

MODUL 1

Pengantar MySQL

A. Tujuan :

1. Memahami tentang sistem database
2. Memahami dasar-dasar MySQL.
3. Memahami relasi di MySQL

B. Dasar Teori

Basis Data:

Himpunan kelompok data (arsip) yang saling berhubungan, yang diorganisasi sedemikian rupa, sehingga kelak dapat dimanfaatkan kembali dengan cepat.

Komponen Sistem Basis Data:

- a. Perangkat Keras (Hardware)
Komputer, memori, storage (Harddisk), peripheral, dll.
- b. Sistem Operasi (Operating System)
Program yang menjalankan sistem komputer, mengendalikan resource komputer dan melakukan berbagai operasi dasar sistem komputer.
- c. Basis Data (Database)
Menyimpan berbagai obyek database (struktur tabel, indeks,dll)
- d. DBMS (Database Management System)
Perangkat lunak yang memaintain data dalam jumlah besar.
- e. Pemakai (User)
Para pemakai database.
- f. Aplikasi (perangkat lunak) lain.
Program lain dalam DBMS.

Bahasa Basis Data

- DBMS merupakan perantara antara user dengan database.
- Cara komunikasi diatur dalam suatu bahasa khusus yang telah ditetapkan oleh DBMS.
Contoh: SQL, dBase, QUEL, dsb.
- Bahasa database, dibagi dalam 2 bentuk:
 - Data Definition Language (DDL)
Digunakan dalam membuat tabel baru, indeks, mengubah tabel, menentukan struktur tabel, dsb.
 - Data Manipulation Language (DML)
 - a. Digunakan dalam memanipulasi dan pengambilan data pada database.
 - b. 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:

Tabel 1. Tipe data di MySQL

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
LOBLOB	Teks dengan panjang maksimum 4294967295 karakter

Untuk masuk ke dalam program MySQL pada prompt jalankan perintah berikut ini:

C:\>Program Files\xampp\mysql\bin>mysql –u root –p (Enter)

Kemudian akan masuk kedalam MySQL seperti tampilan dibawah ini:

```

C:\WINDOWS\system32\cmd.exe - mysql -u root -p
C:\Program Files\xampp\mysql\bin>mysql -u root -p
Enter password: ****
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 8 to server version: 5.0.18-nt

Type 'help;' or '\h' for help. Type '\c' to clear the buffer.

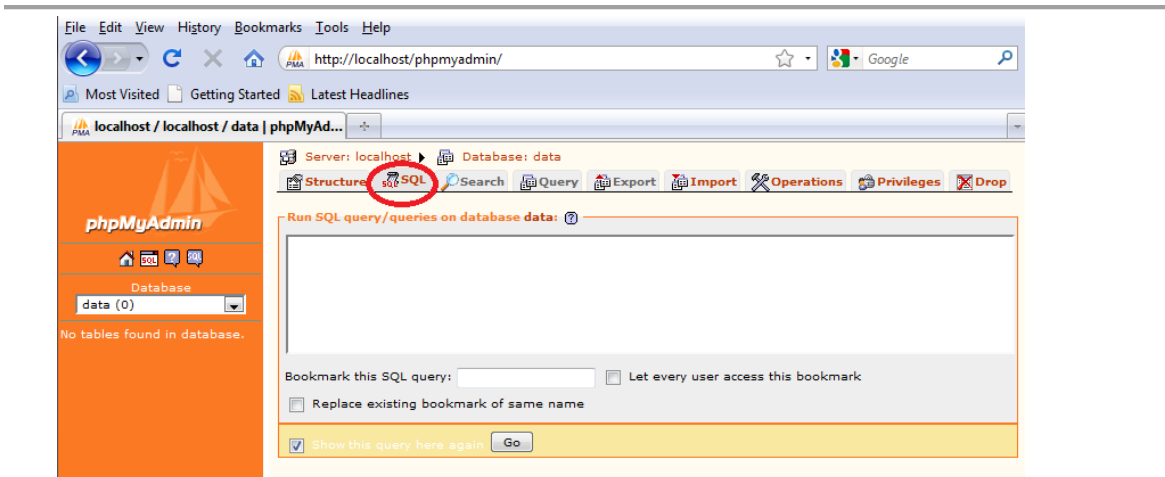
mysql>

```

Gambar 1. Tampilan mysql di command prompt

Bentuk prompt “mysql>” adalah tempat menuliskan perintah-perintah MySQL. Setiap perintah SQL harus diakhiri dengan tanda titik-koma “;” . Atau juga bisa dari phpmyadmin, kemudian pilih tab SQL untuk mengetikkan perintah-perintah DML.

Ketikkan : <http://localhost/phpmyadmin>



Gambar 1. Tampilan mysql di phpmyadmin

Membuat Database Dan Table

Cara untuk membuat sebuah database baru adalah dengan perintah:

```
CREATE DATABASE namadatabase;
```

Contoh:

```
CREATE DATABASE pens;
```

Untuk membuka sebuah database dapat menggunakan perintah berikut ini:

```
USE namadatabase;
```

Contoh:

```
USE pens;
```

Perintah untuk membuat tabel baru adalah:

```
CREATE TABLE namatabel (  

                    struktur  

);
```

Constraint

Constraint adalah batasan atau aturan yang ada pada table. MySQL menyediakan beberapa tipe constraint berikut :

- NOT NULL

Suatu kolom yang didefinisikan dengan constraint NOT NULL tidak boleh berisi nilai NULL. Kolom yang berfungsi sebagai kunci primer (primary key) otomatis tidak boleh NULL.

- UNIQUE

Mendefinisikan suatu kolom menjadi bersifat unik, artinya antara satu data dengan data lainnya namanya tidak boleh sama, misal alamat email.

- PRIMARY KEY

Constraint PRIMARY KEY membentuk key yang unik untuk suatu table.

- FOREIGN KEY

FOREIGN KEY constraint didefinisikan pada suatu kolom yang ada pada suatu table, dimana kolom tersebut juga dimiliki oleh table yang lain sebagai suatu PRIMARY KEY, biasa dipakai untuk menghubungkan antara 2 tabel.

Contoh Pembuatan Tabel :

Misalkan kita ingin menyimpan data anggota yaitu: nomor, nama, email, alamat, kota. Sedangkan strukturnya seperti tabel dibawah ini :

Tabel 2. Pembuatan tabel

Kolom/Field	Tipe data	Keterangan
nomor	int(6) not null primary key	angka dengan panjang maksimal 6, sebagai <i>primary key</i> , tidak boleh kosong
nama	char(40) not null	teks dengan panjang maksimal 40 karakter, tidak boleh kosong
email	char(255) not null	teks dengan panjang maksimal 255 karakter, tidak boleh kosong
alamat	char(80) not null	teks dengan panjang maksimal 80 karakter, tidak boleh kosong
kota	char(20) not null	teks dengan panjang maksimal 20 karakter, tidak boleh kosong

Perintah MySQL untuk membuat tabel seperti diatas bisa dengan 2 cara sbb :

```
create table anggota (
    nomor int(6) not null primary
    key, nama char(40) not null,
    email char(255) not
    null, alamat char(80)
    not null, kota char(20)
    not null
);
```

```
create table anggota ( nomor
    int(6) not null, nama
    char(40) not null, email
    char(255) not null, alamat
    char(80) not null, kota
    char(20) not null, primary
    key (nomor)
);
```

Perintah-Perintah Dasar SQL

A. Pemasukan data

Untuk memasukkan sebuah baris (record) kedalam tabel MySQL adalah sebagai berikut:

```
INSERT INTO table [(column1, column2,...)]
VALUES (value1, value2,...);
```

Contoh :

```
INSERT INTO anggota (nomor,nama,email,alamat,kota) VALUES
(100,'Adi','adi@yahoo.com','Jl. Keputih 2A no 5','Surabaya');
```

atau

```
INSERT INTO anggota VALUES (100,'Adi','adi@yahoo.com','Jl. Keputih 2A no 5',
'Surabaya');
```

Latihan :

Masukkan data-data berikut ini ke tabel anggota :

Nomor	Nama	Email	Alamat	Kota
100	Adi	adi@yahoo.com	Jl. Keputih 2A no 5	Surabaya
101	Arif	arif@hotmail.com	Jl. Gebang Lor 32D	Surabaya
102	Iqbal	iqbal@yahoo.com	Jl. Klampis 3	Surabaya
103	Setyo	setyo@mailcity.com	Jl. Madiun 5	Sidoarjo
104	Ilham	ilham@yahoo.com	Jl. Surabaya 9	Malang
105	Syamsudin	sam@yahoo.com	Jl. Pabean 11	Surabaya
106	Farug	farug@yahoo.com	Jl. Raya 5	Kediri
107	Hari	hari@yahoo.com	Jl. Raya 9	Banyuwangi

B. Menghapus Record

Untuk menghapus suatu record dengan kondisi tertentu digunakan perintah sebagai berikut:

```
DELETE [FROM] table [WHERE condition];
```

Contoh :

- a. Untuk menghapus record dengan no = 107

```
DELETE FROM anggota WHERE no=107;
```

- b. Untuk menghapus seluruh record

```
DELETE FROM anggota;
```

C. Memodifikasi Record

Untuk memodifikasi (merubah) isi record tertentu adalah dengan menggunakan perintah sebagai berikut:

```
UPDATE      table  
SET         column = value [, column = value, ...]  
[WHERE      condition];
```

Contoh :

Untuk meng-update data email di tabel anggota, nomor 107.

```
UPDATE anggota SET email='hari_s@hotmail.com' WHERE nomor=107;
```

D. Menampilkan Isi Tabel

Bentuk umum :

```
SELECT [DISTINCT] < attribute-list >  
FROM      < table-list >  
WHERE     < condition >
```

Latihan :

1. Untuk menampilkan semua kolom(field) pada tabel anggota

```
SELECT * FROM anggota;
```

2. Untuk menampilkan kolom (field) nomor dan nama pada tabel anggota

SELECT nomor, nama FROM anggota;

3. Untuk menampilkan semua kolom pada tabel anggota yang berada pada kota 'Surabaya'

SELECT * FROM anggota WHERE kota='Surabaya';

4. Untuk menampilkan semua kolom pada tabel anggota denganurut nama

SELECT * FROM anggota ORDER BY nama;

5. Untuk menghitung jumlah record pada tabel anggota

SELECT COUNT(*) FROM anggota;

6. Untuk menampilkan kota dengan tidak menampilkan kota yang sama pada tabel anggota

SELECT DISTINCT kota FROM anggota;

7. Untuk menampilkan nama dan email yang mempunyai email di 'yahoo.com'

SELECT nama,email FROM anggota WHERE email like '%yahoo.com';

8. Untuk menampilkan nomor, nama dan email yang nomornya diatas 103 dan yang berawalan dengan huruf S.

SELECT nomor, nama, email FROM anggota WHERE nomor >= 103 AND nama LIKE 'S%';

9. Untuk menampilkan nomor, nama yang nomornya diantara 103 ~ 105

SELECT nomor, nama FROM anggota WHERE nomor BETWEEN 103 AND 105.

C. Percobaan

1. Buatlah database baru dengan nama lat_mysql
2. Buatlah tabel dengan nama pegawai yang memiliki field sebagai berikut :

Kolom / Field	Tipe data
ID	int (5) not null
Nama	varchar(20) not null
Alamat	varchar(20) not null
Gaji	int (10) not null

3. Masukkan data-data berikut ini :

ID	Nama	Alamat	Gaji
100	Arif	Surabaya	10000
101	Andi	Jakarta	14000
102	Burhan	Malang	12000
103	Fikri	Madiun	15000
104	Fariz	Malang	17000
105	Sigit	Surabaya	20000
106	Ifan	Kediri	16000
107	Hanif	Yogyakarta	12000
108	Zakiuddin	Surabaya	21000

4. Hapuslah record dengan ID 100.

5. Update data untuk ID 101 dan 102 sbb :

ID	Nama	Alamat	Gaji
101	Naila	Surabaya	10000

102	Rafi	Semarang	14000
-----	------	----------	-------

6. Buat perintah SQL berikut ini untuk menampilkan :
- Semua kolom dari tabel pegawai
 - Kolom nama dan gaji dari tabel pegawai
 - Kolom nama dan alamat yang mempunyai nama dengan awalan F.
 - Kolom nama dan alamat yang mempunyai nama dengan akhiran n
 - Kolom nama dan gaji yang mempunyai gaji diatas 15000.
 - Kolom nama, alamat dan gaji yang bergaji diatas 13000 dan beralamat di Surabaya.
 - Kolom nama, gaji dengan range gaji antara 15000 ~ 20000.
 - Gaji max, min,rata2 dari tabel pegawai > *select max(gaji), min(gaji), avg(gaji) from pegawai.*

Tugas:

1. Buatlah 2 tabel yaitu **Pegawai** ,**Departemen** yang saling berhubungan yang memiliki field-field sbb :

a. Tabel Pegawai

Kolom / Field	Tipe data	Keterangan
ID_Peg	int (5) not null	Primary Key
Nama_Peg	varchar(20) not	
Alamat	varchar(20) not	
ID_Dept	varchar (5) not null	Foreign Key

b. Tabel Departemen

Kolom / Field	Tipe data	Keterangan
ID_Dept	varchar (5) not null	Primary Key
Nama_Dept	varchar(20) not	

2. Isikan data untuk kedua tabel diatas :

a. Tabel Pegawai

ID_Peg	Nama_Peg	Alamat	ID_Dept
100	Arif	Surabaya	A001
101	Andi	Jakarta	A002
102	Burhan	Malang	A001

b. Tabel Departemen

ID_Dept	Nama_Dept
A001	IT
A002	Produksi
A003	HRD
A004	Engineering

3. Buatlah perintah SQL berikut :

- Menampilkan nama pegawai, alamat pegawai dan nama departemen.
- Menampilkan id pegawai, nama pegawai dan nama departemen.