

# Modul Praktikum Sistem Manajemen Basis Data

Disusun Oleh :

Moch. Kautsar Sophan, S.Kom., M.MT.

Fifin Ayu Mufarroha, S.Kom., M.Kom.

Doni Abdul Fatah



**Sistem Informasi  
Fakultas Teknik  
UNIVERSITAS TRUNOJOYO MADURA**

# MODUL I

## DATA DEFINITION LANGUAGE (DDL) DAN

## DATA MANIPULATION LANGUAGE (DML)

### Tujuan :

- Mampu memahami perintah-perintah untuk menjelaskan objek dari database dan mendefinisikan atribut-atribut database, table dan batasan-batasan terhadap suatu atribut serta hubungan antar table.
- Mampu memahami dan mengisi table dalam database dan memanipulasi data dalam database.

### A. Dasar Teori

#### I. Data Definition Language

DDL SQL memungkinkan dilakukannya spesifikasi tidak hanya pada himpunan relasi tetapi juga informasi untuk setiap relasi, yang meliputi :

- Skema setiap relasi
- Domain nilai setiap atribut relasi
- Konstrain integritas
- Himpunan indeks untuk setiap relasi
- Sekuriti dan otorisasi setiap relasi
- Struktur penyimpanan secara fisik untuk setiap relasi dalam disk.

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. DATABASE

Database dapat dibayangkan sebagai sebuah lemari arsip, dimana arsip yang disimpan didalam lemari tersebut akan disusun dan dikelompokkan berdasarkan

urutan tertentu seperti urutan abjad atau urutan kronologis. Hal ini dilakukan sebagai upaya untuk mempermudah pencarian arsip, karena arsip yang tersusun dengan rapi maka proses pencarian arsip dapat lebih cepat.

Syntax untuk membuat database sebagai berikut :

```
CREATE DATABASE namadatabase;
```

Sedangkan syntax tambahan untuk menampilkan daftar nama database yang terdapat dalam database server pada MySQL menggunakan perintah :

```
SHOW DATABASES;
```

Sebelum kita membuat suatu tabel yang digunakan untuk menyimpan data, terlebih dahulu harus memilih/mengaktifkan salah satu database sebagai database aktif yang akan digunakan untuk menyimpan beberapa tabel yang akan kita buat.

Untuk memilih/mengaktifkan salah satu database menggunakan syntax :

```
USE namadatabase;
```

Contohnya kita akan mengaktifkan database db\_universitas : USE db\_universitas;  
Selain perintah - perintah di atas, terdapat perintah yang berfungsi untuk menghapus database maupun tabel. Perintah tersebut adalah sebagai berikut :

```
DROP DATABASE namadatabase;
```

## **b. TABEL**

- **Membuat Tabel**

Dalam membuat sebuah tabel, nama tabel tidak boleh mengandung spasi (space). Ketika membuat tabel ada beberapa yang harus dideklarasikan dalam pembuatannya, yaitu antara lain meliputi :

- ✓ Nama tabel,
- ✓ Nama Field (Kolom),
- ✓ Type data dari field dan
- ✓ Panjang data.

Adapun syntax yang digunakan untuk membuat tabel secara umum adalah sebagai berikut:

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

Contoh berikut syntax untuk membuat Tabel mahasiswa:

```
CREATE TABLE mahasiswa (nim CHAR (20), nama_mhs CHAR (50), login CHAR(20),  
pass CHAR(20), umur INT, ipk real, PRIMARY KEY(nim));
```

- **Menampilkan Tabel**

Untuk menampilkan daftar nama tabel yang terdapat dalam database yang sedang aktif/digunakan menggunakan perintah :

```
SHOW TABLES;
```

- Menampilkan deskripsi atribut table

Untuk menampilkan deskripsi atribut – atribut yang terdapat pada suatu tabel dengan menggunakan perintah:

```
DESC namaTabel;
```

- Menghapus table

Untuk menghapus Tabel perintahnya sama dengan untuk menghapus database yaitu dengan menggunakan perintah DROP. Syntax yang digunakan adalah:

```
DROP TABLE namaTable;
```

Tabel yang akan dihapus harus sesuai dengan nama tabel. Misal kita akan menghapus tabel mahasiswa, maka syntax nya adalah: DROP TABLE mahasiswa;

- Mendefinisikan Null/Not Null

Null ataupun Not Null merupakan pernyataan yang digunakan untuk membuat kolom yang kita buat boleh kosong (Null) atau tidak boleh kosong (Not Null). Ketika pada kolom tabel tidak diset, maka secara default akan bernilai Null (boleh kosong).

Untuk mendefinisikannya maka perintah yang digunakan adalah:

```
CREATE TABLE mahasiswa (nim CHAR (20) NOT NULL, nama_mhs CHAR (50) NOT NULL, login CHAR(20) NOT NULL, pass CHAR(20) NOT NULL, umur INT, ipk real, PRIMARY KEY(nim));
```

- Mendefinisikan Nilai Default

Nilai default merupakan nilai yang diberikan secara otomatis oleh sistem untuk suatu kolom ketika terjadi penambahan baris baru, sementara nilai pada kolom tersebut tidak diisi oleh pengguna. Contohnya adalah :

```
CREATE TABLE mahasiswa (nim CHAR (20), nama_mhs CHAR (50), login CHAR(20), pass CHAR(20), umur INT DEFAULT 0, ipk real, PRIMARY KEY(nim));
```

- Mendefinisikan PRIMARY KEY pada Tabel

Suatu keharusan dalam suatu tabel adalah harus memiliki satu kolom yang dijadikan sebagai perwakilan dari tabel tersebut. Pembuatan perwakilan tabel ini berfungsi untuk melakukan hubungan/relasional dengan tabel lain. Bentuk perwakilan ini dalam database disebut sebagai PRIMARY KEY yang aturan pembuatannya adalah sebagai berikut:

- Satu tabel hanya diperbolehkan memiliki satu kolom kunci.
- Nama kolom kunci tidak digunakan pada kolom lain dalam satu tabel
- Nama kolom kunci tidak boleh sama dengan kolom kunci yang ada pada tabel lain
- Bentuk kolom kunci harus diset NOT NULL.

Terdapat tiga cara untuk mendefinisikan primary key. Berikut ini syntax yang digunakan:

```
CREATE TABLE mahasiswa (nim CHAR (20), nama_mhs CHAR (50), login CHAR(20), pass CHAR(20), umur INT, ipk real, PRIMARY KEY(nim));
```

Atau

```
CREATE TABLE mahasiswa (nim CHAR (20) NOT NULL PRIMARY KEY, nama_mhs CHAR (50), login CHAR(20), pass CHAR(20), umur INT, ipk real);
```

Atau

```
ALTER TABLE mahasiswa ADD CONSTRAINT namaconstraint PRIMARY KEY(namakolom);
```

- Menghapus PRIMARY KEY pada Tabel

Cara 1: Jika primary key dibuat menggunakan alter table:

```
ALTER TABLE namatabel DROP CONSTRAINT namaconstraint;
```

Cara 2: jika primary key dibuat melalui create table:

```
ALTER TABLE namatabel DROP PRIMARY KEY;
```

- Menambah kolom baru pada tabel

Pada saat kita membuat tabel terkadang kita ingin menambahkan kolom lagi pada tabel yang sudah kita buat. Dalam database, hal ini dapat dilakukan dengan menggunakan perintah sebagai berikut:

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

Namatabel merupakan nama tabel yang akan ditambahkan kolomnya.

Field baru merupakan nama kolom yang akan ditambahkan, typedata(lebar) merupakan type data dan lebar data yang akan ditambahkan.

Misal kita akan menambahkan kolom telepon pada tabel mahasiswa setelah kolom umur:

```
ALTER TABLE mahasiswa ADD COLUMN telepon CHAR(15) AFTER umur;
```

- Mengubah Tipe Data atau Lebar Kolom pada Tabel

Perintah yang digunakan adalah:

```
ALTER TABLE namatabel MODIFY COLUMN field
```

Contoh :

ALTER TABLE mahasiswa MODIFY COLUMN CHANGE COLUMN telepon(12);

- Mengubah Nama Kolom (Field)

Perintah yang digunakan adalah:

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

Contoh :

ALTER TABLE mahasiswa CHANGE COLUMN telepon phone CHAR(25);

- Menghapus Kolom pada Tabel

Perintah yang digunakan:

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

Contoh :

ALTER TABLE mahasiswa DROP COLUMN phone;

- Membuat dan Menghapus Index

Index berfungsi untuk mempercepat proses pencarian data dalam suatu tabel. Dengan adanya index pada suatu field tabel menyebabkan proses pencarian otomatis akan dilakukan terlebih dahulu ke dalam index, apabila ditemukan baru akan diambilkan data sesungguhnya dari tabel, apabila tidak ditemukan dalam index, sudah dapat dipastikan bahwa data tersebut tidak ada dalam tabel. Terdapat perintah untuk membuat dan menghapus index, tapi tidak ada perintah untuk merubah index. Perhatikan contoh berikut:

```
CREATE INDEX IDXNIM ON mahasiswa(nim);
```

Atau

```
ALTER TABLE mahasiswa ADD INDEX IDXNIM(nim);
```

Sedangkan untuk menghapus :

```
DROP INDEX IDXNIM ON mahasiswa(nim);
```

Atau

```
ALTER TABLE mahasiswa DROP INDEX IDXNIM(nim);
```



## II. Data Manipulation Language

Merupakan bentuk bahasa basis data yang berguna untuk melakukan manipulasi data dan pengambilan data pada suatu basis data. Manipulasi data dapat berupa :

- a. penyisipan/penambahan data baru ke suatu basis data
- b. penghapusan data dari suatu basis data
- c. pengubahan data di suatu basis data

Pada level fisik, kita harus mendefinisikan algoritma yang memungkinkan pengaksesan yang efisien terhadap data. Pada level yang lebih tinggi, yang dipentingkan bukan hanya efisiensi akses, tetapi juga efisiensi interaksi manusia (pemakai) dengan sistem (kemudahan permintaan akses). *Data Manipulation Language* (DML) merupakan bahasa yang bertujuan memudahkan pemakai untuk mengakses data sebagaimana direpresentasikan oleh model data. Ada 2 jenis DML, yaitu :

- Prosedural, yang mensyaratkan agar pemakai menentukan, data apa yang diinginkan serta bagaimana cara mendapatkannya.
- Nonprosedural, yang membuat pemakai dapat menentukan data apa yang diinginkan tanpa menyebutkan bagaimana cara mendapatkannya.

Perintah yang termasuk dalam DML adalah: INSERT, DELETE, UPDATE, dan SELECT

### 1) 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-
```

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

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

Ket : Jika data bertipe string, date, atau time (contoh : didi, basis data, 1984-03-18) maka pemberian nilainya diapit menggunakan tanda petik tunggal ('Didi') atau petik ganda ("Basis Data"). Jika data bertipe numerik (29, 4) maka pemberian nilainya tidak diapit tanda petik tunggal maupun ganda.

### 2) 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 mahasiswa WHERE nim 13120070;

### 3) 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];
```

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

### 4) 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;
```

Menampilkan data untuk field/kolom tertentu:

```
SELECT field1, field2, field-n FROM namatabel;
```

Menampilkan data dengan kondisi tertentu menggunakan klausa WHERE:

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

## B. Tugas Praktikum

1. Buatlah database tentang sebuah toko swalayan ingin mengembangkan database untuk mencatat transaksi yang terjadi.

✓ Transaksi berupa :

- POS / Penjualan : ke pelanggan
- Pengembalian barang / retur
- Stok
- Pembelian barang dari supplier
- Koreksi stok

✓ Keuangan berupa :

- Penjualan, masuk ke akun penjualan
- Pembelian dari supplier, masuk ke akun : hutang (pembayaran ke supplier, mengurangi akun hutang)

✓ Laporan berupa :

- Penjualan
- Stok

2. Isi data pada tabel masing – masing 10 data



3. Tampilkan keseluruhan data pada setiap tabel.
4. Lakukan perubahan pada salah satu nama tabel
5. Hapus database tersebut.

#### **C. Referensi**

- R. Ramakrisnan & J. Gehrke. (2004). Sistem Manajemen Database (Edisi ketiga). Andi and McGraw-Hill Education.
- Wahana Komputer. (2010). Panduan Belajar MySQL Database Server. MediaKita
- Kadir, Abdul. (2010). Mudah Mempelajari Database MySQL. Yogyakarta: ANDI
- R. H. Sianipar. (2016). Pemrograman Database Menggunakan Mysql. Andi Offset.