



MODUL PEMBELAJARAN PRAKTEK BASIS DATA (MySQL)

BAB I

MENGENAL MySQL

1.1 Mengenal MySQL

MySQL adalah Sebuah program database server yang mampu menerima dan mengirimkan datanya sangat cepat, multi user serta menggunakan perintah dasar SQL (Structured Query Language).

MySQL merupakan dua bentuk lisensi, yaitu FreeSoftware dan Shareware. MySQL yang biasa kita gunakan adalah MySQL FreeSoftware yang berada dibawah Lisensi GNU/GPL (General Public License).

MySQL Merupakan sebuah database server yang free, artinya kita bebas menggunakan database ini untuk keperluan pribadi atau usaha tanpa harus membeli atau membayar lisensinya. MySQL pertama kali dirintis oleh seorang programmer database bernama **Michael Widenius** . Selain database server, MySQL juga merupakan program yang dapat mengakses suatu database MySQL yang berposisi sebagai Server, yang berarti program kita berposisi sebagai Client. Jadi MySQL adalah sebuah database yang dapat digunakan sebagai Client maupun server.

Database MySQL merupakan suatu perangkat lunak database yang berbentuk database relasional atau disebut Relational Database Management System (RDBMS) yang menggunakan suatu bahasa permintaan yang bernama SQL (Structured Query Language).

1.2 Kelebihan MySQL

Database MySQL memiliki beberapa kelebihan dibanding database lain, diantaranya :

- MySQL merupakan Database Management System (DBMS)
- MySQL sebagai Relation Database Management System (RDBMS) atau disebut dengan database Relational
- MySQL Merupakan sebuah database server yang free, artinya kita bebas menggunakan database ini untuk keperluan pribadi atau usaha tanpa harus membeli atau membayar lisensinya
- MySQL merupakan sebuah database client
- MySQL mampu menerima query yang bertupuk dalam satu permintaan atau Multi-Threading.

- MySQL merupakan Database yang mampu menyimpan data berkapasitas sangat besar hingga berukuran GigaByte sekalipun.
- MySQL didukung oleh driver ODBC, artinya database MySQL dapat diakses menggunakan aplikasi apa saja termasuk berupa visual seperti visual Basic dan Delphi.
- MySQL adalah database menggunakan enkripsi password, jadi database ini cukup aman karena memiliki password untuk mengakses nya.
- MySQL merupakan Database Server yang multi user, artinya database ini tidak hanya digunakan oleh satu pihak orang akan tetapi dapat digunakan oleh banyak pengguna.
- MySQL mendukung field yang dijadikan sebagai kunci primer dan kunci unqi (Unique).
- MySQL memiliki kecepatan dalam pembuatan table maupun peng-update an table.

1.3 Mengenal SQL (Structured Query Language)

SQL (Structured Query Language) adalah sebuah bahasa permintaan database yang terstruktur. Bahasa SQL ini dibuat sebagai bahasa yang dapat merelasikan beberapa tabel dalam database maupun merelasikan antar database.

SQL dibagi menjadi tiga bentuk Query, yaitu :

1.3.1 DDL (Data Definition Language)

DDL adalah sebuah metode Query SQL yang berguna untuk mendefinisikan data pada sebuah Database, Query yang dimiliki DDL adalah :

- CREATE : Digunakan untuk membuat Database dan Tabel
- Drop : Digunakan untuk menghapus Tabel dan Database
- Alter : Digunakan untuk melakukan perubahan struktur tabel yang telah dibuat, baik menambah Field (Add), mengganti nama Field (Change) ataupun menamakannya kembali (Rename), dan menghapus Field (Drop).

1.3.2 DML (Data Manipulation Language)

DML adalah sebuah metode Query yang dapat digunakan apabila DDL telah terjadi, sehingga fungsi dari Query DML ini untuk melakukan pemanipulasian database yang telah dibuat. Query yang dimiliki DML adalah :

- INSERT : Digunakan untuk memasukkan data pada Tabel Database
- UPDATE : Digunakan untuk perubahan terhadap data yang ada pada Tabel Database
- DELETE : Digunakan untuk Penhapusan data pada tabel Database

1.3.3 DCL (Data Control Language)

DCL adalah sebuah metode Query SQL yang digunakan untuk memberikan hak otorisasi mengakses Database, mengalokasikan space, pendefinisian space, dan pengauditan penggunaan database. Query yang dimiliki DCL adalah :

- GRANT : Untuk mengizinkan User mengakses Tabel dalam Database.
- REVOKE : Untuk membatalkan izin hak user, yang ditetapkan oleh perintah GRANT
- COMMIT : Menetapkan penyimpanan Database
- ROLLBACK : Membatalkan penyimpanan Database

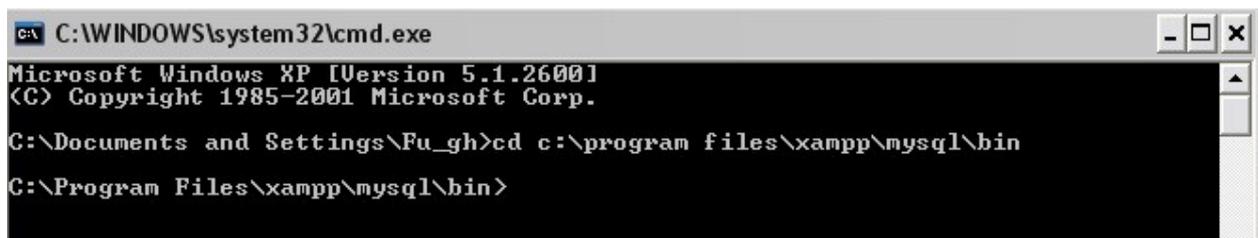
BAB 2

Fungsi – Fungsi Pada MySQL

2.1 Mengaktifkan Direktori MySQL Server

Untuk dapat menggunakan MySQL terlebih dahulu aktifkan Server MySQL dengan menghidupkan daemon MySQL. Program MySQL yang digunakan pada modul ini adalah XAMPP 1.7, maka untuk menjalankan daemon MySQL terdapat pada direktori yaitu C:\Program Files\Xampp\Mysql\Bin

Untuk masuk kedalam server MySQL, bukalah MS-DOS Prompt anda melalui Run kemudian ketik Command atau cmd. Maka anda dapat masuk ke dalam direktori MySQL melalui MS-DOS Prompt seperti dibawah ini.



```
C:\WINDOWS\system32\cmd.exe
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.
C:\Documents and Settings\Fu_gh>cd c:\program files\xampp\mysql\bin
C:\Program Files\xampp\mysql\bin>
```

2.2 Masuk dan Keluar dari Server MySQL

MySQL adalah sebuah database server yang sangat aman. MySQL memiliki kemampuan manajemen user dalam mengakses. Jadi, tidak sembarang user dapat mengakses sebuah database yang diciptakan MySQL. Maka sebelum anda memiliki User untuk mengakses MySQL anda juga dapat Mengakses database MySQL menggunakan User **Root**.

Berikut adalah perintah yang digunakan untuk mengkoneksikan kedalam Server Mysql :

```
Shell > MySQL -u Root -p
```

```
Enter Password: *****
```

Keterangan : Tanda -u menerangkan bahwa kita akan masuk menggunakan User Name bernama Root.

Tanda -p menyatakan kita akan masuk menggunakan Password.

Berikut adalah perintah yang digunakan untuk mengkoneksikan kedalam Server Mysql melalui Root :

Shell> Mysql -u root

```
C:\Program Files\xampp\mysql\bin>mysql -u root
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 4 to server version: 5.0.27-community-nt
Type 'help;' or '\h' for help. Type '\c' to clear the buffer.
```

Untuk dapat keluar dari Server MySQL kita dapat mengetikkan Intruksi quit ata \q :

Mysql> quit

Bye

Mysql> \q

Bye

2.3 Bantuan dalam MySQL

Database MySQL menyediakan beberapa fasilitas bantuan yang berguna untuk mendokumentasikan atau memanipulasikan server yaitu dengan cara mengetikkan intruksi \h atau \?.

Mysql> \?

Semua Query harus diakhiri dengan tanda titik koma (;). Tanda ini menunjukkan bahwa query telah berakhir dan siap dieksekusi.

Help (\h) : Digunakan untuk menampilkan file bantuan pada MySQL

? (\?) : Perintah ini sama dengan perintah Help

Clear (\c) : Berguna untuk membersihkan atau menggagalkan semua perintah yang telah berjalan dalam suatu prompt

Connect (\r) : untuk melakukan penyegaran koneksi ke dalam database yang ada pada Server Host

Ego (\G) : berguna untuk menampilkan data secara horizontal.

Go (\g) : member perintah server untuk mengeksekusi

tee (\T) : mengatur tempat file yang akan didokumentasikan.

Contoh :

```
mysql> \T d:\belajar mysql.doc
```

```
Logging to file 'd:\data.doc;'
```

Note (\t) : akhir dari (\T) yang berguna untuk mendokumentasikan semua query.

Print (\p) : mencetak semua query yang telah kita perintahkan kelayar.

Prompt (\R) : Mengubah prompt standar sesuai keinginan.

Source (\.) : berguna untuk mengeksekusi query dari luar yang berbentuk **.sql**

Use (\u) : berguna untuk memasuki database yang akan digunakan maupun mengganti database yang akan di gunakan.

BAB 3

Administrasi MySQL

MySQL Selaku database server yang mampu berjalan pada jaringan, tentu saja MySQL harus memiliki kemampuan khusus yang berguna untuk melakukan manajemen user atau mendukung system database yang bersifat client/server.

3.1 Membuat User baru

Untuk dapat menciptakan user baru pada database mysql yang terdapat pada tabel user. Dapat dilakukan dengan menggunakan pernyataan SQL bernama INSERT. Sintax seperti berikut :

```
INSERT INTO user(host,user,password) VALUES('%', 'nama_user', 'password');
```

Contoh :

```
mysql> INSERT INTO user(host,user,password) VALUES('localhost', 'haris', MD5('if060017'));
```

Query OK, 1 row affected, 4 warnings (0.00 sec)

Setelah anda memberikan perintah diatas, berikan perintah :

```
FLUSH PRIVILEGES;
```

Contoh :

```
mysql> FLUSH PRIVILEGES;
```

Query OK, 0 rows affected (0.00 sec)

3.2 Memberikan Wewenang Untuk User

Apabila User telah dibuat terlebih dahulu dan lupa untuk memberikan Hak Wewenang untuk User. Kita dapat memberikan hak wewenang dengan menggunakan Perintah Query UPDATE. Sintax yang digunakan seperti berikut :

```
UPDATE user
SET select_priv ='y',
      insert_priv ='y',
      update_priv ='y',
      delete_priv ='y',
      create_priv ='y',
      drop_priv ='y',
      alter_priv ='y'
WHERE user ='haris';
```

BAB 4

DDL (DATA DEFINITON LANGUAGE)

DDL adalah sebuah metode Query SQL yang berguna untuk mendefinisikan data pada sebuah Database.

4.1. Type Data pada MySQL

Tipe data adalah suatu bentuk pemodelan data yang dideklarasikan pada saat melakukan pembuatan tabel. Tipe data ini akan mempengaruhi setiap data yang akan dimasukkan ke dalam sebuah tabel. Data yang akan dimasukkan harus sesuai dengan tipe data yang dideklarasikan.

Berbagai type data pada MySQL dapat dilihat pada tabel berikut :

Type Data	Keterangan
TINYINT	Ukuran 1 byte. Bilangan bulat terkecil, dengan jangkauan untuk bilangan bertanda: -128 sampai dengan 127 dan untuk yang tidak bertanda : 0 s/d 255. Bilangan tak bertandai dengan kata UNSIGNED
SMALLINT	Ukuran 2 Byte. Bilangan bulat dengan jangkauan untuk bilangan bertanda : -32768 s/d 32767 dan untuk yang tidak bertanda : 0 s/d 65535
MEDIUMINT	Ukuran 3 byte. Bilangan bulat dengan jangkauan untuk bilangan bertanda : -8388608 s/ d 8388607 dan untuk yang tidak bertanda : 0 s/d 16777215
INT	Ukuran 4 byte. Bilangan bulat dengan jangkauan untuk bilangan bertanda : -2147483648 s/d 2147483647 dan untuk yang tidak bertanda : 0 s/d 4294967295
INTEGER	Ukuran 4 byte. Sinonim dari int
BIGINT	Ukuran 8 byte. Bilangan bulat terbesar dengan jangkauan untuk bilangan bertanda : -9223372036854775808 s/d 9223372036854775807 dan untuk yang tidak bertanda : 0 s/d 1844674473709551615
FLOAT	Ukuran 4 byte. Bilangan pecahan
DOUBLE	Ukuran 8 byte. Bilangan pecahan
DOUBLEPRECISION	Ukuran 8 byte. Bilangan pecahan
REAL	Ukuran 8 byte. Sinonim dari DOUBLE
DECIMAL (M,D)	Ukuran M byte. Bilangan pecahan, misalnya DECIMAL(5,2) dapat digunakan untuk menyimpan bilangan -99,99 s/d 99,99
NUMERIC (M,D)	Ukuran M byte. Sinonim dari DECIMAL, misalnya NUMERIC(5,2) dapat digunakan untuk menyimpan bilangan -99,99 s/d 99,99

Type Data untuk Bilangan (Number)

Type Data	Keterangan
DATETIME	Ukuran 8 byte. Kombinasi tanggal dan jam, dengan jangkauan dari '1000-01-01 00:00:00' s/d '9999-12-31 23:59:59'
DATE	Ukuran 3 Byte. Tanggal dengan jangkauan dari '1000-01-01' s/d '9999-12-31'
TIMESTAMP	Ukuran 4 byte. Kombinasi tanggal dan jam, dengan jangkauan dari '1970-01-01 00:00:00' s/d '2037'
TIME	Ukuran 3 byte. Waktu dengan jangkauan dari '839:59:59' s/d '838:59:59'
YEAR	Ukuran 1 byte. Data tahun antara 1901 s/d 2155

Type Data untuk Tanggal dan Jam

Type Data	Keterangan
CHAR	Mampu menangani data hingga 255 karakter. Tipe data CHAR mengharuskan untuk memasukkan data yang telah ditentukan oleh kita.
VARCHAR	Mampu menangani data hingga 255 karakter. Tipe data VARCHAR tidak mengharuskan untuk memasukkan data yang telah ditentukan oleh kita.
TINYBLOB, TINYTEXT	Ukuran 255 byte. Mampu menangani data sampai 2^8-1 data.
BLOB, TEXT	Ukuran 65535 byte. Tipe string yang mampu menangani data hingga $2^{16}-1$ (16M-1) data.
MEDIUMBLOB, MEDIUMTEXT	Ukuran 16777215 byte. Mampu menyimpan data hingga $2^{24}-1$ (16M-1) data.
LONGBLOB, LONGTEXT	Ukuran 4294967295 byte. Mampu menyimpan data hingga berukuran GIGA BYTE. Tipe data ini memiliki batas penyimpanan hingga $2^{32}-1$ (4G-1) data.
ENUM('nilai1','nilai2',..., 'nilaiN')	Ukuran 1 atau 2 byte. Tergantung jumlah nilai enumerasinya (maksimum 65535 nilai)
SET('nilai1','nilai2',..., 'nilaiN')	1,2,3,4 atau 8 byte, tergantung jumlah anggota himpunan (maksimum 64 anggota)

Type Data untuk Karakter dan Lain-lain

4.2 Menciptakan Database

Database adalah sebuah media utama yang harus dibuat dalam membangun sebuah basis data agar nantinya dapat kita letakkan beberapa tabel dengan field-fieldnya.

Perintah yang digunakan untuk menciptakan database pada MySQL dengan Syntax berikut :

```
CREATE DATABASE nama_database;
```

Contoh :

```
mysql> create database pendaftaran;
```

```
Query OK, 1 row affected (0.11 sec)
```

Pada contoh diatas, query OK menyatakan bahwa pembuatan database dengan nama pendaftaran berhasil dibuat, untuk melihat database yang ada pada MySQL dapat menggunakan Sintax berikut ;

```
SHOW DATABASES;
```

Contoh :

```
mysql> show databases;
+-----+
| Database |
+-----+
| information_schema |
| cdcol |
| mysql |
| pendaftaran |
| phpmyadmin |
| test |
| webauth |
+-----+
7 rows in set (0.14 sec)
```

4.3 Menghapus Database

Untuk menghapus Database yang telah dibuat dapat menggunakan query SQL berikut :

```
DROP DATABASE nama_database;
```

Drop berarti menghapus. Query SQL ini berfungsi untuk menghapus sebuah database, seperti contoh berikut :

```
mysql> drop database pendaftaran;
```

```
Query OK, 0 rows affected (0.02 sec)
```

4.4 Menciptakan Tabel

Tabel adalah obyek utama yang harus ada pada sebuah basis data karena di dalamnya semua data akan disimpan. Tabel terletak pada sebuah database, sehingga pembuatan tabel dilakukan setelah sebuah database telah dibuat. Dalam tabel terdapat baris dan kolom. Baris diistilahkan dengan recordset dan kolom dengan field.

Id	Nama	Alamat	Phone
1	Boy Trimoyo	Jl. Ujung berung	08156849511
2	Irfan Nurhudin	Kp. Panyileukan Cibiru	08122295434

Untuk membuat sebuah tabel atau lebih, database harus diaktifkan dulu karena tabel akan dimasukkan ke dalam database yang akan diaktifkan. Sintax untuk mengaktifkan Database adalah :

```
USE nama_database;
```

Contoh :

```
mysql> use pendaftaran;
```

```
Database changed
```

Setelah masuk ke dalam database anda dapat membuat sebuah tabel atau lebih. Untuk membuat tabel dapat menggunakan sintax dibawah ini :

```
CREATE TABLE nama_tabel ( field-1 type(length), field-2 type(length), field-3 type(length), ..... ....(.....));
```

Contoh :

```
mysql> create table data_diri (  
    -> no int(3),  
    -> nama varchar(35),  
    -> alamat varchar(60),  
    -> email varchar(40),  
    -> no_telepon varchar(15),  
    -> sex char(1));
```

Query OK, 0 rows affected (0.08 sec)

Pada contoh diatas, query OK menyatakan bahwa pembuatan tabel dengan nama data_diri berhasil dibuat, untuk melihat tabel yang ada pada database dapat menggunakan Sintax berikut ;

SHOW TABLES;

Contoh :

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

4.5 Melihat Struktur Tabel

Setelah tabel dibuat, anda dapat melihat tipe data dan panjang recordset dengan cara menampilkan struktur tabel. Perintah yang digunakan untuk menampilkan struktur tabel adalah :

DESC nama_tabel;

Atau

DESCRIBE nama_tabel;

Contoh :

```
mysql> desc data_diri;
+-----+-----+-----+-----+-----+-----+
| Field | Type   | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| no    | int(3) | YES  |     | NULL    |       |
| nama  | varchar(35) | YES  |     | NULL    |       |
| alamat | varchar(60) | YES  |     | NULL    |       |
| email | varchar(40) | YES  |     | NULL    |       |
| no_telepon | varchar(15) | YES  |     | NULL    |       |
| sex   | char(1) | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
6 rows in set (0.11 sec)
```

4.6 Menghapus Tabel

Untuk menghapus Tabel yang telah dibuat dapat menggunakan query SQL berikut :

```
DROP TABLE nama_tabel;
```

Drop berarti menghapus. Query SQL ini berfungsi untuk menghapus sebuah Tabel, seperti contoh berikut :

```
mysql> drop table data_diri;
```

```
Query OK, 0 rows affected (0.03 sec)
```

4.7 Membuat Kunci Primer (Primary Key)

Dalam membuat sebuah database, kita akan menemukan sebuah record yang data nya tidak boleh sama dengan record yang lain. Agar data tidak kembar maka harus membuat sebuah kolom yang di deklarasikan sebagai kunci primer (primary key), Primary key hanya diperbolehkan dibuat satu kunci. Syntax untuk menciptakan kunci primer (primary key) adalah :

```
CREATE TABLE nama_tabel ( field-1 type(length)PRIMARY KEY, field-2 type(length), ..... ..(.....));
```

Contoh :

```
mysql> create table data_diri (
-> no int(3)primary key,
-> nama varchar(35),
-> alamat varchar(60),
-> email varchar(40),
-> no_telepon varchar(15),
-> sex char(1));
Query OK, 0 rows affected (0.06 sec)

mysql> desc data_diri;
+-----+-----+-----+-----+-----+-----+
| Field      | Type          | Null | Key  | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| no         | int(3)        | NO   | PRI  |          |       |
| nama      | varchar(35)   | YES  |      | NULL    |       |
| alamat    | varchar(60)   | YES  |      | NULL    |       |
| email     | varchar(40)   | YES  |      | NULL    |       |
| no_telepon| varchar(15)   | YES  |      | NULL    |       |
| sex       | char(1)       | YES  |      | NULL    |       |
+-----+-----+-----+-----+-----+-----+
6 rows in set (0.03 sec)
```

4.8 Membuat Kolom Unik (Unique)

Kolom Unique adalah sebuah bentuk kolom yang tidak mengizinkan adanya data kembar. Apabila pada proses input terdapat data kembar maka proses tersebut akan digagalkan atau ditolak oleh database.

Syntax untuk menciptakan Kolom unik (Unique) adalah :

```
CREATE TABLE nama_tabel ( field-1 type(length), field-2 type(length), ..... ..(.....),UNIQUE (field-1,field-2));
```

Contoh :

```
mysql> Create table pribadi (
-> kd_pribadi CHAR(3),
-> panggilan char(4),
-> nama varchar(35),
-> email varchar(50),
-> sex char(1),
-> UNIQUE (kd_pribadi,panggilan));
```

Query OK, 0 rows affected (0.08 sec)

4.9 Manipulasi Tabel

Perubahan tabel yang telah dibuat akan selalu dilakukan mengingat perkembangan database, termasuk diantaranya menambahkan beberapa field pada tabel, mengganti nama field maupun tabel.

4.9.1 Mengganti nama tabel

Query SQL untuk merubah nama tabel dengan menggunakan **RENAME**, Sintax seperti berikut :

```
RENAME TABLE tabel_lama TO tabel_baru;
```

Contoh :

```
mysql> rename table pribadi to data_pribadi;
```

```
Query OK, 0 rows affected (0.02 sec)
```

4.9.2 Menambah Field pada Tabel

Menambah kolom dapat diartikan sebagai langkah untuk menyisipkan field baru pada sebuah tabel. Untuk melakukan penambahan Field maka **ALTER** spesifikasi yang digunakan adalah **ADD**. Sintax yang digunakan adalah :

```
ALTER TABLE nama_tabel ADD nama_field Type_data(length);
```

Contoh :

```
mysql> alter table data_diri add gol_darah char(1);
```

```
Query OK, 0 rows affected (0.14 sec)
```

```
Records: 0 Duplicates: 0 Warnings: 0
```

4.9.3 Menghapus Field pada Tabel

Pada pembuatan database pasti terdapat kesalahan seperti pada field tabel yang berlebihan dan lain-lain. Untuk melakukan Penghapusan Field maka ALTER spesifikasi yang digunakan adalah **DROP**. Sintax yang digunakan adalah :

ALTER TABLE nama_tabel **DROP** nama_field;

Contoh :

```
mysql> alter table data_diri drop gol_darah;
```

```
Query OK, 0 rows affected (0.05 sec)
```

```
Records: 0 Duplicates: 0 Warnings: 0
```

BAB 5

DML (DATA MANIPULATION LANGUAGE)

DML adalah sebuah metode Query yang dapat digunakan apabila DDL telah terjadi, sehingga fungsi dari Query DML ini untuk melakukan pemanipulasian database yang telah dibuat.

Buatlah Database Buku_tamu dan didalamnya terdapat tabel tb_tamu dengan Struktur tabel berikut :

Field	Type	Length	Keterangan
No	Int	3	Primary key
Nama	Varchar	35	
Alamat	Varchar	60	
Email	Varchar	40	
No_telp	Varchar	15	

5.1 Memasukkan Data pada tabel (INSERT)

Memasukkan data atau entry data, dalam semua program yang menggunakan query SQL sebagai standar permintaannya, digunakan perintah INSERT. Syarat untuk memasukkan data adalah telah terciptanya tabel pada sebuah database. Sintax yang digunakan adalah :

```
INSERT INTO nama_tabel VALUES ('isi_field1', 'isi_field2', 'isi_field3',....., 'isi_fieldN');
```

Contoh :

```
mysql> insert into tb_tamu values('1','Boi trimoyo','ujung  
berung','bo_i77@yahoo.com','085613548789');
```

Query OK, 1 row affected (0.05 sec)

Maka data telah masuk ke dalam tabel seperti berikut :

```
mysql> select * from tb_tamu;
+----+-----+-----+-----+-----+
| no | nama      | alamat    | email          | no_telp  |
+----+-----+-----+-----+-----+
| 1  | Boi trimoyo | ujung berung | bo_i77@yahoo.com | 085613548789 |
+----+-----+-----+-----+-----+
1 row in set (0.01 sec)
```

5.2 Memperbarui Isi Data (UPDATE)

Memperbarui isi data atau update data adalah sebuah proses meremajakan data lama menjadi data yang lebih baru. Namun tidak semua data dalam database yang perlu diremajakan, melainkan sebagian data yang dianggap perlu untuk diremajakan. Query SQL yang digunakan adalah UPDATE yang di ketikkan seperti berikut :

UPDATE nama_tabel **SET**

field_1 = 'data_baru',

field_2 = 'data_baru',

..... ,

Field_N = 'data_baru';

Contoh :

```
mysql> update tb_tamu set
```

```
-> nama='irfan nurhudin' where nama="Boi trimoyo";
```

```
Query OK, 1 row affected (0.08 sec)
```

```
Rows matched: 1 Changed: 1 Warnings: 0
```

Maka hasilnya akan berubah seperti berikut :

```
mysql> select * from tb_tamu;
+----+-----+-----+-----+-----+
| no | nama      | alamat    | email          | no_telp |
+----+-----+-----+-----+-----+
| 1  | irfan nurhudin | ujung berung | bo_i77@yahoo.com | 085613548789 |
+----+-----+-----+-----+-----+
1 row in set (0.00 sec)
```

Data yang asalnya bernama Boi trimoyo berubah menjadi irfan nurhudin yang dihasilkan dari query SQL UPDATE.

5.3 Menghapus Data (DELETE)

Untuk menghapus data, MySQL memiliki query bernama DELETE. Penggunaannya diikuti dengan nama data yang akan dihapus. Berikut Sintax untuk menghapus semua data yang terdapat pada tabel :

DELETE FROM nama_tabel;

Sedangkan berikut sintax untuk menghapus data yang diinginkan dari sebuah tabel :

DELETE FROM nama_tabel **WHERE** kondisi;

Contoh :

Isikan data pada tabel tb_tamu seperti dibawah ini :

```
+----+-----+-----+-----+-----+
| no | nama      | alamat    | email          | no_telp |
+----+-----+-----+-----+-----+
| 1  | irfan nurhudin | ujung berung | bo_i77@yahoo.com | 085613548789 |
| 2  | Boi trimoyo    | cibiru    | bo_i77@yahoo.com | 082246864846 |
| 3  | Muswanto      | kopo     | muswanto@yahoo.com | 0229166478 |
| 4  | mohammad ridwan | bale endah | mory_89@yahoo.com | 02270598723 |
+----+-----+-----+-----+-----+
4 rows in set (0.00 sec)
```

Untuk menghapus data nomor 4 yang terdapat pada tabel tb_tamu maka gunakan query seperti dibawah ini :

```
mysql> delete from tb_tamu where no='4';
```

Query OK, 1 row affected (0.03 sec)

Maka hasilnya akan seperti dibawah ini bahwa data nomor 4 yang bernama ridwan telah dihapus menggunakan query DELETE :

```
mysql> select * from tb_tamu;
+----+-----+-----+-----+-----+
| no | nama      | alamat  | email                | no_telp |
+----+-----+-----+-----+-----+
| 1  | irfan nurhudin | ujung berung | bo_i77@yahoo.com | 085613548789 |
| 2  | Boi trimoyo   | cibiru   | bo_i77@yahoo.com | 082246864846 |
| 3  | Muswanto     | kopo    | muswanto@yahoo.com | 0229166478  |
+----+-----+-----+-----+-----+
3 rows in set (0.00 sec)
```

BAB 6

SELEKSI DATA

Menampilkan data adalah hal yang sangat penting karena kita harus melihat dan menyeleksi suatu data dalam table maupun antar table. Untuk Melihat data atau *Selection*, Query yang digunakan adalah **SELECT** yang diikuti beberapa pernyataan khusus berkenaan dengan tabel yang diseleksi.

6.1 Menampilkan Data Dari sebuah Tabel

Untuk menampilkan dari sebuah tabel dapat menggunakan Sintax berikut :

```
SELECT (Field1, field2, ....., FieldN) FROM nama_tabel;
```

Query diatas mengartikan bahwa data yang akan ditampilkan didalam tabel hanya filed – filed tertentu.

Atau

```
SELECT * FROM nama_tabel;
```

Query diatas mengartikan bahwa data dari seluruh Field yang terdapat dalam tabel akan ditampilkan.

Contoh :

```
mysql> select * from th_tamu;
+----+-----+-----+-----+-----+
| no | nama      | alamat    | email          | no_telp  |
+----+-----+-----+-----+-----+
| 1  | irfan nurhudin | ujung berung | bo_i77@yahoo.com | 085613548789 |
| 2  | Boi trimoyo   | cibiru    | bo_i77@yahoo.com | 082246864846 |
| 3  | Muswanto     | kopo     | muswanto@yahoo.com | 0229166478  |
+----+-----+-----+-----+-----+
3 rows in set (0.03 sec)
```

Atau

```
mysql> select no,nama,alamat from th_tamu;
+----+-----+-----+
| no | nama      | alamat    |
+----+-----+-----+
| 1  | irfan nurhudin | ujung berung |
| 2  | Boi trimoyo   | cibiru    |
| 3  | Muswanto     | kopo     |
+----+-----+-----+
3 rows in set (2.41 sec)
```

6.2 Menampilkan Data dengan Perintah WHERE

WHERE yang artinya dimana, untuk menampilkan data menggunakan perintah where (dimana) dapat menggunakan perintah berikut :

```
SELECT * FROM nama_tabel WHERE kondisi
```

Contoh :

Data sebelumnya yang ada pada tabel tb_tamu seperti berikut :

```
mysql> select * from tb_tamu;
+----+-----+-----+-----+-----+
| no | nama      | alamat    | email          | no_telp |
+----+-----+-----+-----+-----+
| 1  | irfan nurhudin | ujung berung | bo_i77@yahoo.com | 085613548789 |
| 2  | Boi trimoyo   | cibiru    | bo_i77@yahoo.com | 082246864846 |
| 3  | Muswanto     | kopo      | muswanto@yahoo.com | 0229166478 |
| 4  | aji fauziyaman | ujung berung | aji@yahoo.com    | 085222278892 |
+----+-----+-----+-----+-----+
4 rows in set (0.02 sec)
```

Maka akan menampilkan data menggunakan perintah where :

```
SELECT * FROM tb_tamu WHERE alamat='ujung berung';
```

```
mysql> select * from tb_tamu where alamat='ujung berung';
+----+-----+-----+-----+-----+
| no | nama      | alamat    | email          | no_telp |
+----+-----+-----+-----+-----+
| 1  | irfan nurhudin | ujung berung | bo_i77@yahoo.com | 085613548789 |
| 4  | aji fauziyaman | ujung berung | aji@yahoo.com    | 085222278892 |
+----+-----+-----+-----+-----+
2 rows in set (0.02 sec)
```

Contoh diatas mengartikan bahwa sintax meminta untuk menampilkan semua data yang ada pada tabel tb_tamu yang dimana akan ditampilkan dari field alamat yang isi data dari field alamat hanya ujung berung. Maka yang keluar adalah data ang filed alamatnya hanya ujung berung.

6.3 Menampilkan Data dengan BETWEEN

Between artinya diantara, between berfungsi untuk menampilkan data yang tertentu misalnya diantara 2000 dan 5000. Untuk menampilkan data dengan between dapat menggunakan sintax berikut :

```
SELECT * FROM nama_tabel WHERE kondisi BETWEEN nilai_1 AND nilai_2;
```

Buatlah sebuah DATABASE dengan nama db_stock kemudian buatlah tabel didalamnya dengan nama tabel t_barang kemudian isi data seperti berikut :

```
mysql> SELECT * from t_barang;
+-----+-----+-----+-----+-----+
| kode_brg | nama_barang | satuan | banyak | harga_brg |
+-----+-----+-----+-----+-----+
| AP001 | MICROSOFT WORD 2003 | BUAH | 20 | 35000 |
| AP002 | MICROSOFT EXCEL 2003 | BUAH | 15 | 37500 |
| AP003 | MICROSOFT POWER POINT 2003 | BUAH | 5 | 40000 |
| DB001 | MYSQL SERUER | BUAH | 5 | 45000 |
| DB002 | MICROSOFT SQL SERVER | BUAH | 10 | 55000 |
| DB003 | ORACLE 9i | BUAH | 10 | 65000 |
+-----+-----+-----+-----+-----+
6 rows in set (0.00 sec)
```

Contoh :

```
SELECT * FROM t_barang WHERE harga_brg BETWEEN 37500 AND 45000
```

```
mysql> SELECT * from t_barang where harga_brg between 37500 and 45000;
+-----+-----+-----+-----+-----+
| kode_brg | nama_barang | satuan | banyak | harga_brg |
+-----+-----+-----+-----+-----+
| AP002 | MICROSOFT EXCEL 2003 | BUAH | 15 | 37500 |
| AP003 | MICROSOFT POWER POINT 2003 | BUAH | 5 | 40000 |
| DB001 | MYSQL SERVER | BUAH | 5 | 45000 |
+-----+-----+-----+-----+-----+
3 rows in set (0.02 sec)
```

Contoh diatas menunjukkan bahwa semua data ditunjukkan dari tabel t_barang dimana yang ditampilkan dari field harga_brg diantara 37500 dan 45000. Maka data yang tampil hanya data yang bernilai 37500 sampai 45000.

6.4 Menampilkan Data dengan Perintah LIKE

Perintah Like kadang dibutuhkan dalam pembuatan database yaitu dalam menampilkan data tertentu yang hanya berkaitan dengan kata-kata yang diinginkan. Query yang digunakan adalah :

```
SELECT * FROM nama_tabel WHERE Kondisi LIKE '%nama_kaitan%';
```

```
mysql> SELECT * from t_barang where nama_barang like '%MICROSOFT%';
+-----+-----+-----+-----+-----+
| kode_brg | nama_barang | satuan | banyak | harga_brg |
+-----+-----+-----+-----+-----+
| AP001 | MICROSOFT WORD 2003 | BUAH | 20 | 35000 |
| AP002 | MICROSOFT EXCEL 2003 | BUAH | 15 | 37500 |
| AP003 | MICROSOFT POWER POINT 2003 | BUAH | 5 | 40000 |
| DB002 | MICROSOFT SQL SERVER | BUAH | 10 | 55000 |
+-----+-----+-----+-----+-----+
4 rows in set (0.00 sec)
```

Maka data yang ditampilkan hanya nama barang yang berkaitan dengan kata MICROSOFT.

6.5 Menampilkan Data dengan Pengurutan Sorting (ORDER BY)

Fungsi ini digunakan untuk melakukan pengurutan data, sehingga data dari sebuah atau beberapa tabel dapat tampil berurutan sesuai keinginan. Pengurutan data terbagi menjadi dua :

- *ASC* (pengurutan dengan Ascending)
- *DESC* (pengurutan dengan Descending)

Sintax yang digunakan adalah :

```
SELECT * FROM nama_tabel ORDER BY kolom Type
```

Contoh :

```
Select * from t_barang order by nama_barang asc;
```

```
mysql> Select * from t_barang order by nama_barang asc;
+----+-----+-----+-----+-----+
| kode_brg | nama_barang          | satuan | banyak | harga_brg |
+----+-----+-----+-----+-----+
| AP002    | MICROSOFT EXCEL 2003 | BUAH   | 15     | 37500     |
| AP003    | MICROSOFT POWER POINT 2003 | BUAH   | 5      | 40000     |
| DB002    | MICROSOFT SQL SERUER | BUAH   | 10     | 55000     |
| AP001    | MICROSOFT WORD 2003  | BUAH   | 20     | 35000     |
| DB001    | MYSQL SERUER        | BUAH   | 5      | 45000     |
| DB003    | ORACLE 9i           | BUAH   | 10     | 65000     |
+----+-----+-----+-----+-----+
6 rows in set (0.02 sec)
```

Dan

```
Select * from t_barang order by nama_barang desc;
```

```
mysql> Select * from t_barang order by nama_barang desc;
+----+-----+-----+-----+-----+
| kode_brg | nama_barang          | satuan | banyak | harga_brg |
+----+-----+-----+-----+-----+
| DB003    | ORACLE 9i           | BUAH   | 10     | 65000     |
| DB001    | MYSQL SERUER        | BUAH   | 5      | 45000     |
| AP001    | MICROSOFT WORD 2003  | BUAH   | 20     | 35000     |
| DB002    | MICROSOFT SQL SERUER | BUAH   | 10     | 55000     |
| AP003    | MICROSOFT POWER POINT 2003 | BUAH   | 5      | 40000     |
| AP002    | MICROSOFT EXCEL 2003 | BUAH   | 15     | 37500     |
+----+-----+-----+-----+-----+
6 rows in set (0.00 sec)
```

6.6 Menampilkan Data dengan Pengelompokan data (GROUP BY)

Group By adalah fungsi untuk mengelompokkan data dalam sebuah kolom yang ditunjuk. Fungsi ini akan menghasilkan kelompok data dengan menghilangkan data yang sama dalam satu tabel. Maka apabila dalam satu kolom terdapat beberapa data yang sama maka data yang akan ditampilkan hanya salah satu. Sintax yang digunakan seperti berikut :

```
SELECT * FROM nama_tabel GROUP BY nama_kolom;
```

Contoh :

```
mysql> Select * from t_barang group by nama_barang;
+-----+-----+-----+-----+-----+
| kode_brg | nama_barang          | satuan | banyak | harga_brg |
+-----+-----+-----+-----+-----+
| AP002   | MICROSOFT EXCEL 2003 | BUAH   | 15     | 37500     |
| AP003   | MICROSOFT POWER POINT 2003 | BUAH   | 5      | 40000     |
| DB002   | MICROSOFT SQL SERUER | BUAH   | 10     | 55000     |
| AP001   | MICROSOFT WORD 2003  | BUAH   | 20     | 35000     |
| DB001   | MYSQL SERVER        | BUAH   | 5      | 45000     |
| DB003   | ORACLE 9i           | BUAH   | 10     | 65000     |
+-----+-----+-----+-----+-----+
6 rows in set (0.00 sec)
```

6.6 Menampilkan sesuai dengan Fungsi Statistic

6.6.1 Fungsi COUNT

Fungsi ini biasanya digunakan untuk melakukan pengecekan jumlah data dalam sebuah tabel yang isinya ratusan hingga ribuan, sehingga kita tidak dapat menghitungnya secara manual. MySQL memiliki perintah untuk mengatasinya yaitu dengan menggunakan COUNT(). Sintaxnya seperti berikut :

```
SELECT COUNT (*) FROM nama_tabel;
```

Contoh :

```
mysql> SELECT COUNT(*) FROM tb_tamu;
```

```
+-----+
```

```
| COUNT(*) |
```

```
+-----+
```

```
| 4 |
```

```
+-----+
```

```
1 row in set (0.00 sec)
```

6.6.2 Fungsi SUM

SUM berfungsi untuk mencari nilai total dalam suatu kolom pada sebuah tabel didalam database. Query pada MySQL adalah SUM(). Sintax yang digunakan seperti berikut :

```
SELECT SUM(nama_kolom) FROM nama_tabel;
```

Contoh :

```
mysql> select SUM(harga_brg) from t_barang;
```

```
+-----+
| SUM(harga_brg) |
+-----+
|    277500    |
+-----+
1 row in set (0.00 sec)
```

6.7 Menampilkan Data Sesuai dengan Fungsi String

6.7.1 Fungsi LEFT (x,n)

Berfungsi mengambil data berdasarkan string dari sejumlah n karakter dari string X dari kiri. Sintax yang digunakan seperti berikut :

```
SELECT * FROM nama_tabel WHERE LEFT(nama_kolom,jumlah_karakter)="karakter";
```

Contoh :

Data sebelumnya

```
mysql> Select * from t_barang group by nama_barang;
+-----+-----+-----+-----+-----+
| kode_brg | nama_barang | satuan | banyak | harga_brg |
+-----+-----+-----+-----+-----+
| AP002 | MICROSOFT EXCEL 2003 | BUAH | 15 | 37500 |
| AP003 | MICROSOFT POWER POINT 2003 | BUAH | 5 | 40000 |
| DB002 | MICROSOFT SQL SERVER | BUAH | 10 | 55000 |
| AP001 | MICROSOFT WORD 2003 | BUAH | 20 | 35000 |
| DB001 | MYSQL SERVER | BUAH | 5 | 45000 |
| DB003 | ORACLE 9i | BUAH | 10 | 65000 |
+-----+-----+-----+-----+-----+
6 rows in set (0.00 sec)
```

Dengan menggunakan fungsi LEFT akan menjadi berikut :

```
mysql> select * from t_barang where left(kode_brg,1)="D";
+-----+-----+-----+-----+-----+
| kode_brg | nama_barang      | satuan | banyak | harga_brg |
+-----+-----+-----+-----+-----+
| DB001    | MYSQL SERVER    | BUAH   | 5      | 45000     |
| DB002    | MICROSOFT SQL  SERVER | BUAH   | 10     | 55000     |
| DB003    | ORACLE 9i       | BUAH   | 10     | 65000     |
+-----+-----+-----+-----+-----+
3 rows in set (0.00 sec)
```

Maka yang ditampilkan adalah isi data yang kode barangnya hanya berawal dari huruf D.

6.7.2 Fungsi RIGHT (x,n)

Fungsi Right hamper sama dengan fungsi LEFT hanya Query RIGHT Berfungsi mengambil data berdasarkan string dari sejumlah n karakter dari string X dari Kanan.

Sintax yang digunakan seperti berikut :

SELECT * FROM nama_tabel WHERE RIGHT(nama_kolom,jumlah_karakter)="karakter";

Contoh :

```
mysql> select * from t_barang where right(kode_brg,1)="2";
+-----+-----+-----+-----+-----+
| kode_brg | nama_barang      | satuan | banyak | harga_brg |
+-----+-----+-----+-----+-----+
| AP002    | MICROSOFT EXCEL 2003 | BUAH   | 15     | 37500     |
| DB002    | MICROSOFT SQL  SERVER | BUAH   | 10     | 55000     |
+-----+-----+-----+-----+-----+
2 rows in set (0.14 sec)
```

Maka yang ditampilkan adalah isi data yang kode barangnya hanya berakhir dari nomor 2.

6.7.2 Fungsi MID (x,n,z)

Berfungsi mengambil data berdasarkan string dari sejumlah dari string X n karakter sejumlah n karakter dari posisi kiri. Sintax yang digunakan seperti berikut :

SELECT * FROM nama_tabel WHERE MID(nama_kolom,posisi_n,jumlah_karakter)="karakter";

Contoh :

```
mysql> select * from t_barang where mid(kode_brg,2,1)="P";
+-----+-----+-----+-----+-----+
| kode_brg | nama_barang      | satuan | banyak | harga_brg |
+-----+-----+-----+-----+-----+
| AP001    | MICROSOFT WORD 2003 | BUAH   | 20     | 35000     |
| AP002    | MICROSOFT EXCEL 2003 | BUAH   | 15     | 37500     |
| AP003    | MICROSOFT POWER POINT 2003 | BUAH   | 5      | 40000     |
+-----+-----+-----+-----+-----+
3 rows in set (0.02 sec)
```

Maka yang ditampilkan adalah isi data yang kode barangnya berasal Huruf P dari 2 huruf karakter sebelah kiripada kolom kode barang.

6.8 Menampilkan data dengan nilai tertinggi dalam sebuah tabel (MAX)

Untuk mencari nilai tertinggi pada suatu data didalam database. SQL menyediakan fungsi MAX. Query yang digunakan seperti berikut :

```
SELECT MAX(nama_kolom) FROM nama_tabel;
```

Contoh :

```
mysql> select max(harga_brg) from t_barang;
```

```
+-----+
| max(harga_brg) |
+-----+
| 65000          |
+-----+
```

```
1 row in set (0.02 sec)
```

6.9 Menampilkan data dengan nilai terendah dalam sebuah tabel (MIN)

Untuk mencari nilai terendah pada suatu data didalam database. SQL menyediakan fungsi MIN. Query yang digunakan seperti berikut :

```
SELECT MIN(nama_kolom) FROM nama_tabel;
```

```
Contoh : mysql> select min(harga_brg) from t_barang;
```

```
+-----+
| min(harga_brg) |
+-----+
| 35000          |
+-----+
```

```
1 row in set (0.00 sec)
```

BAB 7

SELEKSI ANTAR TABEL

Dalam sebuah database yang berelasional, kita dapat menampilkan data dari dua tabel atau tiga tabel yang berbeda. Akan tetapi, apabila beberapa tabel tersebut merupakan yang berelasi, kita harus menggunakan teknik seleksi relasi khusus. Pada bab ini sebaiknya kita menyediakan database dan tabel nya terlebih dahulu untuk merelasikan antar tabel.

Buatlah Database dengan nama database : dokter

Kemudian buatlah tabel dengan intetas berikut :

- **tb_spesialis**

Field	Tipe	Panjang	Kunci (Key)
Kd_spesialis	char	3	Primari key
spesialis	varchar	35	

- **tb_dokter**

Field	Tipe	Panjang	Kunci (Key)
Kd_dokter	char	3	Primary key
Nama_dokter	varchar	35	
Kd_spesialis	char	3	Foreign Key
telepon	varchar	15	
sex	char	1	

- **tb_jaga**

Field	Tipe	Panjang	Kunci (Key)
Kd_dokter	Char	3	Foreign Key
hari	Varchar	15	
Jam_mulai	Time		
Jam_selesai	time		

Kemudian isilah data pada setiap tabel seperti dibawah ini :

- **tb_spesialis**

Kd_spesialis	spesialis
UMM	Dokter Umum
DLM	Penyakit Dalam
ANK	Anak
KDG	Kandungan
BDH	Bedah
JTG	Jantung
THT	Telinga Hidung dan Tenggorokan
GIG	Gigi
SRF	Saraf
KLT	Kulit
MAT	Mata

- tb_dokter

Kd_dokter	Nama_dokter	Kd_spesialis	telepon	Sex
D01	Dr. Boi Trimoyo	UMM	08111111111	P
D02	Dr. Irfan nurhuddin	DLM	08222222222	P
D03	Dr. Muswanto	BDH	08333333333	P
D04	Dr. umar	JTG	08444444444	P
D05	Dr. Ibrahim	KDG	08555555555	P
D06	Dr. aji	SRF	08666666666	P
D07	Dr. ridwan	MAT	08777777777	P
D08	Dr. fajar	ANK	08888888888	P
D09	Dr. mory	UMM	08999999999	P
D10	Dr. serly	BDH	08000000000	W
D11	Dr. Chandra	UMM	08101010101	P
D12	Dr. bayhaqi	BDH	08202020202	P
D13	Dr.rina	ANK	08303030303	W
D14	Dr.agus	UMM	08404040404	P
D15	Dr. andin	KDG	08505050505	w
D16	Dr. labala	BDH	08606060606	P
D17	Dr. fauzi	BDH	08707070707	P
D18	Dr. neneng	KDG	08080808080	W

- tb_jaga

Kd_dokter	hari	Jam_mulai	Jam_selesai
D01	senin	08:15:00	12:15:00
D09	Senin	12:15:00	16:30:00
D11	Senin	17:30:00	20:00:00
D03	selasa	08:15:00	16:00:00
D08	Selasa	16:00:00	21:00:00
D04	Selasa	08:15:00	20:00:00
D05	selasa	08:15:00	14:00:00
D15	Selasa	14:00:00	20:00:00
D13	Rabu	08:15:00	12:00:00
D12	Rabu	08:15:00	14:00:00
D16	Rabu	14:00:00	21:00:00
D14	Rabu	08:15:00	14:00:00
D06	Kamis	08:15:00	14:00:00
D07	Jum'at	08:15:00	11:30:00
D10	Sabtu	08:15:00	14:00:00
D02	minggu	09:00:00	14:00:00