terkomputerisasi, umumnya menggunakan database relasional. Prinsip database relasioanal adalah informasi dibagi menjadi beberapa data yang terpisah secara logis. Data—data yang terpisah tersebut diletakkan dalam bentuk tabel. Tabel adalah objek dasar yang merupakan jantung dari database relasional. Tabel adalah dasar penyimpanan informasi dan pengambilannya (retrieval). Ketika informasi sudah tersimpan di tabel-tabel yang terpisah, Anda nantinya dapat melihat (*view*), mengeit (*edit*), menambah (*Add*), dan menghapus (delete) informasi dengan berbagai metode. Selain itu, Anda juga bisa mengambil informasi menggunakan query dan menampilkan informas menggunakan report.

Keunggulan penyimpanan data menggunakan databes relasional sangat banyak dibandingkan dengan penyimpanan satu tabel ukuran besar dua dimensi (yang disebut file flat) seperti di dokumen Word atau spreadsheet Excel. Salah satu keunggulan database adalah berkurangnya rendundansi data. Konsekuensinya tidak hanya ruang penyimpanan harddisk menjadi berkurang, tetapi kecepatan pemrosesan data juga berkurang. Selain itu, ada kelebihan lainnya yaitu:

1) Fleksibiltas

Jika data berubah, Anda dapat meng-update nilainya hanya di satu tempat. Dengan demikian, semua query, form, dan lainnya akan berubah secara konsekuen dengan nilai yang baru tersebut.

2) Simple

Modal penyimpanan tabel yang merupakan dasar dari sistem relasinal memang simple, dan merupakan metode penyimpanan data yang tak redundan. Tiap tabel didesain relasion untuk objek tunggal yanng mengandung data, yang konsekuen terhadap aspek tertentu dari database, seperti pegawai produk, atau order.

3) Power

Menyimpan data dalam tabel-tabel yang terpisah lebih memudahkan adanya pengelompokan, searching, dan pengambilan data yang menggunakan banyak cara yang tak terbatas.

4) Kemudahan manajemen

Dengan tabel yang kecil dan tidak kompleks, informasi jadi muah dilacak dan diatur. Misalnya Anda menggunakan database untuk menyimpan data order dari penggalan. Jika menggunakan tabel tunggal, ada banyak record yang dibuat untuk tiap order yang dilakukan pelanggan. Dengan demikian, tiap kali ada order, tiap kali pula sebuah informasi dibuat, walaupun pelanggannya sama. Karena itu, jika suatu saat ada informasi yang berubah (misalnya informasi pelanggan), semua record yang mengandung informasi pelanggan harus diubah. Ini akan sangat merepotkan. Begitu pula informasi tentang pegawai pemrosesan order juga akan terus berulang. Akan lebih efisien seandainya ada satu tabel pelanggan dan satu tabel untuk pegawai. Kemudian, ada field identitas pelanggan yang di-tambahkan ke tabel pelanggan order sehingga ada hubungan/koneksi antar keduanya. Hubungan ini disebut *relationship*. Begitu pula idenitas pegawai juga bisa dihubungkan dengan tabel order menggunakan relationship.

d) Tabel

Tabel adalah inti konsep database. Tujuannya adalah menyimpan informasi. Tabel satu dengan yang lainnya bisa dihubungkan. Satu database bisa mengandung banyak tabel juga umumnya tak dibatasi, batasnya hanya pada kapasitas disk yang digunakan. Tiap tabel memiliki beberapa kolom, kolom yang mengandung nilai-nilai apa yang bisa disimpan. Misalnya, tabel yang menyimpan informasi produk perlu menyimpan data seperti nama, harga, dan berat produk. Setiap baris dari tabel bisa mengandung nilai untuk kolom-kolom yang ditentukan di tabel. Tipe data menunjukkan jenis data yang bisa disimpan. Tipe data bisa dibatasi seperti jenis numerik atau tanggal, atau lain sebagainya.

e) Query

Anda dapat mengambil informasi tertentu yang disimpan di tabel atau multitable menggunakan query. Untuk membuat query, Anda harus menentukan parameter-parameter dari infirmasi yang ingin di cari. Misalnya, Anda ingin mengambil data penlanggan yang telah membeli barang tipe X selama 3 bulan yang lalu. Dengan demikian, Anda harus mengisikan variabel nama barang dan periode. Informasi yang disimpan juga nantinya bisa diurutkan, di filter dan bisa diatur cara mnampilkannya.

III. Alat dan Bahan

- 1. PC Processor minimal Intel i3, RAM 2GB
- 2. Sistem Operasi Windows
- 3. Xampp, MySql

IV. Langkah Kerja/Percobaan

A. Memulai MySQL XAMPP PhpMyAdmin

Buka XAMPP Control Panel kemudian klik start pada MySQL



Buka jendela command Prompt (cmd), kemudian arahkan direktori ke dalam direktori C:\xampp\mysql\bin



Kemudian ketikkan perintah **mysql –u root –p**. sehingga muncul pesan masukkan password. Karena password masih default dan belum dilakukan perubahan maka tekan enter.

```
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 2
Server version: 10.1.25-MariaDB mariadb.org binary distribution
Copyright (c) 2000, 2017, Oracle, MariaDB Corporation Ab and others.
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.
MariaDB [(none)]>
```

B. Membuat Database

Format penulisan query untuk membuat database adalah sebagai berikut:

CREATE DATABASE [IF NOT EXISTS] nama database

Contoh :

Untuk membuat sebuah database mahasiswa, querynya adalah sebagai berikut:

```
mysql> CREATE DATABASE mahasiswa;
Query OK, 1 row affected (0.00 sec)
```

```
MariaDB [(none)]> create database mahasiswa;
Query OK, 1 row affected (0.04 sec)
```

Keterangan:

Tambahan query **[IF NOT EXISTS]** digunakan untuk membuat MySQL tidak menampilkan pesan error jika database tersebut telah ada sebelumnya dalam database. Jika database yang dibuat belum ada maka query CREATE DATABASE akan membuat database baru sesuai query, namun jika database sudah ada maka CREATE DATABASE tidak akan menghasilkan database baru (database yang lama juga tidak akan tertimpa).

Contohnya, jika kita menjalankan lagi query untuk membuat database mahasiswa, MySQL akan menampilkan pesan error.

Contoh :

```
mysql> CREATE DATABASE mahasiswa;
ERROR 1007 (HY000): Can't create database 'mahasiswa';
database exists
```

```
MariaDB [(none)]> create database mahasiswa;
ERROR 1007 (HY000): Can't create database 'mahasiswa'; database exists
MariaDB [(none)]> _
```

Keterangan:

Munculnya pesan error dalam eksekusi query dapat menyebabkan proses eksekusi query terhenti dan dalam system yang menggunakan database maka proses akan menemukan kendala dalam memproses data.

Berikut ini contoh apabila database sudah ada namun akan dibuat database baru dengan nama yang sama dan menggunakan perintah IF NOT EXISTS.

mysql> CREATE DATABASE IF NOT EXISTS mahasiswa; Query OK, 1 row affected, 1 warning (0.00 sec)

```
MariaDB [(none)]> create database mahasiswa;
ERROR 1007 (HY000): Can't create database 'mahasiswa'; database exists
MariaDB [(none)]> create database IF NOT EXISTS mahasiswa;
Query OK, 0 rows affected, 1 warning (0.00 sec)
```

MariaDB [(none)]>

C. Melihat daftar database yang sudah dibuat

Query untuk melihat seluruh database yang berada pada MySQL Server adalah

sebagai berikut:

SHOW DATABASES;

Berikut adalah contoh tampilannya:

```
mysql> SHOW DATABASES;
+----+
| Database |
+----+
| information_schema |
| mahasiswa |
| mysql |
| performance_schema |
| test |
+----+
```

5 rows in set (0.00 sec)



Kita dapat melihat bahwa MySQL Server telah berisi beberapa database bawaan, selain database mahasiswa yang kita buat sebelumnya.

D. Cara memilih dan menggunakan database MySQL

Format query untuk memilih database ini adalah:

```
USE nama database;
```

Catatan:

nama_database adalah database yang akan kita gunakan. Jika kita ingin menggunakan database mahasiswa, maka querynya adalah:

```
mysql> USE mahasiswa;
Database changed
```

```
MariaDB [(none)]≻ use mahasiswa
Database changed
MariaDB [mahasiswa]≻ _
```

E. Cara Menghapus Database MySQL

Format query untuk menghapus database:

DROP DATABASE [IF EXISTS] database name;

Sama seperti query pada pembuatan database, pilihan **[IF EXISTS]** digunakan untuk menghilangkan pesan error jika seandainya database tersebut memang tidak ada. Jika kita ingin menghapus database mahasiswa, contoh querynya adalah:

mysql> DROP DATABASE mahasiswa; Query OK, 0 rows affected (0.01 sec)

```
MariaDB [mahasiswa]> DROP DATABASE mahasiswa
-> ;
Query OK, 0 rows affected (0.48 sec)
```

MariaDB [(none)]> _

Jika kita berusaha menghapus database mahasiswa lagi (yang memang sudah tidak ada), akan muncul pesan error:

```
mysql> DROP DATABASE mahasiswa;
ERROR 1008 (HY000): Can't drop database 'mahasiswa';
database doesn't exist
```

MariaDB [(none)]≻ DROP DATABASE mahasiswa; ERROR 1008 (HY000): Can't drop database 'mahasiswa'; database doesn't exist MariaDB [(none)]≻

Dengan menggunakan perintah opsional [IF EXISTS], pesan error tidak tampil:

```
mysql> DROP DATABASE IF EXISTS mahasiswa;
Query OK, 0 rows affected, 1 warning (0.00 sec)
```

```
MariaDB [(none)]> DROP DATABASE IF EXISTS mahasiswa;
Query OK, 0 rows affected, 1 warning (0.00 sec)
```

MariaDB [(none)]>

F. Membuat Tabel

Format dasar *query* untuk membuat tabel adalah:

```
CREATE TABLE [IF NOT EXISTS] nama_tabel (daftar_kolom) [type=tipe tabel]
```

- Query opsional [IF NOT EXISTS] sama seperti pada query pembuatan database, digunakan untuk membuat MySQL *tidak menampilkan pesan error* jika tabel tersebut telah ada sebelumnya dalam database.
- Nama_tabel adalah nama dari tabel yang ingin kita buat.
- Daftar_kolom berisi definisi dari kolom-kolom yang akan kita buat untuk database ini.

Rancangan tabel yang akan dibuat di dalam database mahasiswa adalah tabel dengan nama data_mahasiswa dengan struktur sebagai berikut:

No	Field	Tipe Data	Length/Values	Keterangan
1	nim	Varchar	15	Primary Key
2	nama	Varchar	40	
3	tempat_lahir	Varchar	35	
4	tgl_lahir	Date		
5	alamat	Varchar	50	
6	no_telp	Varchar	30	

Query:

```
CREATE TABLE IF NOT EXISTS data_mahasiswa (nim varchar(15),
nama varchar(40), tempat_lahir varchar(35), tgl_lhr date,
alamat varchar(50), no_telp varchar(30));
Query OK, 0 rows affected (0.38 sec)
```

MariaDB [(none)]≻ use mahasiswa

```
Database changed
MariaDB [mahasiswa]> CREATE TABLE IF NOT EXISTS data_mahasiswa (nim varchar(15),
nama varchar(40), tempat_lahir varchar(35), tgl_lhr date, alamat varchar(50), n
o_telp varchar(30));
Query OK, 0 rows affected (0.38 sec)
```

Query ok, o rows arrected (0.5

MariaDB [mahasiswa]≻

G. Melihat Daftar Tabel

Untuk melihat seluruh tabel yang ada pada database aktif saat ini, format querynya adalah:

SHOW TABLES;

```
MariaDB [mahasiswa]> show tables;
+-----+
| Tables_in_mahasiswa |
+-----+
| data_mahasiswa |
+----+
1 row in set (0.00 sec)
MariaDB [mahasiswa]> _
```

H. Melihat Struktur Tabel

Untuk melihat struktur dari sebuah tabel, seperti nama kolom yang tersedia, dan tipenya, dapat dilihat dengan format query:

```
DESCRIBE nama tabel;
```

Field	Туре	Null	Key	Default	Extra
nim	varchar(15)	YES		NULL	
nama	varchar(40)	YES		NULL	
tempat_lahir	varchar(35)	YES		NULL	
tgl_lhr	date	YES		NULL	
alamat	varchar(50)	YES		NULL	
no_telp	varchar(30)	YES		NULL	
+	+	+	++	+	+

I. Cara menghapus tabel

Query untuk menghapus tabel adalah:

```
DROP TABLE[IF EXISTS] table_name [, table_name,...]
```

Query opsional **[IF EXISTS]** digunakan untuk menghilangkan pesan error jika tabel yang akan dihapus memang sudah tidak ada.

Dari format query **DROP TABLE**, dapat dilihat bahwa dalam satu query, kita dapat menghapus *beberapa tabel* sekaligus.

```
MariaDB [mahasiswa]> DROP TABLE iF EXISTS data_mahasiswa;
Query OK, 0 rows affected (0.57 sec)
MariaDB [mahasiswa]> show tables;
Empty set (0.00 sec)
MariaDB [mahasiswa]> _
```

V. Lembar Kerja

1) Silahkan Anda mencoba seluruh rangkaian praktikum di atas

VI. Tugas dan Pertanyaan

Silahkan anda membuat basis data relasional pada kasus tugas yang ada pada slide materi pertemuan hari ini, screenshoot hasil tabel yang Anda buat dengan konten:

- 1) Screenshoot tabel yang anda buat
- 2) Screenshoot struktur per tabel
- 3) Screenshoot isi data per tabel