

Mohammad Mahmudi, S.Kom., M.Pd

SMKN 4 Malang Jl. Tanimbar No. 22 Kota Malang

Telp. 0341-353798

# MODUL BASISDATA

## #3

*Kelompok perintah dalam basis data*

# MODUL 3

## KELOMPOK PERINTAH DALAM BASIS DATA

### Daftar Isi

A. Pendahuluan	1
B. Pembelajaran	3
C. Evaluasi	20
D. Kunci Jawaban	21
E. Daftar Pustaka	24
F. Penutup	24
Company Information	25

# MODUL 3

## KELOMPOK PERINTAH DALAM BASIS DATA

### A. Pendahuluan

Dalam modul ini digunakan bahasa query yang digunakan dalam mengolah basis data yakni SQL. SQL adalah singkatan dari Structured Query Language. Sedangkan **pengertian SQL adalah** suatu bahasa (language) yang digunakan untuk mengakses data di dalam sebuah database relasional. SQL sering juga disebut dengan istilah query, dan bahasa SQL secara praktiknya digunakan sebagai bahasa standar untuk manajemen database relasional. Hingga saat ini hampir seluruh server database atau software database mengenal dan mengerti bahasa SQL.

Sejarah SQL awal mula lahirnya bahasa SQL yaitu pada bulan juni 1970, dimana saat Jhonny Oracle yang merupakan seorang peneliti dari perusahaan IBM memiliki gagasan pembuatan basis data relasional, ide tersebut dituangkan dalam sebuah artikel. Di dalam artikel tersebut juga dibahas mengenai kemungkinan membuat sebuah bahasa standar untuk mengakses data dalam database tersebut. Bahasa standar tersebut diberinama sequel (Structured English Query Language). Setelah kemunculan artikel tersebut lalu IBM memutuskan untuk mengembangkan pembuatan bahasa sequel. Namun penamaan sequel dalam bahasa standar tersebut bermasalah dengan hukum sehingga diubahlah menjadi SQL.

Jenis perintah SQL dalam penggunaan SQL terdapat beberapa perintah yang berguna untuk mengakses dan memanajemen data yang terdapat dalam database. Jenis perintah SQL secara umum dibagi kepada empat sub perintah, yaitu DDL (*Data Definition Language*), DML (*Data Manipulation Language*), DCL (*Data Control Language*) dan TCL (*Transaction Control Language*). Keempat sub perintah tersebut sangat perlu untuk dipahami bagi anda yang ingin menguasai bahasa SQL dan mahir dalam pembuatan database.

#### 1. STANDART KOMPTENESI

- 3.4 Memahami kelompok perintah yang termasuk dalam DDL, DML dan DCL dalam basis data
- 4.4 Membuat kelompok perintah yang termasuk dalam DDL, DML dan DCL dalam basis data
- 3.5 Memahami tipe data pada basis data
- 4.5 Mempresentasikan tipe data pada basis data
- 3.6 Menerapkan bahasa SQL dasar untuk mengelola tabel dalam basis data
- 4.6 Membuat kode SQL dasar untuk mengelola tabel dalam basis data
- 3.8 Menerapkan bahasa SQL dasar untuk memanipulasi tabel dalam basis data
- 4.8 Membuat kode SQL dasar untuk memanipulasi tabel dalam basis data
- 3.10 Menerapkan penggunaan SQL untuk mengakses *multi table*
- 4.10 Membuat perintah SQL untuk mengakses *multi table*

# MODUL 3

## KELOMPOK PERINTAH DALAM BASIS DATA

### 2. DESKRIPSI

Modul ini merupakan modul pembelajaran mata pelajaran Basis Data untuk siswa SMK kelas XI semester 3. Modul pembelajaran ini dibuat dalam rangka untuk mempermudah proses pembelajaran. Modul 3 ini berisi materi Kelompok perintah basis data yakni perintah DDL, DML dan DCL.

### 3. WAKTU

Alokasi pembelajaran untuk modul 3 Kelompok perintah basis data ini dapat digunakan dalam waktu pembelajaran selama lima kali tatap muka.

### 4. PETUNJUK PENGGUNAAN MODUL

#### **Sebelum Pembelajaran**

1. Sebelum masuk pada materi, disajikan pendahuluan sebagai pengantar menuju materi utama.
2. Disajikan kompetensi dasar dan alokasi waktu sebagai pedoman bagi pengguna modul untuk mencapai tujuan pembelajaran.

#### **Selama Pembelajaran**

1. Mempelajari dan memahami materi pada modul.
2. Mempelajari dan mencatat materi yang dianggap penting.
3. Mengerjakan tugas yang terdapat pada bagian evaluasi.
4. Mengerjakan tes untuk mengukur kemampuan dalam memahami modul.

#### **Setelah Pembelajaran**

1. Mengevaluasi jawaban dengan kunci jawaban.
2. Mengetahui hasil evaluasi (sudah memenuhi kriteria ketuntasan atau belum).
3. Memutuskan untuk meneruskan belajar pada materi selanjutnya atau tetap pada materi yang sama.

# MODUL 3

## KELOMPOK PERINTAH DALAM BASIS DATA

### B. Pembelajaran

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:

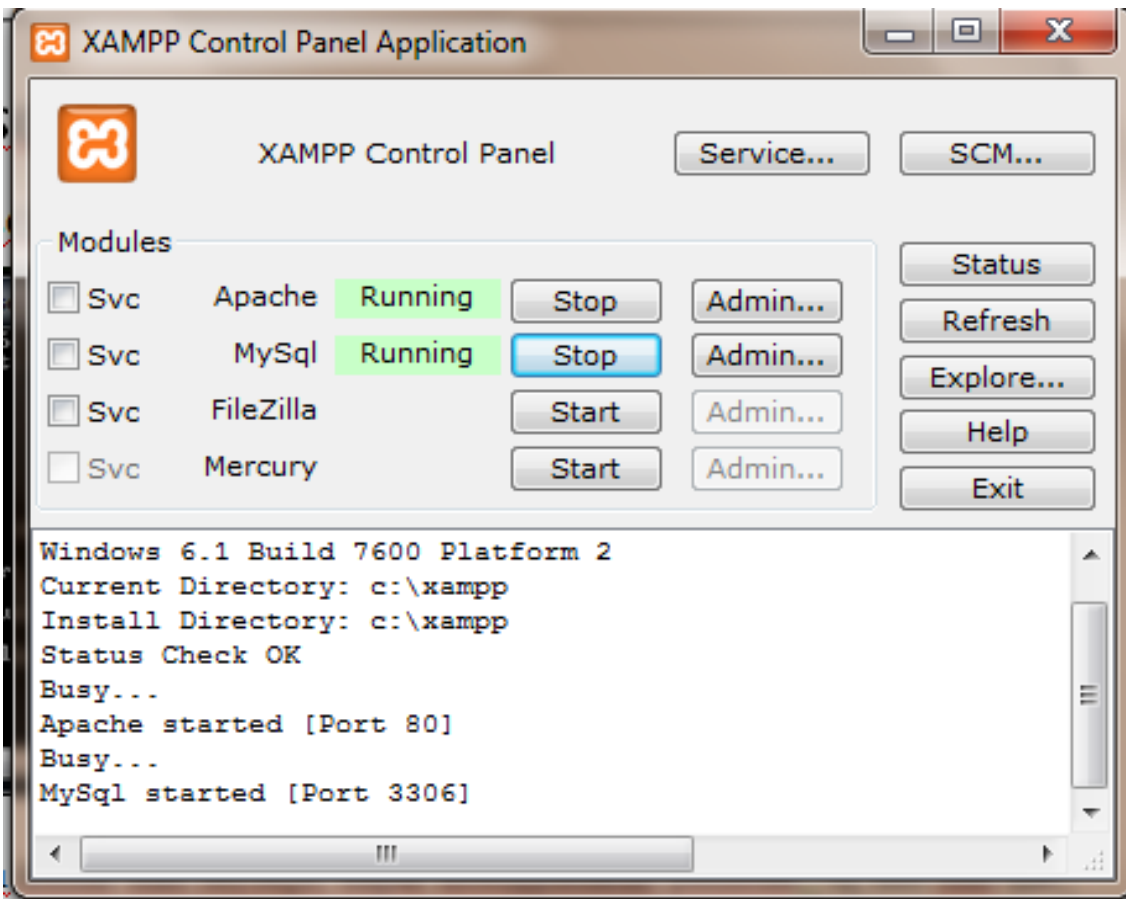
- Setiap perintah harus diakhiri dengan tanda titik koma , kecuali untuk perintah tertentu, misal : quit
- Setiap perintah akan disimpan dalam buffer (memori sementara) untuk menyimpan histori perintah-perintah yang pernah diberikan.
- Perintah dapat berupa perintah SQL atau perintah khusus MySQL.
-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

# MODUL 3

## KELOMPOK PERINTAH DALAM BASIS DATA

### Start dan Stop MySQL

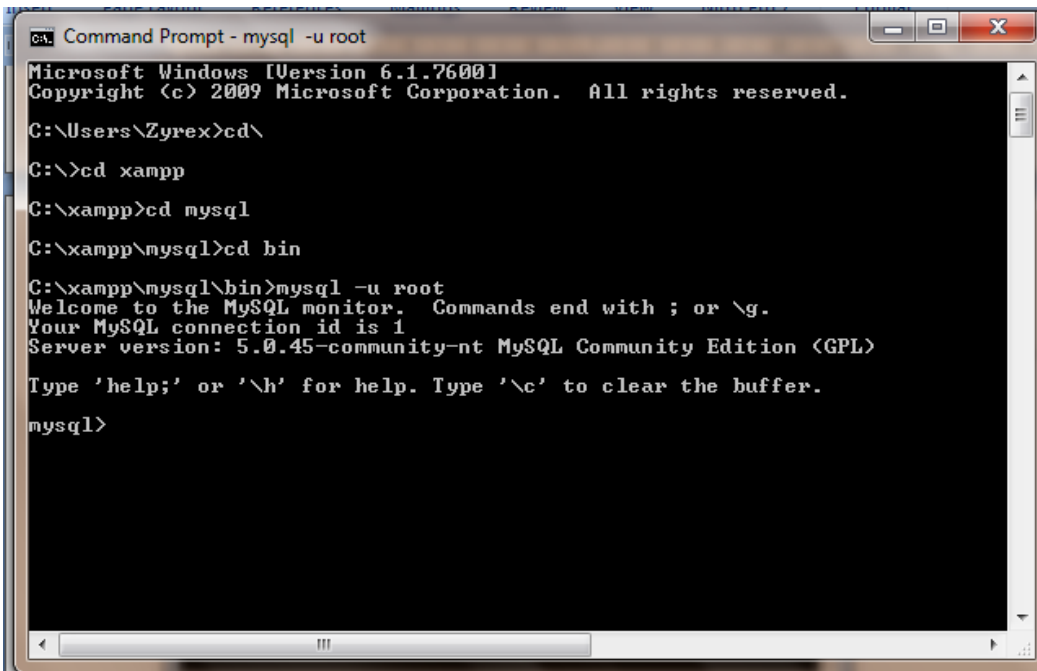
Berikut cara memulai MySQL dengan menggunakan bantuan XAMPP. Aktifkan XAMPP Control Panel Application, klik start apache dan mysql.



Aktifkan command prompt, lalu ketik seperti gambar berikut:

## MODUL 3

# KELOMPOK PERINTAH DALAM BASIS DATA



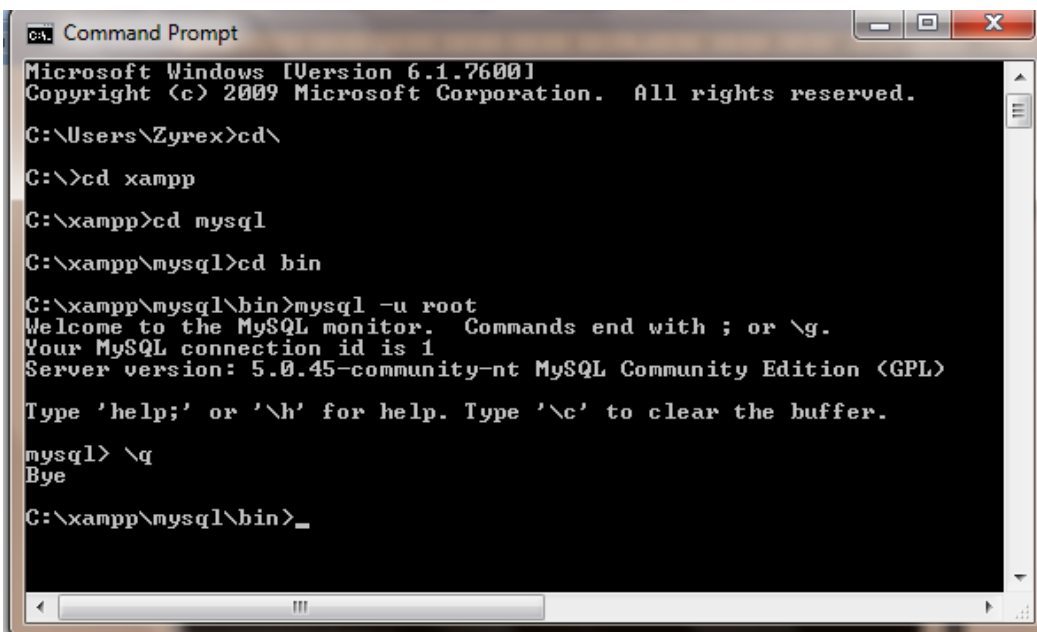
```
ca. Command Prompt - mysql -u root
Microsoft Windows [Version 6.1.7600]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

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.



```
ca. 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>_
```

# MODUL 3

## KELOMPOK PERINTAH DALAM BASIS DATA

### A. TIPE DATA

#### KARAKTER

- **CHAR**: Teks dengan maksimal 255 karakter
- **VARCHAR**: Teks maksimal 255 karakter dan bersifat variabel
- **TEXT**: Teks dengan panjang maksimal 65535

#### BILANGAN

- **TINYINT**: Bilangan 1 byte
- **SMALLINT**: Bilangan 2 byte
- **INT** atau **INTEGER**: Bilangan 4 byte
- **BIGINT**: Bilangan 8 byte
- **FLOAT**: Bilangan pecahan (4 byte)
- **DOUBLE** atau **REAL**: Bilangan pecahan (8 byte)
- **DECIMAL(M, D)** atau **NUMERIC(M, D)**: Bilangan pecahan

#### LAIN-LAIN

- **DATE**: Tanggal
- **DATETIME**: Waktu (tanggal dan jam)
- **TIME** : Jam
- **ENUM('nilai1', 'nilai2', ...)**: Nilai enumerasi
- **BOOLEAN**: tipe benar atau salah



# MODUL 3

## KELOMPOK PERINTAH DALAM BASIS DATA

### B. DDL (Data Definition Language)

*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;

## MODUL 3

# KELOMPOK PERINTAH DALAM BASIS DATA

```
mysql> use CU_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_cu_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)
```

## MODUL 3

# KELOMPOK PERINTAH DALAM BASIS DATA

- 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.

## MODUL 3

# KELOMPOK PERINTAH DALAM BASIS DATA

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;
```

- l. 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, typedatanya 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);
```

## MODUL 3

# KELOMPOK PERINTAH DALAM BASIS DATA

- n. Menghapus Kolom Pada Tabel :

ALTER TABLE namatabel DROP COLUMN namakolom;

### C. DML (Data Manipulation Language)

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.

## MODUL 3

# KELOMPOK PERINTAH DALAM BASIS DATA

### 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.

- 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)
```

- Menampilkan data untuk kolom tertentu :

SELECT kolom1,kolom2,kolom-n FROM namatabel;

## MODUL 3

# KELOMPOK PERINTAH DALAM BASIS DATA

```
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.

- d. 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;

- e. 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;

## MODUL 3

# KELOMPOK PERINTAH DALAM BASIS DATA

- f. Menampilkan data lebih dari dua tabel: `SELECT * FROM namatabel1, namatabel2, namatabel-n;`
- g. 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:



## MODUL 3

# KELOMPOK PERINTAH DALAM BASIS DATA

```
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);
```

### h. 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
```

## MODUL 3

# KELOMPOK PERINTAH DALAM BASIS DATA

FROM

JENISFILM);

### i. 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;
```

### j. 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.

### k. 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 film dimana jenisnya action dan horor:

## MODUL 3

# KELOMPOK PERINTAH DALAM BASIS DATA

```
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.

# MODUL 3

## KELOMPOK PERINTAH DALAM BASIS DATA

### D. DATA CONTROL LANGUAGE (DCL)

DCL adalah sub bahasa SQL yang berfungsi untuk melakukan pengontrolan data dan server databasenya, seperti manipulasi user dan hak akses (priviledges). Yang termasuk perintah dalam DCL ada dua, yaitu GRANT dan REVOKE.

#### GRANT

Perintah ini digunakan untuk memberikan hak akses oleh admin ke salah satu user atau pengguna. Hak akses tersebut bisa berupa hak membuat (CREATE), mengambil data (SELECT), menghapus data (DELETE), mengubah data (UPDATE), dan hak khusus lainnya yang berhubungan dengan sistem database.

caranya buatlah user terlebih dahulu dengan perintah sebagai berikut:

```
CREATE USER "admin_sekolah"@"localhost" IDENTIFIED BY "admin";
```

Atau

```
CREATE USER "opt_sekolah"@"localhost" IDENTIFIED BY "opt_sekolah";
```

Kemudian berikan hak akses terhadap user yang telah dibuat dengan perintah sebagai berikut:

```
GRANT ALL PRIVILEGES ON sekolah.* TO "admin_sekolah"@"localhost";
```

*Catatan:*

Hak akses ALL diberikan untuk user dapat menggunakan seluruh perintah SQL yakni CREATE, DROP, INSERT, UPDATE, DELETE, SELECT, ALTER

Untuk merubah password dapat dilakukan dengan perintah SQL seperti contoh sebagai berikut:

```
UPDATE mysql.user SET Password = PASSWORD('password_baru') WHERE User = 'root';
```

## MODUL 3

# KELOMPOK PERINTAH DALAM BASIS DATA

Sedangkan perintah SQL yang digunakan untuk menghapus user, dapat digunakan perintah sebagai berikut:

```
DROP user 'nama_user';
```

### **REVOKE**

perintah ini digunakan untuk mencabut hak akses yang telah diberikan kepada user. Dalam ini merupakan kebalikan dari perintah GRANT.

Contoh:

```
REVOKE SELECT ON universitas.* FROM 'admin_data'@'localhost';
```

```
REVOKE ALL ON dbsekolah.* TO 'user1'@'localhost';
```

# MODUL 3

## KELOMPOK PERINTAH DALAM BASIS DATA

### C. Evaluasi

#### TUGAS

1. Buatlah Database dengan nama Hotel ?
2. Buatlah tabel dengan nama Penginap, dengan data sbb:

Field Name	Data type	Field Size	Description
KdPenginap	Varchar	9	Kode Penginap
NamaPenginap	Varchar	50	Nama Penginap
AlamatPenginap	Varchar	50	Alamat Penginap
Usia	Integer	3	Usia Penginap
TlpPenginap	Varchar	15	Telepon Penginap

3. Tampilkan deskripsi dari table Penginap yang sudah pernah dibuat ?
4. Tambahkan kolom baru bernama Status dengan posisi setelah kolom/field Usia dengan tipe ENUM dengan keterangan Menikah atau Belum Menikah pada tabel Penginap?
5. Rubahlah kolom TlpPenginap menjadi TeleponPenginap dengan typedata sama pada table Penginap tersebut?
6. Masukkan lima data kedalam tabel Penginap ?
7. Rubahlah nama tabel Penginap dengan DataPenginap ?
8. Tampilkan data Kode, Nama, dan Usia pada tabel Penginap?
9. Ubah salah satu AlamatPenginap pada tabel Penginap dengan perintah UPDATE ?
10. Hapus salah satu data penginap dengan perintah DELETE ?