

MODUL I

MYSQL & DATA DEFINITION LANGUAGE (DDL)

1. TUJUAN

Setelah menyelesaikan modul ini, anda diharapkan dapat :

- Membuat database dan tabel dengan data definition language
- Mampu memodifikasi tabel
- Mengetahui lingkungan kerja MySQL
- Mengetahui format perintah di MySQL
- Mengetahui perintah-perintah sederhana di MySQL

2. TEORI DASAR

MySQL adalah suatu perangkat lunak database relasi (Relational Database Management System atau RDBMS), seperti halnya ORACLE, PostgreSQL, MS SQL, dan sebagainya. MySQL dikembangkan sekitar tahun 1994 oleh sebuah perusahaan pengembang software dan konsultan database bernama MYSQL AB yang berada di Swedia. Waktu itu perusahaan tersebut masih bernama TcX DataKonsult AB, dan tujuan awal dikembangkannya MySQL adalah untuk mengembangkan aplikasi berbasis web pada client. MySQL menyebut produknya sebagai database open source terpopuler di dunia. Berdasarkan riset dinyatakan bahwa bahwa di platform Web, dan baik untuk kategori open source maupun umum, MySQL adalah database yang paling banyak dipakai. Menurut perusahaan pengembangnya, MySQL telah terpasang di sekitar 3 juta komputer. Puluhan hingga ratusan ribu situs mengandalkan MySQL bekerja siang malam memompa data bagi para pengunjungnya.

Format Perintah

Berikut adalah ketentuan-ketentuan memberi perintah pada MySQL:

- Setiap perintah harus diakhiri dengan tanda titik koma , kecuali untuk perintahtertentu, misal : quit
- Setiap perintah akan disimpan dalam buffer (memori sementara) untuk menyimpan histori perintah-perintah yang pernah diberikan.
- Perintah dapat berupa perintah SQL atau perintah khusus MySQL.

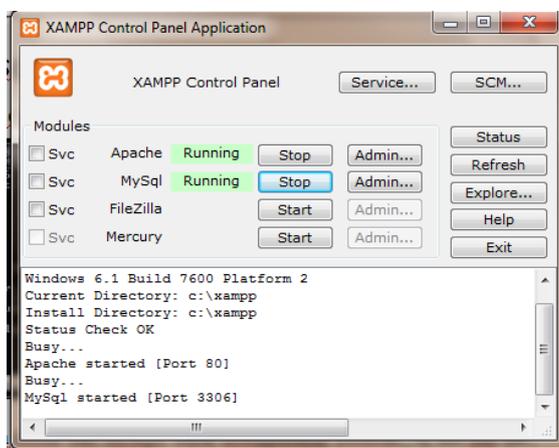
- Perintah-perintah dalam lingkungan MySQL tidak menerapkan aturan case sensitive, tetapi case insensitive yaitu perintah bisa dituliskan dalam huruf besar atau pun huruf kecil.
- Aturan case sensitive diterapkan pada penamaan objek-objek dalam database seperti nama database atau nama table, namun aturan ini hanya ada dalam lingkungan Unix dan Linux.

Ada beberapa tanda yang sering muncul di prompt :

Prompt	Arti
mysql>	Siap menerima perintah baru
->	Menunggu baris berikut untuk perintah yang lebih dari satu baris
'>	Menunggu baris berikut, menunggu penutup string yang dimulai dengan tanda kutip satu (""")
">	Menunggu baris berikut, menunggu penutup string yang dimulai dengan tanda kutip dua (""")
`>	Menunggu baris berikutnya, menunggu penutup identifier yang dimulai dengan tanda backtick (""")

Start dan Stop MySQL

Berikut cara memulai MySQL dengan menggunakan bantuan XAMPP. Aktifkan Xampp Control Panel Application, klik start apache dan mysql.



Aktifkan command prompt, lalu ketik seperti gambar berikut:

```

Command Prompt - mysql -u root
Microsoft Windows [Version 6.1.7600]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Users\Zyrex>cd \
C:\>cd xampp
C:\xampp>cd mysql
C:\xampp\mysql>cd bin
C:\xampp\mysql\bin>mysql -u root
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 1
Server version: 5.0.45-community-nt MySQL Community Edition (GPL)

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

mysql>

```

Sedangkan untuk stop atau keluar dari MySQL dapat menggunakan perintah : \q, exit dan quit.

```

Command Prompt
Microsoft Windows [Version 6.1.7600]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Users\Zyrex>cd \
C:\>cd xampp
C:\xampp>cd mysql
C:\xampp\mysql>cd bin
C:\xampp\mysql\bin>mysql -u root
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 1
Server version: 5.0.45-community-nt MySQL Community Edition (GPL)

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

mysql> \q
Bye
C:\xampp\mysql\bin>_

```

DDL (Data Definition Language), DDL merupakan kelompok perintah yang berfungsi untuk mendefinisikan atribut-atribut basis data, tabel, atribut(kolom), batasan-batasan terhadap suatu atribut, serta hubungan antar tabel. Yang termasuk dalam kelompok DDL ini adalah CREATE, ALTER, dan DROP.

- a. Syntax Membuat Database : CREATE DATABASE namadatabase;

Namadatabase tidak boleh mengandung spasi dan tidak boleh memiliki nama yang sama antar database. Berikut ini perintah untuk membuat database dengan nama rental : CREATE DATABASE CV_SEJAHTERA;

```
mysql> create database CV_SEJAHTERA;
Query OK, 1 row affected (0.01 sec)
```

Syntax tambahan untuk menampilkan daftar nama database yang ada pada mysql menggunakan perintah : SHOW DATABASES;

```
mysql> show databases;
+-----+
| Database |
+-----+
| information_schema |
| akademik |
| akademik_t |
| belajar_ajah |
| belajarif |
| cdcol |
| contohlooping |
| cv_sejahtera |
| data_mahasiswa |
| dataku |
| dataku2 |
| db_akademik |
+-----+
```

- b. Memilih Database : USE namadatabase;

Sebelum membuat suatu tabel, terlebih dahulu harus memilih salah satu database sebagai database aktif yang akan digunakan untuk menyimpan tabel-tabel, Berikut ini perintah untuk menggunakan database dengan nama CV_SEJAHTERA:
USE SEJAHTERA;

```
mysql> use CV_SEJAHTERA;
Database changed
```

- c. Syntax Menghapus Database : DROP DATABASE namadatabase;

Database yang akan dihapus sesuai dengan namadatabase. Berikut ini perintah

untuk menghapus database dengan nama rental : DROP DATABASE RENTAL;

- d. Membuat Tabel : CREATE TABLE namatabel2 (Field1 TipeData1,Field2 TipeData2);

Nama tabel tidak boleh mengandung spasi (space). Field1 dan TipeData1 merupakan nama kolom pertama dan tipe data untuk kolom pertama. Jika ingin membuat tabel dengan kolom lebih dari satu, maka setelah pendefinisian tipe data sebelumnya diberikan tanda koma (,).

Berikut ini perintah untuk membuat tabel dengan nama barang :

```
mysql> create table barang(KDBARANG char(3) primary key,
-> NAMA_BARANG varchar(20),
-> SATUAN varchar(10),
-> HARGA int(11));
Query OK, 0 rows affected (2.70 sec)
```

- e. Menampilkan Tabel

Untuk menampilkan daftar nama tabel yang ada pada database yang sedang aktif/digunakan (dalam hal ini database rental) : SHOW TABLES;

```
mysql> show tables;
+-----+
| Tables_in_cv_sejahtera |
+-----+
| barang                  |
+-----+
1 row in set (0.00 sec)
```

- f. Menampilkan Atribut Tabel : DESC namatabel;

Untuk menampilkan deskripsi tabel (dalam hal ini jenisfilm) syntaxnya adalah : DESC barang;

```
mysql> desc barang;
+-----+-----+-----+-----+-----+-----+
| Field      | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| KDBARANG   | char(3)       | NO   | PRI |          |       |
| NAMA_BARANG | varchar(20)   | YES  |     | NULL    |       |
| SATUAN     | varchar(10)   | YES  |     | NULL    |       |
| HARGA      | int(11)       | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
4 rows in set (0.05 sec)
```

- g. Syntax Menghapus Tabel : DROP TABLE namatabel;

Tabel yang akan dihapus sesuai dengan namatabel, berikut ini perintah untuk menghapus tabel dengan nama jenisfilm : DROP TABLE BARANG;

- h. Mendefinisikan Null/Not Null : CREATE TABLE namatabel (Field1 TipeData1 NOT NULL, Field2 TipeData2);
- i. Mendefinisikan Primary Key Pada Tabel

Terdapat tiga cara untuk mendefinisikan primary key. Berikut ini adalah Syntax mendefinisikan primary key untuk Field1
CREATE TABLE namatabel(Field1 TipeData1 NOT NULL PRIMARY KEY, Field2 TipeData2);

Atau

CREATE TABLE namatabel (Field1 TipeData1, Field2 TipeData2, PRIMARY KEY(Field1));

Atau

ALTER TABLE namatabel ADD CONSTRAINT namaconstraint PRIMARY KEY (namakolom);

- j. Menghapus Primary Key Pada Tabel

Cara 1 : Jika primary key dibuat dengan menggunakan alter table :

ALTER TABLE namatabel DROP CONSTRAINT namaconstraint;

Cara 2 : Jika primary key dibuat melalui create table :

ALTER TABLE namatabel DROP PRIMARY KEY;

- k. Menambah Kolom Baru Pada Tabel : ALTER TABLE namatabel ADD fieldbaru tipe;

Namatabel adalah nama tabel yang akan ditambah fieldnya. Fieldbaru adalah nama kolom yang akan ditambahkan, tipe adalah tipe data dari kolom yang akan ditambahkan.

Berikut ini contoh perintah untuk menambah kolom keterangan dengan tipe data varchar(25):

```
ALTER TABLE JENISFILM ADD KETERANGAN VARCHAR(25);
```

Untuk meletakkan field diawal, tambahkan sintaks first :

```
ALTER TABLE PELANGAN ADD COLUMN KODE CHAR(5) FIRST;
```

Untuk menyisipkan field setelah field tertentu, tambahkan sintaks after :

```
ALTER TABLE PELANGAN ADD COLUMN PHONE CHAR(5) AFTER ALAMAT;
```

- I. Mengubah Tipe Data atau Lebar Kolom Pada Tabel : ALTER TABLE NAMATABEL MODIFY COLUMN FIELD TIPE

Namatabel adalah nama tabel yang akan diubah tipe data atau lebar kolomnya. Field adalah kolom yang akan diubah tipe data atau lebarnya. Tipe adalah tipe data baru atau tipe data lama dengan lebar kolom yang berbeda. Berikut ini contoh perintah untuk mengubah tipe data untuk kolom keterangan dengan char(20) :

```
ALTER TABLE JENISFILM MODIFY COLUMN KETERANGAN VARCHAR(20);
```

- m. Mengubah Nama Kolom : ALTER TABLE namatabel CHANGE COLUMN namalamakolom namabarukolom tipedatabaru;

Namatabel adalah nama tabel yang akan diubah nama kolomnya, namalamakolom adalah kolom yang akan diganti namanya, namabarukolom adalah nama baru kolom, tipedatanya adalah tipe data dari kolom tersebut. Berikut ini contoh perintah untuk mengubah nama kolom keterangan menjadi ket :

```
ALTER TABLE JENISFILM CHANGE COLUMN KETERANGAN KET VARCHAR(20);
```

- n. Menghapus Kolom Pada Tabel : ALTER TABLE namatabel DROP COLUMN namakolom;

3. LATIHAN

1. Buat sebuah database dengan nama coba !
2. Buat sebuah tabel dengan nama mahasiswa di dalam database coba !
3. Tambahkan sebuah kolom : keterangan (varchar 15), sebagai kolom terakhir !

4. Tambahkan kolom nim (int 11) di awal (sebagai kolom pertama) !
5. Sisipkan sebuah kolom dengan nama phone (varchar 15) setelah kolom alamat varchar(15) !
6. Ubah kolom nim menjadi char(11) !
7. Ubah nama kolom phone menjadi telepon (varchar 20) !
8. Hapus kolom keterangan dari tabel !
9. Ganti nama tabel menjadi student!
10. Jadikan nim sebagai primary key !

MODUL II

DML DAN RETRIEVE DATA (BAGIAN I)

1. TUJUAN

Setelah menyelesaikan modul ini, Anda diharapkan dapat :

- Mengenal data manipulation language dan mampu menggunakannya
- Mampu mengelola record dan retrieve data

2. TEORI DASAR

DML (Data Manipulation Language) DML adalah kelompok perintah yang berfungsi untuk memanipulasi data dalam basis data, misalnya untuk pengambilan, penyisipan, pengubahan dan penghapusan data. Perintah yang termasuk dalam kategori DML adalah : INSERT, DELETE, UPDATE dan SELECT.

1. INSERT

Perintah INSERT digunakan untuk menambahkan baris pada suatu tabel. Terdapat dua cara untuk menambah baris, yaitu:

Cara 1: Menambah baris dengan mengisi data pada setiap kolom :

INSERT INTO namatabel VALUES (nilai1,nilai2,nilai-n);

```
mysql> insert into barang values('B01','BUKU TULIS','LUSIN',50000),
-> ('B02','PULPEN','LUSIN',80000),
-> ('B03','PENGHAPUS','LUSIN',20000);
Query OK, 3 rows affected (2.61 sec)
Records: 3 Duplicates: 0 Warnings: 0
```

Cara 2 : Menambah baris dengan hanya mengisi data pada kolom tertentu :

INSERT INTO namatabel (kolom1,kolom2,kolom-n) VALUES (nilai1,nilai2,nilai-n);

```
mysql> insert into barang (KDBARANG,NAMA_BARANG,SATUAN,HARGA)
-> values('B04','PERAUT','LUSIN',45000);
Query OK, 1 row affected (0.00 sec)
```

Keterangan :

Jika data bertipe string, date atau time (contoh : action, horror, 2007-11-10) maka pemberian nilainya diapit dengan tanda petik tunggal ('horror') atau petik ganda ("horror"). Jika data bertipe numerik (2500, 400) maka pemberian nilainya tidak diapit tanda petik tunggal maupun ganda.

2. DELETE

Perintah DELETE digunakan untuk menghapus satu baris, baris dengan kondisi tertentu atau seluruh baris. Syntax : DELETE FROM namatabel [WHERE kondisi];

Perintah dalam tanda [] bersifat opsional untuk menghapus suatu baris dengan suatu kondisi tertentu.

3. UPDATE

Perintah UPDATE digunakan untuk mengubah isi data pada satu atau beberapa kolom pada suatu table. Syntax :

UPDATE namatabel SET kolom1 = nilai1, kolom2 = nilai2 [WHERE kondisi];

```
mysql> update barang set NAMA_BARANG='PERAUT'
-> where KDBARANG='B03';
Query OK, 1 row affected (2.65 sec)
Rows matched: 1  Changed: 1  Warnings: 0
```

Perintah dalam tanda [] bersifat opsional untuk mengubah suatu baris dengan suatu kondisi tertentu.

4. SELECT

Perintah SELECT digunakan untuk menampilkan isi dari suatu tabel yang dapat dihubungkan dengan tabel yang lainnya.

- a. Menampilkan data untuk semua kolom menggunakan asterisk (*) :

SELECT * FROM namatabel;

```
mysql> select*from barang;
+-----+-----+-----+-----+
| KDBARANG | NAMA_BARANG | SATUAN | HARGA |
+-----+-----+-----+-----+
| B01      | BUKU TULIS  | LUSIN  | 50000 |
| B02      | PULPEN      | LUSIN  | 80000 |
| B03      | PERAUT      | LUSIN  | 20000 |
| B04      | PERAUT      | LUSIN  | 45000 |
+-----+-----+-----+-----+
4 rows in set (0.00 sec)
```

- b. Menampilkan data untuk kolom tertentu :

SELECT kolom1,kolom2,kolom-n FROM namatabel;

```
mysql> select KDBARANG,NAMA_BARANG from barang;
+-----+-----+
| KDBARANG | NAMA_BARANG |
+-----+-----+
| B01      | BUKU TULIS  |
| B02      | PULPEN      |
| B03      | PERAUT      |
| B04      | PERAUT      |
+-----+-----+
4 rows in set (0.00 sec)
```

- c. Menampilkan data dengan kondisi data tertentu dengan klausa WHERE:

SELECT * FROM namatabel WHERE kondisi;

```
mysql> select *from barang where KDBARANG='B02';
+-----+-----+-----+-----+
| KDBARANG | NAMA_BARANG | SATUAN | HARGA |
+-----+-----+-----+-----+
| B02      | PULPEN      | LUSIN  | 80000 |
+-----+-----+-----+-----+
1 row in set (0.00 sec)
```

Beberapa operator perbandingan yang dapat digunakan pada klausa WHERE adalah "="(sama dengan) , > (lebih dari), < (kurang dari), < > (tidak sama dengan), >= (lebih dari atau sama dengan), <= (kurang dari atau sama dengan). Adapun operator lain, yaitu : AND, OR, NOT, BETWEEN-AND, IN dan LIKE.

3. LATIHAN

1. Buatlah sebuah database dengan nama coba2!
2. Buatlah sebuah tabel dengan nama pet pada database coba2!

Field	Type	Null	Key	Default	Extra
name	varchar(20)	NO			
owner	varchar(20)	YES		NULL	
species	varchar(20)	YES		NULL	
sex	char(1)	YES		NULL	
birth	date	YES		NULL	
death	date	YES		NULL	

3. Isi data pada tabel pet, sbb:

name	owner	species	sex	birth	death
Puffball	Diane	hamster	f	1977-03-03	NULL
Claws	Gwen	cat	m	1994-03-17	NULL
Fluffy	Harold	cat	f	1973-02-04	NULL
Buffy	Harold	dog	f	1989-05-13	NULL
Fang	Benny	dog	m	1970-08-27	NULL
Bowser	Diane	dog	m	1989-08-31	1995-07-29
Chirpy	Gwen	bird	f	1978-09-11	NULL
Whistler	Gwen	bird	NULL	1997-12-09	NULL
Slim	Benny	snake	m	1996-04-29	NULL

4. Tampilkan semua isi/record tabel pet!
5. Ubah data tanggal lahir hewan yang bernama Bowser menjadi: 1979-08-31 !
6. Tampilkan satu baris / record data yang telah diubah tadi yaitu record dengan nama Bowser saja!
7. Hapus hewan yang bernama Chirpy!
8. Tampilkan record atau data yang tanggal kelahirannya lebih dari atau sama dengan

1998-1-1 !

9. Tampilkan semua hewan dengan spesis anjing dan anjing tersebut berjenis kelamin betina!
10. Dengan satu perintah select, tampilkan semua hewan dengan spesis ular atau spesis burung (dalam satu tabel ada hewan spesies ular dan burung saja)!
11. Dengan satu perintah select, tampilkan semua hewan yang berspesis kucing dengan kelamin laki-laki atau hewan yang berspesis anjing dengan kelamin betina (tampilkan dalam satu tabel)!

MODUL III

DML DAN RETRIEVE DATA (BAGIAN 2)

1. TUJUAN

Setelah menyelesaikan modul ini, Anda diharapkan dapat :

- Mencari dan menampilkan data dengan perintah select
- Mengkombinasikan perintah select dengan perintah lainnya

2. TEORI DASAR

- a. Memberikan nama lain pada kolom :SELECT namakolomlama AS namakolombaru FROM namatabel;

Berikut ini perintah untuk memberikan nama lain pada kolom jenis menjadi jenis_film pada tabel jenisfilm:

```
SELECT JENIS AS TYPE FROM JENISFILM;
```

- b. Menggunakan alias untuk nama tabel: SELECT namalias .jenis, namalias .harga FROM namatabel namalias;

Berikut ini perintah untuk memberikan alias pada tabel jenisfilm :

```
SELECT J.JENIS, J.HARGA FROM JENISFILM J;
```

- c. Menampilkan data lebih dari dua tabel: SELECT * FROM namatabel1, namatabel2, namatabel-n;
- d. Nested Queries / Subquery (IN, NOT IN, EXISTS, NOT EXISTS)

Subquery berarti query di dalam query. Dengan menggunakan subquery, hasil dari query akan menjadi bagian dari query di atasnya. Subquery terletak di dalam klausa WHERE atau HAVING. Pada klausa WHERE, subquery digunakan untuk memilih baris-

baris tertentu yang kemudian digunakan oleh query. Sedangkan pada klausa HAVING, subquery digunakan untuk memilih kelompok baris yang kemudian digunakan oleh query.

Contoh 1: perintah untuk menampilkan data pada tabel jenisfilm yang mana data pada kolom jenis-nya tercantum pada tabel film menggunakan IN :

```
SELECT * FROM JENISFILM WHERE JENIS IN (SELECT JENIS FROM FILM);
```

atau menggunakan EXISTS

```
SELECT * FROM JENISFILM WHERE EXISTS (SELECT * FROM FILM WHERE HARGA > 2000);
```

Pada contoh di atas:

SELECT JENIS FROM FILM disebut subquery, sedangkan :

SELECT * FROM JENISFILM berkedudukan sebagai query. Perhatikan, terdapat data jenis dan harga pada tabel jenisfilm yang tidak ditampilkan. Hal ini disebabkan data pada

kolom jenis tidak terdapat pada kolom jenis di tabel film.

Contoh 2: perintah untuk menampilkan data pada tabel jenisfilm yang mana data pada kolom jenis-nya tidak tercantum pada tabel film menggunakan NOT IN:

```
SELECT * FROM JENISFILM WHERE JENIS NOT IN (SELECT JENIS FROM FILM);
```

atau menggunakan NOT EXISTS

```
SELECT * FROM JENISFILM WHERE NOT EXISTS (SELECT * FROM FILM WHERE HARGA > 2000);
```

e. Operator comparison ANY dan ALL

Operator ANY digunakan berkaitan dengan subquery. Operator ini menghasilkan TRUE (benar) jika paling tidak salah satu perbandingan dengan hasil subquery menghasilkan nilai TRUE. Ilustrasinya jika:

Gaji > ANY (S)

Jika subquery S menghasilkan G1, G2, ..., Gn, maka kondisi di atas identik dengan:

(gaji > G1) OR (gaji > G2) OR ... OR (gaji > Gn)

Contoh: perintah untuk menampilkan semua data jenisfilm yang harganya bukan yang terkecil:

```
SELECT * FROM JENISFILM WHERE HARGA > ANY (SELECT HARGA FROM
JENISFILM);
```

Operator ALL digunakan untuk melakukan perbandingan dengan subquery. Kondisi dengan ALL menghasilkan nilai TRUE (benar) jika subquery tidak menghasilkan apapun atau jika perbandingan menghasilkan TRUE untuk setiap nilai query terhadap hasil subquery.

Contoh : perintah untuk menampilkan data jenisfilm yang harganya paling tinggi:

```
SELECT * FROM JENISFILM WHERE HARGA >= ALL (SELECT HARGA FROM
JENISFILM);
```

f. Sintak ORDER BY

Klausa ORDER BY digunakan untuk mengurutkan data berdasarkan kolom tertentu sesuai dengan tipe data yang dimiliki. Contoh : perintah untuk mengurutkan data film berdasarkan kolom judul:

```
SELECT * FROM FILM ORDER BY JUDUL;
```

atau tambahkan ASC untuk pengurutan secara ascending (menaik) :

```
SELECT * FROM FILM ORDER BY JUDUL ASC;
```

atau tambahkan DESC untuk pengurutan secara descending (menurun):

```
SELECT * FROM FILM ORDER BY JUDUL DESC;
```

g. Sintak DISTINCT

Distinct adalah kata kunci ini untuk menghilangkan duplikasi. Sebagai Contoh, buat sebuah tabel pelanggan yang berisi nama dan kota asal dengan beberapa record isi dan beberapa kota asal yang sama. Kemudian ketikkan perintah berikut:

```
SELECT DISTINCT KOTA FROM PELANGGAN;
```

Dengan perintah di atas maka nama kota yang sama hanya akan ditampilkan satu saja.

h. UNION, INTERSECT dan EXCEPT

UNION merupakan operator yang digunakan untuk menggabungkan hasil query, dengan ketentuan jumlah, nama dan tipe kolom dari masing-masing tabel yang akan ditampilkan datanya harus sama. Berikut ini perintah untuk memperoleh data pada tabel

Modul Praktikum Pemrograman Database – STMIK Yadika BANGIL

film dimana jenisnya action dan horor:

```
SELECT JENIS, JUDUL FROM FILM WHERE JENIS = 'ACTION' UNION SELECT
JENIS, JUDUL FROM FILM WHERE JENIS = 'HOROR';
```

Perintah di atas identik dengan:

```
SELECT JENIS, JUDUL FROM FILM WHERE JENIS = 'ACTION' OR JENIS =
'HOROR';
```

Namun tidak semua penggabungan dapat dilakukan dengan OR, yaitu jika bekerja pada dua tabel atau lebih.

INTERSECT merupakan operator yang digunakan untuk memperoleh data dari dua buah query dimana data yang ditampilkan adalah yang memenuhi kedua query tersebut dengan ketentuan jumlah, nama dan tipe kolom dari masing-masing tabel yang akan ditampilkan datanya harus sama.

```
SELECT * FROM namatabel1 INTERSECT SELECT * FROM namatabel2;
```

Pada MySQL tidak terdapat operator INTERSECT namun sebagai gantinya dapat menggunakan operator IN seperti contoh 1 pada bagian Nested Queries.

EXCEPT / Set Difference merupakan operator yang digunakan untuk memperoleh data dari dua buah query dimana data yang ditampilkan adalah data yang ada pada hasil query 1 dan tidak terdapat pada data dari hasil query 2 dengan ketentuan jumlah, nama dan tipe kolom dari masing-masing tabel yang akan ditampilkan datanya harus sama.

```
SELECT * FROM namatabel1 EXCEPT SELECT * FROM namatabel2;
```

Pada MySQL tidak terdapat operator EXCEPT namun sebagai gantinya dapat menggunakan operator NOT IN seperti contoh 2 pada bagian Nested Queries.

3. LATIHAN

1. Buat tabel pegawai sebagai berikut:

Field	Type	Null	Key	Default	Extra
idpegawai	char(6)	NO			
namadepan	varchar(20)	YES		NULL	
namabelakang	varchar(25)	NO			
email	varchar(25)	NO			
telepon	varchar(20)	YES		NULL	
tglkontrak	date	NO			
idjoh	varchar(10)	NO			
gaji	int(8)	YES		NULL	
tunjangan	int(8)	YES		NULL	
idmanajer	char(6)	YES		NULL	
iddepartemen	char(4)	YES		NULL	

2. Isi data tabel (data dapat dilihat pada halaman terakhir)!
3. Tampilkan semua kolom di tabel!
4. Tampilkan kolom idpegawai, namabelakang dan gaji saja!
5. Tampilkan kolom idpegawai, namabelakang, gaji, tunjangan dan sebuah kolom baru yaitu tunjangan+gaji yang berisi jumlah tunjangan dan gaji !
6. Ubah tunjangan menjadi NULL untuk pegawai dengan idpegawai = E003. Kemudian lakukan kembali percobaan 5.
7. Seperti percobaan 5, tampilkan kolom idpegawai, namabelakang, gaji, tunjangan dan sebuah kolom baru (gunakan alias) yaitu total_pendapatan yang berisi jumlah tunjangan dan gaji!
8. Tambahkan record baru dengan value: E006,lincoln, burrows, linc@yahoo.com, 085275384544, 2008-09-01, L0006, 1750000, NULL, ex, coml.
9. Untuk pegawai yang ber-id E004 dan E005 ubah idmanajernya menjadi a!
10. Sekarang tampilkan kolom idmanajer saja!
11. Dari percobaan 10, terdapat 3 idmanajer yang sama dengan total record 6, sekarang tampilkan
12. idmanajer tanpa duplikasi idmanajer sehingga akan tampil 4 record dengan idmanajer yang berbeda!
13. Tampilkan pegawai yang gajinya antara 1750000 - 1250000!
14. Tampilkan tabel pegawai yang terurut berdasarkan namabelakang (dari a ke z)!
15. Tampilkan tabel pegawai yang diurutkan berdasarkan nama depan dengan urutan terbalik (dari z ke a)!

MODUL IV RETRIEVE, GROUP, FILTER DAN PATTERN MATCHING

1. TUJUAN

Setelah melakukan percobaan ini, Anda diharapkan dapat :

- Mampu meretrieve data dan mengelompokkannya
- Mampu menampilkan data dengan pencocokan pola / karakter

2. DASAR TEORI

❖ RETRIEVE SQL dengan GROUP BY dan HAVING

Klausa GROUP BY digunakan untuk melakukan pengelompokkan data. Syntax :

```
SELECT field1,SUM(field2) FROM namatable GROUP BY field1;
```

Sebagai contoh, terdapat table barang dengan data sebagai berikut :

kode_barang	nama_barang	satuan_barang	stok_barang	harga_barang
B1	HARDISK	BUAH	12	500000
B2	MP3 PLAYER	UNIT	30	200000
B3	TAPE	UNIT	25	350000
B4	FLASHDISK	BUAH	12	100000

Klausa HAVING digunakan untuk menentukan kondisi bagi klausa GROUP BY. Kelompok yang memenuhi HAVING saja yang akan dihasilkan. Syntax :

```
SELECT field1 FROM namatable GROUP BY field1 HAVING COUNT(field2);
```

❖ PATTERN MATCHING (Pencocokan Pola / Karakter)

Fungsi string digunakan untuk menampilkan data yang di dasarkan pada pencarian dengan karakter. Pada pencarian data digunakan syntax LIKE, pada dasarnya syntax LIKE hampir sama dengan syntax = .

Bedanya kalau syntax = , maka pencarian karakter harus sesuai dengan kata yang kita buat tetapi dengan menggunakan LIKE karakter yang akan kita tampilkan tidak harus lengkap hanya dengan menuliskan salah satu huruf atau kata saja, maka semua data yang akan kita cari akan ditampilkan.

SQL mempunyai dua symbol khusus yang dipakai untuk pencocokan pola :

1. % : digunakan untuk mencocokkan karakter sebelum atau sesudah tana %;
2. _ : digunakan untuk mencari karakter sebanyak jumlah tanda _.

Contoh:

LIKE '%GLASGOW%' artinya digunakan untuk mencari data pada kolom tertentu yang mengandung karakter 'GLASGOW'.

Bentuk umumnya :

```
SELECT * FROM nama_tabel WHERE nama_kolom LIKE 'char%';
SELECT * FROM nama_tabel WHERE nama_kolom LIKE '%char';
SELECT * FROM nama_tabel WHERE nama_kolom LIKE '%char%';
SELECT * FROM nama_tabel WHERE nama_kolom NOT LIKE '%char%';
SELECT * FROM nama_tabel WHERE nama_kolom LIKE '_';
```

3. LATIHAN

Tampilkan record / data hanya kolom satuan barang dan digabungkan dengan jumlah stok barang yang dikelompokkan berdasarkan kolom satuan barang pada table barang diatas!

```
mysql> select satuan_barang,sum(stok_barang) from barang group by satuan_barang;
+-----+-----+
| satuan_barang | sum(stok_barang) |
+-----+-----+
| BUAH         | 24                |
| UNIT         | 55                |
+-----+-----+
2 rows in set (0.08 sec)
```

Evaluasi dan Pertanyaan !

1. Jika syntax ini : SELECT kode_barang, satuan_barang, SUM(stok_barang) from barang GROUP BY satuan_barang.
2. Jika syntax ini diketikkan pada cmd : SELECT nama_barang, satuan_barang, SUM(stok_barang) GROUP BY satuan_barang.

MODUL V

QUERY, VIEW & JOIN

1. TUJUAN

Setelah melakukan percobaan ini, Anda diharapkan dapat :

- Mampu menampilkan data dengan view
- Mampu menampilkan gabungan data dari tabel yang berbeda.
- Mampu mengenal beberapa join
- Mampu melakukan operasi join beberapa tabel

2. TEORI DASAR

❖ Query

Query adalah pernyataan yang meminta pengguna mengambil informasi. Bagian DML yang terlibat dalam pengambilan informasi disebut bahasa query. Istilah bahasa query sering disamakan dengan istilah bahasa manipulasi data. Sedangkan SQL adalah sebuah sintaks untuk mengeksekusi query.

❖ View

Merupakan salah satu objek database, yang secara logika merepresentasikan sub himpunan dari data yang berasal dari satu atau lebih tabel. Kegunaan view adalah untuk membatasi akses database, membuat query kompleks secara mudah, mengijinkan independensi data dan untuk menampilkan view (pandangan) data yang berbeda dari data yang sama.

Tujuan dari view adalah:

- a. Menurunkan Network Traffic (beban Network).
- b. Menyimpan suatu perintah SQL (terutama yang kompleks) dimana perintah tersebut sering digunakan dan diakses.
- c. Mencegah user untuk dapat mengakses suatu tabel sepenuhnya.
- d. Misal user dapat mengakses nama dan nomor telepon tetapi tidak bisa mengakses tanggal lahir dan gaji.

Sintak dari View adalah :

```
CREATE [OR REPLACE] [<algorithm attributes>]
```

```
VIEW [database.]< name> [(<columns>)]
AS <SELECT statement> [<check options>];
```

Contoh:

a) Membuat View:

```
CREATE VIEW pelanggan_simpati AS
SELECT nama,alamat,tgl_lahir,telepon
FROM pelanggan WHERE telepon REGEXP '^081[23]';
ORDER BY nama;
```

b) Cara mengaksesnya:

```
SELECT * FROM pelanggan_simpati;
SELECT nama,alamat FROM pelanggan_simpati;
```

View termasuk dalam komponen database. Secara default, suatu view baru dibuat ke dalam database yang diaktifkan. Untuk membuat secara eksplisit di dalam suatu database tertentu, maka buatlah nama view dengan format: db_name.view_name.

Contoh lain yang akan diberikan adalah view untuk menyimpan informasi detail mahasiswa, dalam hal ini melibatkan 2 tabel, yaitu mahasiswa dan prodi.

Contoh:

```
mysql> create view vDetailMhs as
-> select m.nim, m.nama, m.alamat, p.nama_prodi, p.jurusan
-> from mahasiswa m, prodi p
-> where (m.kode_prodi=p.kode_prodi);
```

```
mysql> select * from vDetailMhs;
```

nim	nama	alamat	nama_prodi	jurusan
00543	Moh. Riyan	Karangmalang A-50	Eks Ilmu Komputer	Matematika
10041	Sugiharti	Karangmalang A-23	Ilmu Komputer	Matematika
10043	Ahmad Sholihun	Karangmalang D-17	Ilmu Komputer	Matematika

Dari contoh diatas dapat dijelaskan bahwa view tersebut berisi informasi mahasiswa (nim, nama, alamat) dan informasi prodi mahasiswa yang bersangkutan (nama_prodi dan jurusan). Implementasi view dalam program aplikasi adalah untuk memudahkan dalam mendesain laporan (report).

❖ **Join**

Operasi Join, Join merupakan operasi yang digunakan untuk menggabungkan dua tabel atau lebih dengan hasil berupa gabungan dari kolom-kolom yang berasal dari tabel-tabel tersebut. Pada join sederhana, tabel-tabel digabungkan dan didasarkan pada pencocokan antara kolom pada tabel yang berbeda.

```
mysql> select judul, harga From film, jenisfilm
-> where jenisfilm.jenis=film.jenis;
```

judul	harga
Spiderman 1	2500
Spiderman 2	2500
Spiderman 3	2500
Love Story	1000
Suster Megasot	4000
Evil Death	4000

```
6 rows in set (8.39 sec)
```

Pada contoh di atas, jenisfilm.jenis=film.jenis merupakan kondisi untuk mencocokkan data antara kolom jenis milik tabel jenisfilm dan film.

a. Inner Join

Inner join digunakan untuk menampilkan data dari dua tabel yang berisi data sesuai dengan syarat dibelakang on (tidak boleh null), dengan kata lain semua data dari tabel kiri mendapat pasangan data dari tabel sebelah kanan. Berikut ini perintah untuk menampilkan data dari tabel jenisfilm dan film dengan syarat berdasarkan kolom jenis:

```
SELECT * FROM JENISFILM INNER JOIN FILM ON (JENISFILM.JENIS = FILM.JENIS);
```

b. Left Join

Left join digunakan untuk menampilkan semua data dari tabel sebelah kiri perintah left join beserta pasangannya dari tabel sebelah kanan. Meskipun terdapat data dari sebelah kiri tidak memiliki pasangan, tetap akan ditampilkan dengan pasangannya berupa nilai NULL.

```
SELECT * FROM JENISFILM LEFT JOIN FILM ON (JENISFILM.JENIS = FILM.JENIS);
```

c. Right Join

Right join digunakan untuk menampilkan semua data dari tabel sebelah kanan perintah right join beserta pasangannya dari tabel sebelah kiri. Meskipun terdapat data dari sebelah kanan tidak memiliki pasangan, tetap akan ditampilkan dengan pasangannya berupa nilai NULL.

```
SELECT * FROM JENISFILM RIGHT JOIN FILM ON (JENISFILM.JENIS = FILM.JENIS);
```

c. Natural Join

Natural join digunakan untuk menampilkan semua data dari dua tabel dimana jika terdapat kolom yang sama, maka yang akan ditampilkan hanya salah satunya saja, yaitu kolom dari tabel sebelah kiri perintah natural join.

```
SELECT * FROM JENISFILM NATURAL JOIN FILM;
```

Terdapat Penggabungan Natural Join dengan Left dan Right Join:

1. Natural Left Join

Natural left join digunakan untuk menampilkan semua data dari tabel sebelah kiri perintah natural left join beserta pasangannya dari tabel sebelah kanan. Meskipun terdapat data dari sebelah kiri tidak memiliki pasangan, tetap akan ditampilkan dengan pasangannya berupa nilai NULL.

```
SELECT * FROM JENISFILM NATURAL LEFT JOIN FILM;
```

2. Natural Right Join

Natural right join digunakan untuk menampilkan semua data dari tabel sebelah kanan perintah natural right join beserta pasangannya dari tabel sebelah kiri. Meskipun terdapat data dari sebelah kanan tidak memiliki pasangan, tetap akan ditampilkan dengan pasangannya berupa nilai NULL.

```
SELECT * FROM JENISFILM NATURAL RIGHT JOIN FILM;
```

3. LATIHAN

1. Buat sebuah database dengan nama sewa mobil :
2. Buat tabel mobil dan isi datanya :

Field	Type	Null	Key	Default	Extra
kode	varchar(5)	NO	PK		
jenis	varchar(10)	NO			
merk	varchar(15)	NO			
tarif	int(11)	NO			
nopol	varchar(8)	NO			

ika BANGIL

kode	jenis	merk	tarif	nopol
M001	SEDAN	BMW E5	500000	BG1234AA
M002	SEDAN	HONDA CRU	350000	BG2345BB
M003	BUS	MERCEDEZ	1000000	BG3456CC
M004	BUS	DYNA	800000	BG8443DD
M005	TRUCK	HYUNDAI ZX	1500000	BG4638EE
M006	TRUCK	DYNA X1	1500000	BG8473FF

Field	Type	Null	Key	Default	Extra
kode	varchar(6)	NO	PRI		
nana	varchar(15)	NO			
kontak	varchar(15)	NO			
alamat	varchar(30)	NO			
kota	varchar(15)	NO			
kodepos	varchar(5)	NO			
telpon	varchar(15)	NO			
fax	varchar(15)	NO			

kode	nana	kontak	alamat	kota	kodepos	telpon	fax
P001	PT FOX RIVER	HENDRA	JL. JEND. SUDIRMAN 657	BENGRULU	30245	1234567	1234568
P002	CV FOXCOM	IWAN	JL. WAHID HASYIM 743	JAKARTA	73429	234567	234568
P003	PT FARMACOM	YANI	JL. AHMAD DAHLAN 45	LAMPUNG	28349	3334445	3334446

4. Buat tabel sewa dan isi datanya:

Field	Type	Null	Key	Default	Extra
nofaktursewa	varchar(5)	NO			
kodepelanggan	varchar(6)	NO			
tglsewa	date	NO			
kodenobil	varchar(5)	NO			
lamasewa	int(11)	NO			
uangmuka	int(11)	NO			

nofaktursewa	kodepelanggan	tglsewa	kodenobil	lamasewa	uangmuka
F001	P001	2008-12-01	M001	2	200000
F001	P001	2008-12-01	M003	2	200000
F002	P002	2008-12-02	M002	1	100000

MODUL VI

MENAMPILKAN & PENCARIAN DATA MYSQL DENGAN PHP

1. TUJUAN

- Agar mahasiswa mampu menampilkan data pada Tabel yang memiliki Relasional
- Agar mahasiswa mengetahui Database Relasional dengan MySQL
- Agar mahasiswa mampu untuk membangaun sebuah Sistem yang melibatkan Perimintaan Relasional dengan SQL.
- Agar mahasiswa mampu menggunakan query SQL yang berkaitan dengan Perbandingan data
Dengan menggunakan perbandingan data diharapkan mahasiswa dapat menerapkanya pada program yang melibatkan pencarian data

2. TEORI DASAR

PENGOLAHAN DATABASE

Perintah **SELECT** adalah *query MySQL* yang paling sering gunakan. Query ini berfungsi untuk menampilkan data dari **database**. Data yang ditampilkan MySQL biasanya diberikan dalam bentuk tabel yang terdiri dari baris dan kolom.

Untuk menampilkan data MySQL di dalam PHP, hasil dari fungsi **mysql_query()** harus diproses lebih lanjut agar bisa ditampilkan di dalam web browser. PHP menyediakan banyak cara untuk menampilkan hasil query MySQL.

Berikut adalah fungsi untuk yang disediakan PHP untuk mengelolah data ke MYSQL.

1. **mysql_query()**

adalah fungsi yang mengirimkan perintah SDL ke database yang kita masuki, sebagaimana layaknya bila kita memasukkan perintah MySQL langsung dari mesin server.

Syntaxnya:

```
int mysql_query(string query, int [[link_identifier] ] );
```

2. **mysql_db_query**

adalah hampir sama halnya dengan **mysql_query** yaitu yang berfungsi mengirimkan pernyataan SQL ke server data base MYSQL selama nama database masih aktif.

Syntaxnya:

```
int mysql_db_query(string query, int [[link_identifier] ] );
```

Menampilkan Data Dari Database

1. **mysql_fetch_array()**

adalah fungsi yang memberikan hasil array dari data yang kita miliki.

Syntaxnya:

```
array mysql_fetch_array(int result, int [result_type] );
```

2. **array mysql_fetch_row**

Syntaxnya:

```
array mysql_fetch_row(int result, int [result_type] );
```

3. **mysql_num_fields()**

adalah fungsi yang memberikan jumlah field dari data yang diambil.

Syntaxnya:

```
int mysql_num_fields(int result);
```

4. **mysql_num_row()**

adalah fungsi yang memberikan jumlah baris dari data yang diambil.

Syntaxnya:

```
int mysql_num_row(int result);
```

Membuat koneksi ke Mysql

Sebelumnya buatlah database dengan phpmyadmin dengan nama 'kemahasiswaan yang berisi tabel dengan nama "mahasiswa" yang berisi field nama,alamat,no_hp.

Lakukan percobaan dengan membuat kode dengan langkah-langkah berikut:

Buatlah sebuah Program koneksi yang digunakan untuk menghubungkan halaman Web kedalam Server MySQL kita, berikut listing lengkapnya:

```
<?php
// nama file koneksi.php
$host = "localhost";
$usermysql="tamu";
$passwordmysql="tamu";

$koneksi=mysql_connect($host,$usermysql,$passwordmysql)
                        or die (" Tidak dapat konek ke server MySQL");
mysql_select_db(mahasiswa)
                        or die ("Database tidak ada");
?
```

a. Membuat halaman Utama Pencarian

Setelah Program pembuka koneksi selesai dibuat maka sekarang Kita akan mencoba untuk membuat halaman utama yang digunakan untuk memasukan data yang ingin Kita cari. Untuk lebih jelasnya ikuti langkahnya sebgai berikut.

Bukalah halaman baru pada editor Anda, dan kemudian ketikkan program berikut ini lalu simpan dengan nama form_pencarian.php,

```
<html>
<head>
<title>Halaman Pencarian</title>
</head>

<body>
<form name="form1" method="post" action="mesin_pencarian.php">
<table width="393" border="1">
<tr>
<td align="center" bgcolor="#00FFFF"><strong>Pencarian data
Mahasiswa</strong></td>
</tr>
<tr>
<td align="center">Nama Mahasiswa
<input name="data_cari" type="text" id="data_cari" size="20" maxlength="30">
<input type="submit" name="Submit" value="Cari">
</td>
</tr>
</table>
</form>
</body>
</html>
```

b. Menampilkan Hasil Pencarian

Setelah halaman utama pencarian selesai dibuat, maka sekarang Kita akan membuat mesin program yang digunakan untuk melakukan pencarian dari data yang dimasukkan pada form utama. Berikut langkahnya.

Buatlah halaman baru, dan ketikkan program berikut

```
<html>
<head>
<title>Mesin Pencarian</title>
</head>
<body>

<table width="500" border="1">
<tr align="center" bgcolor="#00FFFF">
<td colspan="4"><strong>Laporan Mahasiswa</strong></td>
</tr>
<tr>
<td width="104" height="23"><strong>No Mhs</strong></td>
<td width="132"><strong>Nama</strong></td>
<td width="125"><strong>Kode Jurusan</strong></td>
<td width="111"><strong>Kode Propinsi</strong></td>
</tr>
<?php
```

```

include_once "conneksi.php";
$query = "SELECT * FROM mahasiswa
        WHERE nama LIKE '%$data_cari%'
        ORDER BY no_mhs";
$eksekusi= mysql_query ($query) or
die ("Permintaan gagal dilakukan ");
while($hasil=mysql_fetch_array($eksekusi))
{
?>
<tr>
<td height="22"><?php echo "$hasil[no_mhs]"; ?></td>
<td><?php echo "$hasil[nama]"; ?></td>
<td><?php echo "$hasil[alamat]"; ?></td>
<td><?php echo "$hasil[no_hp]"; ?></td>
</tr>
<?php } ?>
</table>
<p>&nbsp;</p>
</body>
</html>

```

Simpanlah Script diatas dengan nama *mesin_pencarian.php*,

c. Jalankan perintah yang telah dibuat melalui browser kemudian analisa hasilnya.
Menampilkan isi data Mysql dengan PHP

Sebelumnya buatlah database dengan phpmyadmin dengan nama 'tokoabc' yang berisi tabel dengan nama "barang" yang berisi field kodebarang,namabarang,stok.

Lakukan percobaan dengan membuat kode dengan langkah-langkah berikut:

Membuat koneksi dengan MySQL (hubung.php)

```

<?mysql_connect("localhost", "", "");
mysql_select_db("tokoabc");

?>

```

a. Membuat file untuk menampilkan isi dari MySQL (tampil.php)

```
<html>
<head>
<title> ABC Online Shop</title>
</head>
<body>
<BR>
DAFTAR BARANG <BR>
<?
include_once "hubung.php";
$ambil="select kode_brg,nama_brg  from barang limit  0,10";
$lakukan=mysql_query($ambil);
while($hasil=mysql_fetch_array($lakukan))
{
?>
Kode Barang :<? echo $hasil[kode_brg]; ?><BR>
Nama Barang : <? echo $hasil[nama_brg]; ?><BR><BR>
<?}?>
</body>
</html>
```

b. Jalankan perintah yang telah dibuat melalui browser kemudian analisa hasilnya.

3. LATIHAN

Lakukan percobaan pada dasar teori dan analisa hasilnya.

Modul VII

Memasukan data di MYSQL dengan PHP

1. TUJUAN

- Agar mahasiswa dapat memahami bagaimana cara membuat form pemasukan data dengan PHP.
- Agar mahasiswa dapat memahami cara merekam data pada database

2. TEORI DASAR

PHP (PHP Hypertext PreeProsesor) merupakan sebuah bahasa Interpreter yang berjalan samping server atau yang sering disebut dengan Server Site. Hasil dari program ini berupa sebuah Script yang hanya dapat dijalankan pada Browser (IE, Mozilla, Netscape, dll)

Untuk dapat menjalankan Script-Script PHP kita harus meletakkan Script tersebut pada sebuah direktori Root yang disediakan oleh Server Web. Bentuk-bentuk Web Server yang dapat digunakan untuk menjalankan PHP adalah Apache, IIS, ataupun Xitami.

Karna program PHP merupakan program yang berjalan disamping server seperti Apache, maka untuk dapat menterjemahkan Code-code programnya komputer anda juga harus terinstal paket PHP itu sendiri, Sorce Code dari paket PHP dapat diperoleh pada alamat <http://www.php.net>, dan dapat memperoleh Apache pada alamat <http://www.apache.org> .

Untuk lebih jelasnya ada baiknya kita teruskan pembahsan bab ini dengan membuat sebuah halaman masukan terhadap database yang telah terbut sebelumnya(apabila belum baca lagi pada Modul 1). Langkah-langkah yang harus anda kerjakan adalah.

Persiapan Memasukan Data

Sebelum melangkah lebih jauh ada baiknya anda persiapkan dulu ruang yang akan digunakan untuk menyimpan program yang akan dibuat. Berikut langkahnya.

1. Buatlah Direktori dengan nama Bdata, direktori ini merupakan direktori utama untuk latihan bab-bab berikutnya.
2. Selanjutnya didalam direktori Bdata, buatlah Sub Direktori dengan nama Bab3
3. Untuk mengetikan program-program yang akan dibahas pada pertemuan selanjutnya anda akan menggunakan Note Pad, Maka sekarang bukalah NotePad anda(apabila menggunakan Linux maka daat menggunakan Edotor VI atau Mcredit).
4. Buatlah sebuah Program Coneksi yang digunakan untuk menghubungkan halaman Web kedalam Server MySQL kita, gunakan NotePad atau VI(untuk linux) sebagai edotornya. Berikut adalah listing programnya.

```
<?php
// nama file koneksi.php
$host = "localhost";
$usermysql="tamu";
$passmysql="tamu";
```

```

$koneksi=mysql_connect($host,$usermysql,$passmysql)
                    or die (" Tidak dapat konek ke server MySQL");
mysql_select_db(kemahasiswaan)
                    or die ("Database tidak ada");

```

?

- Setelah selesai menuliskannya maka anda dapat menyimpan program diatas dengan nama *koneksi.php* dan diletakan pada direktori **Bab3** pada root direktori anda(Bdata), yang harus diingat sebelum menyimpan anda harus menentukan Type File dari Note Pad anda menjadi All File.

Merancang Form dan Membuat Program

Setelah persiapan yang mencakup penentuan direktori dan Koneksi Server selesai anda buat, maka sekarang anda dapat melanjutkannya dengan melakukan perancangan Form dan Program penyimpan-nya. Sekarang kita akan membuat Formulir masukan data Jurusan, langkahnya adalah

- Bukalah halaman baru pada NotePad anda, selanjutnya ketikkan programnya seperti berikut.

```

<html>
<head>
<title>Halaman Masukan Mahasiswa</title>
</head>

<body>
<form name="form1" method="post" action="sim_jurusan.php">
  <table width="300" height="77" border="1">
    <tr bgcolor="#00FFFF">
      <td colspan="2"><strong>Masukan data Jurusan</strong></td>
    </tr>
    <tr>
      <td width="91">Kode Jurusan</td>
      <td width="193">
        <input name="txt_kdjur" type="text" id="txt_kdjur" size="10" maxlength="4"></td>
      </tr>
    <tr>
      <td>Jurusan</td>
      <td>
        <input name="txt_jurusan" type="text" id="txt_jurusan" size="20" maxlength="30">
      </td>
    </tr>
    <tr>
      <td>&nbsp;</td>
      <td>

```

```

<input type="submit" name="Submit" value="Simpan"></td>
</tr>
</table>
</form>
</body>
</html>

```

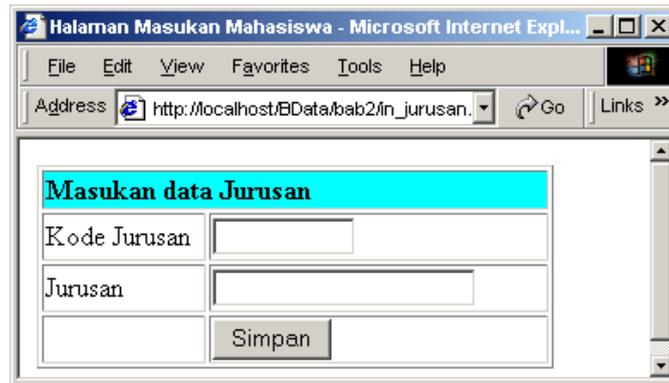
2. Setelah Program diatas selesai anda buat, maka sekarang simpanlah pada direktori Bab3 dengan nama *in_jurusan.php*.
3. Pada program diatas memiliki bentuk Form yang Action atau pengalamatanya menuju halaman sim_jurusan.php, Listing dari program ini adalah.

```

<?php
include_once "conneksi.php";
$masukan= "INSERT INTO jurusan
            (kd_jur,jurusan) VALUES
            ('$txt_kdjur','$txt_jurusan')";
if (mysql_query($masukan,$koneksi))
{
    echo "Data dengan kode $txt_kdjur telah disimpan ";
    include "in_jurusan.php";
}
else
{
    echo " Data dengan kode $txt_kdjur gagal disimpan";
    include "in_jurusan.php";
}
?>

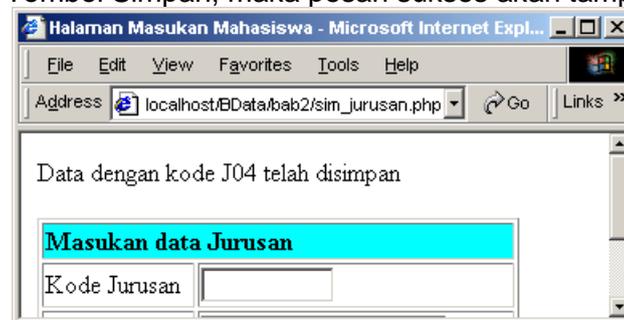
```

4. Simpanlah Program diatas dengan menggunakan nama sim_jurusan.php, tempatkan juga pada diektori Bab3.
5. Setelah semuanya selesai maka anda dapat menjalankan program diatas dengan menggunakan Browser (Internet Explorer) dengan mengetikan alamat http://localhost/Bdata/Bab3/in_jurusan.php . Apabila sukses maka halaman akan tampak seperti gambar berikut.



Gambar 2.1 Halaman Masukan data jurusan

6. Cobalah untuk memasukan sebuah data dengan mengisi kode Jurusan dan Jurusan, kliklah Tombol Simpan, maka pesan sukses akan tampak seperti.



Gambar 2.2 Halaman Pesan data telah tersimpan

3. LATIHAN

untuk melakukan pemasukan data-data propinsi, agar tidak mempersulit maka anda dapat menggunakan program sebelumnya dengan mengubah dan disimpan dalam nama file yang baru. Untuk lebih jelasnya ikuti langkah-langkah berikut.

1. Bukalah halaman baru pada NotePad anda, dan selanjutnya ketikkan program masukan data Propinsi seperti pada Script berikut.

```
<html>
<head>
<title>Halaman Masukan Propinsi</title>
</head>

<body>
<form name="form1" method="post" action="sim_propinsi.php">
  <table width="300" height="77" border="1">
    <tr bgcolor="#00FFFF">
      <td colspan="2"><strong>Masukan data Propinsi</strong></td>
    </tr>
```

```

<tr>
  <td width="91">Kode Propinsi</td>
  <td width="193">
    <input name="txt_kdprop" type="text" id="txt_kdprop" size="10"
maxlength="3"></td>
</tr>
<tr>
  <td>Propinsi</td>
  <td><input name="txt_propinsi" type="text" id="txt_propinsi" size="20"
maxlength="35"></td>
</tr>
<tr>
  <td>&nbsp;</td>
  <td><input type="submit" name="Submit" value="Simpan"></td>
</tr>
</table>
</form>
</body>
</html>

```

2. Simpanlah Program diatas dengan nama file in_jurusan.php, tempatkan program diatas pada direktori Bdata>Bab3
3. Setelah Formulir masukan data Propinsi selesai dibuat, sekrang anda dapat membuat halaman eksekusi penyimpanan data-data propinsi dengan Listing sebagai berikut

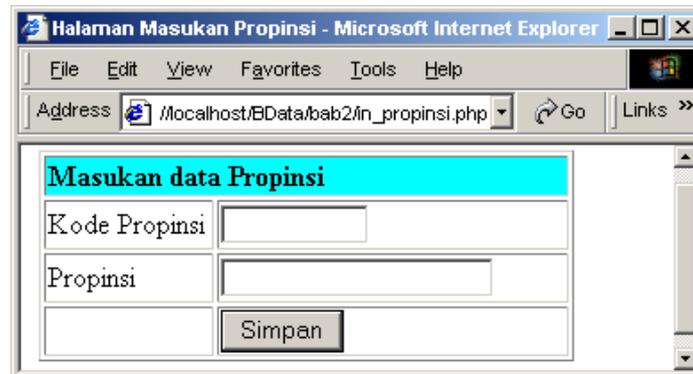
```

<?php
include_once "conneksi.php";

$masukan= "INSERT INTO propinsi
          (kd_prop,propinsi) VALUES
          ('$txt_kdprop','$txt_propinsi)";
if (mysql_query($masukan,$koneksi))
{
  echo "Data dengan kode $txt_kdprop telah disimpan ";
  include "in_propinsi.php";
}
else
{
  echo " Data dengan kode $txt_kdprop gagal disimpan";
  include "in_propinsi.php";
}
?>

```

4. Simpanlah Program diatas pada direktori *Bdata* > *Bab3*, dengan nama *sim_propinsi.php*.
5. Seltelah selesai maka program dapat jijalankan dengan mengetikan alamat apda Address browser anda http://localhost/BData/Bab3/in_propinsi.php, berikut adalah gambar hailnya.



Gambar 2.3 Halaman masukan Propinsi

Ketikkan script berikut dan simpan dengan nama **form_input.php**

form_input.php

```
<HTML>
<HEAD>
Pemasukan Data Barang
</HEAD>
<TITLE>
Pemasukan data Barang dan stok
</TITLE>
<BODY>
<form name='form1' metode="post" action='simpan_barang.php'>
<table width='400' height='77' border='1'>
<tr><td width='100'>Kode Barang</td>
<td ><input name='in_kode_brg' type='text' size='1';
maxlength='4'></td></tr>
<tr><td width='100'>Nama Barang</td>
<td><input name='in_nama_brg' type='text' size='31;
maxlength='30'></td></tr>
<tr><td width='100'>Stok</td>
<td ><input name='in_stok' type='text' size='5' ></td></tr>
<tr><td ><input name='submit' type='submit' value='simpan'></td></tr>
</table>
</form>
</BODY>
</HTML>
```

Ketikkan script berikut dan simpan dengan nama **simpan_barang.php**

simpan_barang.php

```
<?php
include_once "hubung.php";
$rekam="insert into barang(kodebarang,namabarang,stok) values
('$in_kode_brg','$in_nama_brg','$in_stok)";
mysql_query($rekam);
?>
```

Jalankan perintah yang telah dibuat melalui browser kemudian analisa hasilnya.

MODUL VIII EDIT,DELETE DATA MYSQL DENGAN PHP

1. TUJUAN

- Mahasiswa mampu mengedit data dan hapus data dimysql dengan PHP
- Mahaiswa mampu membuat aplikasi menghapus data dimysql dengan PHP

2. TEORI DASAR

Edit Data

Pemeliharaan data pada tahap implementasi suatu website merupakan kewajiban dari pemiliknya. Hal ini diperlukan agar data dan informasi yang ada pada website selalu uptodate.

Untuk kebutuhan tersebut maka digunakan alat bantu yaitu fasilitas edit data.

Contoh Algoritma edit data :

- (1) mencari data yang akan diedit.
Proses pencarian menggunakan perintah query yaitu : `select <namafield> from <namatable> where <kondisi>`; setelah query dijalankan akan memberikan *feedback* berupa *recordset* yang berisi data hasil *query*.
- (2) melakukan pengeditan menggunakan halaman *form edit*.
Data yang telah ditemukan kemudian ditampilkan pada *element form*. Dan dilakukan pengeditan data.
- (3) setelah data diedit lalu dilakukan proses *update* dengan *query update*.
- (4) untuk menampilkan data terbaru dapat menggunakan halaman output.

Delete Data

Ada saat ketika suatu data atau informasi tidak lagi dibutuhkan. Untuk itu diperlukan sebuah mekanisme delete data yang mana akan menghapus record pada database.

Algoritma delete hampir sama dengan edit data, namun pada langkah ke 3 tidak melakukan pengeditan tapi melakukan proses delete yang dapat menggunakan perintah *query delete from table*.

Contoh algoritma delete data :

- (1) mencari data yang akan diedit.
Proses pencarian menggunakan perintah query yaitu : `select <namafield> from <namatable> where <kondisi>`; setelah *query* dijalankan akan memberikan *feedback* berupa *recordset* yang berisi data hasil *query*.
- (2) untuk keamanan proses maka dapat ditambahkan halaman konfirmasi yang berfungsi untuk menampilkan data yang akan di delete.
- (3) melakukan perintah delete dengan `query delete form table`.
- (4) untuk menampilkan data terbaru dapat menggunakan halaman output.

Menampilkan Isi Tabel

Isi tabel dapat ditampilkan dengan menggunakan perintah SELECT, cara penulisan perintah SELECT adalah:

```
select kolom from namatable;
```

Contoh:

- Untuk menampilkan kolom (field) nomor dan nama pada tabel anggota
`select nomor, nama from anggota;`

- Untuk menampilkan semua kolom(field) pada tabel anggota
select * from anggota;
- Untuk menampilkan semua kolom pada tabel anggota yang berada pada kota 'Surabaya'
select * from anggota where kota='Surabaya';
- Untuk menampilkan semua kolom pada tabel anggota dengan urut nama
select * from anggota order by nama;
- Untuk menghitung jumlah record pada tabel anggota
select count(*) from anggota;
- Untuk menampilkan kota pada tabel anggota
select kota from anggota;
- Untuk menampilkan kota dengan tidak menampilkan kota yang sama pada tabel anggota
select distinct kota from anggota;
- Untuk menampilkan nama dan email yang mempunyai email di 'yahoo.com'
select nama,email from anggota where email like'%yahoo.com';

Memodifikasi Record

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

```
update namatabel set kolom1=nilaibaru1, kolom2=nilaibaru2 ... where kriteria;
```

Contoh:

Merubah e-mail dari anggota yang bernomor 12 menjadi 'supri@yahoo.com' dalam tabel anggota.

```
update anggota set email='supri@yahoo.com' where id='12';
```

3. LATIHAN

1. Buatlah database dengan nama "praktikumdb" *tanda tanda petik
2. Aktifkan / masuk di database yang dibuat dan masukkan query dibawah ini

```
CREATE TABLE IF NOT EXISTS anggota (
  id int(11) NOT NULL AUTO_INCREMENT,
  nama varchar(150) NOT NULL,
  alamat text,
  email varchar(150) DEFAULT NULL,
  telp varchar(150) DEFAULT NULL,
  gender int(1) DEFAULT NULL,
  PRIMARY KEY (id)
) ENGINE=MyISAM DEFAULT CHARSET=latin1 AUTO_INCREMENT=3 ;
```

```
INSERT INTO anggota (id, nama, alamat, email, telp, gender) VALUES
(1, 'Joko', 'Jln. Apasaja no 2', 'xxx123@xxx.com', '0237-412123', 1),
(2, 'Herman', 'Jln. Kabeta No 10. Madiun', 'xxx@xxxx.com', '08677666612',2);
```

```
CREATE TABLE IF NOT EXISTS forum (
```

```

idf int(11) NOT NULL AUTO_INCREMENT,
nama varchar(50) DEFAULT NULL,
pesan text,
PRIMARY KEY (idf)
) ENGINE=MyISAM DEFAULT CHARSET=latin1 AUTO_INCREMENT=4 ;

```

```

INSERT INTO forum (idf, nama, pesan) VALUES
(1, 'Anang', 'Hanya mencoba saja, kalau bisa ya sudah. '),
(2, 'Anonymous', 'Hai, apakabar semua'),
(3, 'Herlin', 'Hei, pa kabar semua, baikkah semua ?');

```

3. Kemudian klik GO / execute (pd PHPMysqlAdmin)

4. koneksi.php

```

<?php
// buka koneksi ke database server
$databasename="praktikumdb"; // sesuaikan dengan database
sendiri
$hostname="localhost"; // sesuaikan
$username="root"; // sesuaikan
$password=""; //sesuaikan
if (!$dbh=mysql_connect($hostname,$username,$password))
{
echo mysql_error();
exit;
}else {
// select default database
mysql_select_db($databasename, $dbh);
echo "<h1><b>selamat anda sukses mengakses
databasename</b></h1><br/><hr/>";
}
?>

```

5. edit_db.php

```

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<title>edit_db</title>
</head>
<body>
<table width="75%" border="0" align="center" cellpadding="0"
cellspacing="0" bgcolor="#FFFFCC">
<tr align="center" bgcolor="#CCCCCC">
<td height="19" colspan="2" valign="top">
<? include "koneksi1.php"; ?>
</td>
</tr>
<tr>
<td width="123" height="19" align="center" valign="top"

```

```

bgcolor="#CCCCCC"><font color="#000000">Menu</font></strong></td>
<td width="458" rowspan="4" valign="top" bgcolor="#FFFFFF">
<?
echo "<table border=0 align=center cellpadding=3 cellspacing=0 bgcolor
=#ffffcc>
<tr><h3>pilih data anggota yang akan di edit atau dihapus</h3></tr>
<tr bgcolor=#cccccc>
<td>No.</td>
<td><div align=center>Nama</div></td>
<td>Alamat</td>
<td>E-Mail</td>
<td>telp</td>
<td>gender</td>
<td>order</td></tr>";
//perintah query
$sql="select nama from anggota order by id";
$qry = mysql_query($sql);
//menampilkan data tabel anggota
while ( $a = mysql_fetch_array ( $qry ) ) {
$i++;
echo ("<tr><td><center><font face=verdana size=3>$i.</td>");
echo ("<td><font face=verdana size=3>$a[nama]</td>");
echo ("<td><font face=verdana size=3>$a[alamat]</td>");
echo ("<td><font face=verdana size=3>$a[email]</td>");
echo ("<td><font face=verdana size=3>$a[telp]</td>");
if($a[gender]=='1'){
$gender='male';
}else{ $gender='female';}
echo ("<td><font face=verdana size=3>$gender</td>");
echo "<td><a href=\"paneleditdb.php?id=$a[id]\">edit</a></td>";
echo "<td><a href=\"paneldeltdb.php?id=$a[id]\">delete</a></td></tr>";
}
echo "</table>";
mysql_close();
?>
</td>
</tr>
<tr>
<td height="16" align="center" valign="top"><a href="input_anggota.php"
target="_self">Input
anggota</a></td>
</tr>
<tr>
<td height="18" align="center" valign="top"><a href="lihatdb.php"
target="_self">Lihat
anggota</a></td>
</tr>
<tr>
<td height="147" valign="top"></td>
</tr>

```

```

<tr bgcolor="#CCCCCC">
<td height="19" colspan="2" valign="top"></td>
</tr>
</table>
</body>
</html>

```

6. paneleditdb.php

```

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<title>paneleditdb</title>
</head>
<body>
<table width="667" border="0" align="center" cellpadding="0"
cellspacing="0" bgcolor="#FFFFCC">
<tr align="center" bgcolor="#CCCCCC">
<td height="19" colspan="2" valign="top">
<? include "koneksi.php";
$sql="select * from anggota where Id=$id";
$hasil = mysql_query($sql) or die ("gagal operasi");
$a = mysql_fetch_array ( $hasil );
?>
</td> </tr> <tr>
<td width="123" height="19" align="center" valign="top"
bgcolor="#CCCCCC"><font color="#000000">Menu</font></strong></td>
<td width="458" rowspan="4" valign="top" bgcolor="#FFFFFF">
<form action="proseseditdb.php" method="post" name="form1"
target="_self">
<table width="100%" border="0" align="center" cellpadding="4"
cellspacing="0">
<tr bgcolor="#8B9AA1">
<th colspan="2" class="navi" scope="row">PANEL EDIT
DATABASE</th>
</tr> <tr>
<th class="content" scope="row"><div
align="right">ID</div></th>
<td>
<? if ($hasil){
echo "<input name=\"txtid\" type=\"text\" id=\"txtid\"
value=\"$a[id]\" size=\"15\" maxlength=\"15\">";
}
?>
</td> </tr> <tr>
<th class="content" scope="row"><div
align="right">Nama</div></th>
<td>
<? if ($hasil){
echo "<input name=\"txtnama\" type=\"text\" id=\"txtnama\"

```

```

value="\$a[nama]" size="35" maxlength="35">";
}
?>
</td> </tr> <tr> <th class="content" scope="row"><div
align="right">Alamat</div></th>
<td>
<?
if ($hasil){
echo "<textarea name='txtalamat' cols='35'
id='txtalamat'>\$a[alamat]</textarea>";
}
?>
</td> </tr> <tr> <th class="content" scope="row"><div
align="right">Email</div></th>
<td><? if ($hasil){
echo "<input name='txtemail' type='text' id='txtemail'
value='\$a[email]' size='35' maxlength='35'>";
}
?>
</tr> <tr> <th class="content" scope="row"><div
align="right">Telp</div></th>
<td>
<? if ($hasil){
echo "<input name='txttelp' type='text' id='txttelp'
value='\$a[telepon]' size='35' maxlength='35'>";
}
?>
</tr><tr><td scope="row"><div
align="right"><strong>Gender</strong></div></td>
<td>
<?
if (\$a[gender]==='1'){
echo " <input name='rbgender' type='radio' value='1'
checked> Laki - laki";
echo "<input name='rbgender' type='radio'
value='0'>Perempuan</td>";
}else{
echo " <input name='rbgender' type='radio' value='1' >
Laki - laki";
echo "<input name='rbgender' type='radio' value='0' checked
>Perempuan</td>";
}
?>
</td>
</tr>
<tr>
<td scope="row"></td>
<td scope="row"> <input type="submit" name="Submit" id="Submit"
value="update" />
<input name="reset" type="reset" id="reset2" value="Reset" />

```

```

<input name="batal" type="button" id="batal" value="batal"
onclick=javascript:window.location.href="edit_db.php" /></th>
</tr>
</table>
</form>
</td>
</tr>
<tr>
<td height="16" align="center" valign="top"><a href="input_anggota.php"
target="_self">Input
anggota</a></td>
</tr>
<tr>
<td height="18" align="center" valign="top"><a href="lihatdb.php"
target="_self">Lihat
anggota</a></td>
</tr>
<tr>
<td height="147" valign="top"></td>
</tr>
<tr bgcolor="#CCCCCC">
<td height="19" colspan="2" valign="top"></td>
</tr>
</table>
</body>
</html>

```

7. proseseditdb.php

```

<?
include("koneksi.php");
if ($txtid<>""){
$sql="update anggota set 'nama'='$txtnama', alamat='$txtalamat',
email='txtemail',gender='$rbgender' where id='$txtid'";
$hasil = mysql_qlery($sql) or die ("gagal operasi");
echo "
<script language =\"javascript\" >
window.location.href=\"edit_db.php\";
</script>";
}else{
echo "
<script language =\"javascript\" >
self.history.back();
</script>";
}
?>

```

8. paneldelldb.php

```

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

```

```

<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<title>paneldelete_db</title>
</head>
<body>
<table width="667" border="0" align="center" cellpadding="0"
cellspacing="0" bgcolor="#FFFFCC">
<tr align="center" bgcolor="#CCCCCC">
<td height="19" colspan="2" valign="top">
<? include "koneksi.php";
$sql="select * from anggota where Id=$id";
$hasil = mysql_query($sql) or die ("gagal operasi");
$a = mysql_fetch_array ( $hasil );
?>
</td> </tr> <tr>
<td width="123" height="19" align="center" valign="top"
bgcolor="#CCCCCC"><font color="#000000">Menu</font></strong></td>
<td width="458" rowspan="4" valign="top" bgcolor="#FFFFFF">
<? echo "<form action=\"prosesdeldb.php?id=$a[Id]\" method=\"post\"
name=\"form1\" target=\"_self\">";
?>
<table width="100%" border="0" align="center" cellpadding="4"
cellspacing="0">
<tr bgcolor="#8B9AA1">
<th colspan="2" class="navi" scope="row">KONFIRMASI DELETE
DATABASE</th>
</tr><tr>
<th class="content" scope="row"><div align="right">ID</div></th>
<td>
<? if ($hasil){
echo "<font color=\"red\"><b>$a[Id]</b></font>";
}
?>
</td> </tr><tr> <th class="content" scope="row"><div
align="right">Nama</div></th>
<td>
<? if ($hasil){
echo "<font color=\"red\"><b>$a[nama]</b></font>";
}
?>
</td></tr> <tr> <th class="content" scope="row"><div
align="right">Alamat</div></th>
<td> <?if ($hasil){
echo "<font color=\"red\"><b>$a[alamat]</b></font>";
}
?>
</td> </tr> <tr> <th class="content" scope="row"><div
align="right">Email</div></th>
<td><? if ($hasil){
echo "<font color=\"red\"><b>$a[email]</b></font>";

```

```

}
?>
</tr><tr> <th class="content" scope="row"><div
align="right">Telp</div></th>
<td>
<? if ($hasil){
echo "<font color=\red\"><b>$a[teelp]</b></font>";
}
?>
</tr><tr> <td scope="row"><div
align="right"><strong>Gender</strong></div></td>
<td> <?
if ($b[gender]==1){
echo "<font color=\red\"><b>laki-laki</b></font>";
}else{
echo "<font color=\red\"><b>perempuan</b></font>";
}
?> </td> </tr> <tr> <td scope="row"></td>
<td scope="row"> <input type="submit" name="Submit" id="Submit"
value="hapus" />
<input name="batal" type=button id="batal" value="batal"
onclick=javascript:window.location.href="edit_db.php" /></th>
</tr></table></form>
</td> </tr> <tr> <td height="16" align="center" valign="top"><a
href="input_anggota.php" target="_self">Input
anggota</a></td>
</tr> <tr> <td height="18" align="center" valign="top"><a
href="lihatdb.php" target="_self">Lihat anggota</a></td></tr>
<tr>
<td height="147" valign="top"></td>
</tr>
<tr bgcolor="#CCCCCC">
<td height="19" colspan="2" valign="top"></td>
</tr>
</table>
</body>
</html>

```

9. prosedeldb.php

```

<?
include("koneksi.php");
if ($id<>""){
$sql="delete form anggota where Id='$id'";
$hasil = mysql_query($sql) or die ("gagal operasi");
echo " <script language =\javascript\"
>window.location.href=\edit_db.php\"; </script>";
}else{
echo "
<script language =\javascript\" >
self.history.back();

```

```

</script>";
}
?>

```

10. inputanggota.php

```

<?
include("koneksi.php");
if ($submit){
function exec_time(){
$time = explode(" ",microtime());
$msec = (double)$time[0];
$sec = (double)$time[1];
return $sec + $msec; }
$start_exec = exec_time(); //mulai menghitung waktu eksekusi
query
$sql="insert into anggota (nama,alamat,email,telp,gender)
values
('$txtnama','$txtalamat','$txtemail','$txttelp','$rgender')";
;
$hasil = mysql_query($sql) or die ("insert user $txtnama
GAGAL, harap coba lagi");
if ($hasil){
echo "<center><H2>SELAMAT ANDA TELAH BERHASIL DIDAFTAR
DISITUS INI</h2></center>";
}
else{
echo "PROSES GAGAL HARAP ULANGI LAGI!";
}
$end_exec = exec_time();//mencatat waktu query diselesaikan
$run_time=$end_exec - $start_exec;// menghitung waktu
}
?>
<form method=post action="inputanggota.php">
<table width="278" border="0" align="center"
cellpadding="0" cellspacing="0" bgcolor="#FFFFFF">
<tr bgcolor="#00CCFF">
<td colspan="2"><div align="left">
<h3>Pendaftaran Anggota Baru </h3>
</div></td>
</tr>
<tr>
<td width="53" valign="top"><strong>Nama</strong></td>
<td width="209"><input name="txtnama" type="text"
id="txtnama" size="15" maxlength="15"></td>
</tr>
<tr>
<td valign="top"><strong>Alamat</strong></td>
<td><textarea name="txtalamat" cols="35"
id="txtalamat"></textarea></td>
</tr>

```

```

<tr>
<td height="23"
valign="top"><strong>Email</strong></td>
<td><input name="txtemail" type="text" id="txtemail"
size="30" maxlength="30"></td>
</tr>
<tr>
<td valign="top"><strong>Telp</strong></td>
<td><input name="txttelp" type="text" id="txttelp"
size="20" maxlength="20"></td>
</tr>
<tr>
<td valign="top"><strong>Gender</strong></td>
<td><input name="rgender" type="radio" value="1"
checked>
Male
<input name="rgender" type="radio" value="0">
Female</td>
</tr>
<tr>
<td></td>
<td><input name="submit" type="submit" id="submit"
value="Daftar">
<input name="btnreset" type="reset" id="btnreset"
value="Ulang">
<a href="lihatanggota.php">Lihat Anggota</a> </td>
</tr>
<tr>
<td><strong>Server</strong></td>
<td><? echo $HTTP_SERVER_VARS["REMOTE_ADDR"]; //
mendeteksi ip address user?> </td>
</tr>
</table>
</form>
<? if ($submit){
echo "<center><b>Waktu eksekusi query = $run_time
detik</b></center>";}
?>
11. lihatanggota.php
<?
include("koneksi.php");
echo "<table border=1 align=center cellpadding=0
cellspacing=0 bgcolor =#ffffcc>
<tr><h3>ini adalah data anggota</h3></tr>
<tr bgcolor=#00ccff>
<td>No.</td>
<td><div align=center>Nama</div></td>
<td>Alamat</td>
<td>E-Mail</td>
<td>telp</td>

```

```

<td>gender</td></tr>";
//perintah query
$sql="select * from anggota order by id";
$qry = mysql_query($sql);
//menampilkan data tabel anggota
while ( $a = mysql_fetch_array ( $qry ) ) {
    $i++;
    echo ("<tr><td><center><font face=verdana size=3>$i.</td>");
    echo ("<td><font face=verdana size=3>$a[nama]</td>");
    echo ("<td><font face=verdana size=3>$a[alamat]</td>");
    echo ("<td><font face=verdana size=3>$a[email]</td>");
    echo ("<td><font face=verdana size=3>$a[telp]</td>");
    if($a[gender]=='1'){
        $gender='male';
    }else{ $gender='female';}
    echo ("<td><font face=verdana size=3>$gender</td></tr>");
}
echo "</table>";
echo "<center><a href=\"inputanggota.php\">kembali ke
input</a></center>";
mysql_close();//menutup koneksi database mysql
?>

```