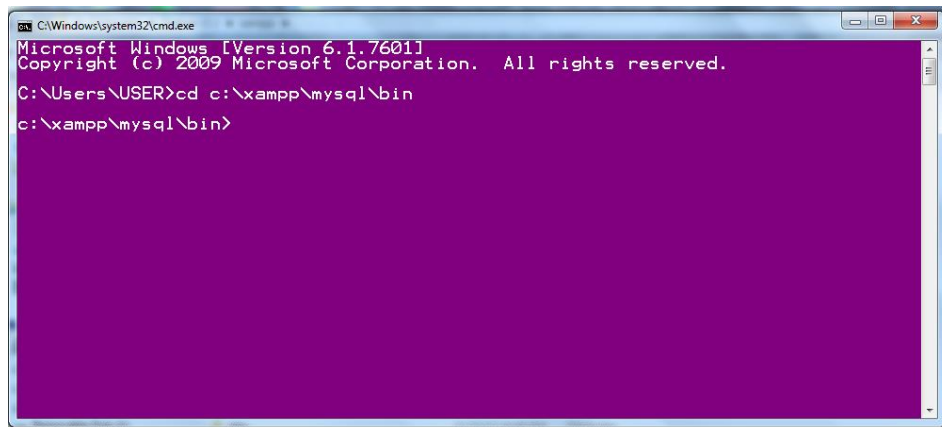


MODUL BASIS DATA – DASAR BASIS DATA 1

A. Mempersiapkan MySQL

1. Lakukan instalasi MySQL client menggunakan salah satu perangkat lunak
2. Instalasi Koneksi ke MySQL

Buka command prompt, masuk ke direktori \AppServ\MySQL\bin,



```
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Users\USER>cd c:\xampp\mysql\bin
c:\xampp\mysql\bin>
```

kemudian ketik:

mysql -u root -p (enter)

password : (ketikkan password) (enter)

3. Diskoneksi dari MySQL

Untuk diskoneksi, ketik:

QUIT atau \q

B. DDL (Data Definition Language)

DDL (Data Definition Language) merupakan sekumpulan set perintah yang bertujuan untuk mendefinisikan atribut – atribut database, tabel, atribut kolom (field), maupun batasan - batasan terhadap suatu atribut dan relasi/hubungan antar tabel. Yang termasuk dalam kelompok perintah DDL adalah : CREATE, ALTER, dan DROP.

- ✓ **CREATE** merupakan perintah DDL yang digunakan untuk membuat database maupun tabel. Nama database maupun tabel tidak boleh mengandung spasi (space). Nama database tidak boleh sama antar database.
- ✓ **ALTER** merupakan perintah DDL yang digunakan untuk mengubah nama/struktur tabel.
- ✓ **DROP** merupakan perintah DDL yang digunakan untuk menghapus database ataupun tabel.

a. Membuat Database

Karena server baru diinstalasi, maka belum terdapat basis data kecuali yang sidatnya internal untuk kebutuhan DBMS. Untuk melihat basis data *default* yang sudah ada, gunakan perintah **show databases**.

Basis data dapat dibuat dengan perintah **create database**.

```
CREATE TABLE namatabel (Field1, TypeData1,
Field2, TypeData2);
```

Misalkan akan dibuat basis data kepegawaian, perintahnya:

```
create database kepeg;
```

Maka daftar basis datanya akan berubah:

Untuk membuka atau menggunakan database yang baru saja dibuat, maka gunakan perintah **use**.

```
use kepeg;
```

b. Membuat Tabel

Untuk membuat tabel di dalam database, struktur tabel yang akan dibuat harus sudah diketahui terlebih dahulu. Adapun syntax yang digunakan untuk membuat tabel secara umum adalah sebagai berikut:

```
CREATE TABLE namatabel (Field1, TypeData1,
Field2, TypeData2);
```

Misalkan ingin dibuat sebuah **tabel divisi** dengan struktur sebagai berikut:

iddiv	nama

No.	Nama Field	Tipe	Panjang	Keterangan
1.	iddiv	karakter	3	Id Divisi, Primary Key
2.	nama	karakter	10	Nama Divisi

Sehingga script untuk membuat tabel divisi adalah sebagai berikut:

```
create table divisi
(iddiv char(3) primary key,
nama varchar(10)
);
```

Untuk melihat tabel divisi yang baru saja dibuat, maka gunakan script **show**.

```
show tables;
```

Sedangkan untuk melihat kembali struktur tabel tersebut, gunakan perintah **desc**.

```
desc divisi;
```

Dengan cara yang sama, tambahkan tabel kedua bernama **pegawai** dengan struktur sebagai berikut:

npp	nama	alamat

No.	Nama Field	Tipe	Panjang	Keterangan
1.	npp	karakter	4	Id pegawai, Primary Key
2.	nama	karakter	40	Nama pegawai
3.	alamat	karakter	50	Alamat pegawai

Adapun scriptnya:

```
create table pegawai
(npp char(6) primary key,
nama varchar(40),
alamat varchar(50)
);
```

c. Mengubah struktur tabel (Alter)

Pengubahan struktur tabel yang dapat dilakukan dengan perintah **alter table** meliputi penambahan *field*, pengurangan *field*, pengubahan tipe data dan ukuran *field*. Untuk menambahkan field pada tabel, script umumnya adalah sebagai berikut:

```
ALTER TABLE namatabel ADD fieldbaru
typedata(lebar);
```

Misalkan tabel pegawai membutuhkan penambahan *field* baru bernama **iddiv** dengan tipe string dan berukuran 3 karakter.

```
alter table pegawai add column iddiv char(3);
```

Untuk mengubah tipe data dan lebar kolom pada tabel dapat menggunakan script umum sebagai berikut:

```
ALTER TABLE namatabel MODIFY COLUMN field  
type(lebar);
```

Contoh :

```
ALTER TABLE pegawai MODIFY COLUMN  
Alamat varchar(62);
```

Untuk mengubah nama kolom pada tabel script umumnya adalah sebagai berikut:

```
ALTER TABLE namatabel CHANGE COLUMN  
namakolomlama namakolombaru  
typedatabaru(lebarbaru);
```

Contoh:

```
ALTER TABLE pegawai CHANGE COLUMN nama  
nama_pegawai CHAR(25);
```

Dan untuk menghapus suatu kolom pada tabel tertentu, script alter tabel yang dapat digunakan adalah:

```
ALTER TABLE namatabel DROP COLUMN namakolom;
```

Contoh:

```
ALTER TABLE pegawai DROP COLUMN phone;
```

d. Menghapus Tabel dan Database (Drop)

Untuk menghapus database maupun tabel pada database dapat menggunakan perintah **drop**. Misalkan kita ingin menghapus salah satu database, maka script umum yang dapat digunakan adalah:

```
DROP DATABASE namadatabase;
```

Contoh:

```
DROP DATABASE coba;
```

Sedangkan untuk menghapus tabel pada database, script yang dapat digunakan adalah:

```
DROP TABLE namaTabel;
```

Contoh:

```
DROP TABLE biaya;
```

C. DML (Data Manipulation Language)

DML (Data Manipulation Language) merupakan kelompok perintah yang berfungsi untuk memanipulasi data dalam database. Contohnya untuk memasukkan, pengambilan, pengubahan maupun penghapusan data. Perintah yang termasuk dalam DML adalah: INSERT, DELETE, UPDATE, dan SELECT.

a. Mengisi data table (**Insert**)

Perintah INSERT bertujuan untuk menambahkan record data pada suatu tabel. Terdapat beberapa cara untuk menambahkan record, yaitu: Cara 1: Menambahkan record dengan mengisi data pada setiap kolom:

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

Cara 2: menambahkan baris dengan hanya mengisi pada kolom tertentu:

```
INSERT INTO namatabel (field1, field2, field-n) VALUES (nilai1, nilai2, nilai-n);
```

Misalkan untuk mengisi table pegawai yang sudah dibuat, script yang dapat dituliskan adalah:

```
insert into pegawai values(  
    '1101', 'Alana', 'Jl. Bunga Matahari Kendari');
```

b. Melihat isi table (**Select**)

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

✓ Menampilkan data semua kolom dengan menggunakan asterisk (*):

```
SELECT * FROM namatabel;
```

Contoh: jika kita ingin menampilkan semua kolom pada table pegawai maka script yang dapat dituliskan adalah:

```
SELECT * FROM pegawai;
```

- ✓ Menampilkan data untuk field/kolom tertentu:

```
SELECT filed1, filed2, filed-n FROM namatabel;
```

Contoh: jika kita ingin melihat kolom npp dan nama saja dari table pegawai, maka script yang bias dituliskan adalah:

```
SELECT npp,nama FROM pegawai;
```

- ✓ Menampilkan data dengan kondisi tertentu menggunakan klausa WHERE:

```
SELECT * FROM namatabel WHERE kondisi;
```

Jika kita ingin membatasi record yang muncul atau untuk mencari record dengan criteria tertentu, dapat digunakan klausa **where**.

Contoh: jika kita ingin melihat data pegawai dengan npp='1102' , maka script yang bias digunakan adalah:

```
SELECT * FROM pegawai WHERE npp='1102';
```

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

c. Mengubah data pada table (**Update**)

Perintah UPDATE digunakan untuk mengubah isi data pada satu atau beberapa kolom pada suatu tabel. Syntax yang digunakan secara umum adalah sebagai berikut:

```
UPDATE namatabel SET field1=nilai1, field2=nilai2 [WHERE kondisi];
```

Update adalah perintah yang ahrus dilakukan dengan hati-hati, karena untuk kondisi ekstrem dapat mengubah keseluruhan isi tabel.

Contoh: Jika kita ingin mengubah nama seorang pegawai, maka scrio yang bias digunakan adalah:

```
UPDATE pegawai SET nama='Andini' where npp='1102';
```

d. Menghapus data pada table (**Delete**)

Perintah DELETE digunakan untuk menghapus satu baris, baris dengan kondisi tertentu maupun seluruh baris. Syntax yang digunakan:

```
DELETE FROM namatabel WHERE [kondisi];
```

Perintah dalam tanda [] bersifat pilihan/opsional untuk menghapus suatu baris dengan kondisi tertentu yang dipersyaratkan.

Contoh perintah untuk menghapus suatu baris dalam tabel dengan kondisi persyaratan tertentu :

```
DELETE FROM pegawai WHERE npp='1103';
```