



**SMK-TI  
TRAINING  
AND CERTIFICATION**

---

---

**Modul 3**

**PENGENALAN SINTAKS DASAR  
DALAM MySQL**

**SMK-TI  
TRAINING  
AND CERTIFICATION**

---

**Tujuan:**

- ❖ Siswa dapat membuat database dan table sederhana yang dilengkapi beberapa kolom dengan tipe tertentu.
- ❖ Siswa dapat menerapkan sintaks dasar SQL dalam MySQL baik dalam level database maupun level table.
- ❖ Siswa dapat memulai dan menghentikan server MySQL
- ❖ Siswa dapat mengatur dan memelihara (maintenance) user account MySQL

# Pengenalan Sintaks Dasar SQL dalam MySQL

## 1. Pendahuluan

Dalam modul ini yang paling penting adalah dengan memperbanyak latihan dan mempraktekkan semua yang telah disampaikan. Disini hanya akan diberikan pengenalan tentang sintaks SQL dalam MySQL yang sederhana dan paling sering digunakan baik untuk penggunaan biasa ataupun untuk administratif.

Berikut adalah beberapa sintaks dasar SQL dalam MySQL

### a. Level Database

- Membuat database

Untuk membuat database baru, sehingga tidak berlaku jika database sudah ada atau anda tidak memiliki privilege.

Sintaksnya :

```
CREATE DATABASE nama_db
```

- Menghapus database

Untuk menghapus database beserta seluruh table di dalamnya. Perintah ini tidak berlaku jika database tidak ada atau anda tidak memiliki privilege.

Sintaksnya :

```
DROP DATABASE nama_db
```

- Menggunakan database

Untuk menjadikan database menjadi default dan referensi dari table yang nantinya anda gunakan. Perintah ini tidak berlaku jika database tidak ada atau anda tidak memiliki privilege. Sintaksnya :

```
USE nama_db
```

- Menampilkan database

Untuk menampilkan daftar yang ada dalam system saat itu. Sintaksnya :

```
SHOW DATABASES
```

Tampilannya adalah :

```

+-----+
| Database |
+-----+
| contoh_db |
|   mysql   |
|   test    |
|   ujian   |
+-----+
4 rows in set (0.00 sec)
```

## b. Level Tabel

- Membuat table

Untuk membuat table minimal anda harus menentukan namanya dan tipe kolom yang anda inginkan. Sintaks yang paling sederhana (tanpa ada definisi lain) adalah :

```
CREATE TABLE nama_tbl
(kolom1 tipekolom1(),kolom2 tipekolom2(), ...)
```

Contoh : Anda ingin membuat table dengan nama profil yang memiliki kolom nama (bertipe char, lebar 20), kolom umur (bertipe integer), kolom jenis\_kelamin (bertipe enum, berisi M dan F). Sintaksnya :

```
CREATE TABLE profil (
nama CHAR(20), umur INT NOT NULL,
jenis_kelamin ENUM('F','M') )
```

Sedangkan perintah yang agak lengkap dalam membuat sebuah table adalah dengan menyertakan definisi tertentu. Misalnya perintah seperti ini :

```
CREATE TABLE peserta (
No SMALLINT UNSIGNED NOT NULL AUTO_INCREMENT,
Nama CHAR(30) NOT NULL,
BidangStudi ENUM('TS','WD') NOT NULL,
PRIMARY KEY (No),
INDEX (Nama, BidangStudi) )
```

Perintah di atas berarti membuat table peserta dengan kolom No sebagai PRIMARY KEY yaitu indeks table yang unik yang tidak bisa diduplikat dengan atribut AUTO\_INCREMENT yaitu kolom yang otomatis dapat mengurutkan angka yang diisikan padanya. Sedangkan kolom Nama dan BidangStudi dijadikan indeks biasa.

- Membuat indeks pada table

Menambahkan indeks pada table yang sudah ada baik yang unik ataupun yang biasa.

Sintaksnya :

```
CREATE INDEX nama_index ON nama_tbl (nama_kolom)
CREATE UNIQUE INDEX nama_index ON nama_tbl (nama_kolom)
```

- Menghapus table

Untuk menghapus table dalam database tertentu. Jika dilakukan maka semua isi, indeks dan atribut lain akan terhapus. Sintaksnya :

```
DROP TABLE nama_tbl
```

- Menghapus indeks

Untuk menghapus indeks pada suatu table. Sintaksnya :

```
DROP INDEX nama-index ON nama_tbl
```

- Melihat informasi table

Untuk melihat table apa saja yang ada di database tertentu. Sintaksnya :

```
SHOW TABLES FROM nama_db
```

Sedangkan untuk melihat deskripsi table atau informasi tentang kolom gunakan sintaks :

```
DESC nama_tbl nama_kolom
```

atau SHOW COLUMNS FROM nama\_tbl

```
FROM nama_db
```

Misal untuk contoh di atas akan ditampilkan :

```
+-----+
| Tables_in_contoh_db |
+-----+
| peserta             |
| profil             |
+-----+
2 rows in set (0.00 sec)

+-----+-----+-----+-----+-----+-----+
| Field          | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| nama           | char(20)      | YES  |     | NULL    |      |
| umur          | int(11)       |      |     | 0       |      |
| jenis_kelamin | enum('F','M') | YES  |     | NULL    |      |
+-----+-----+-----+-----+-----+-----+
3 rows in set (0.02 sec)
```

- Mendapatkan atau menampilkan informasi dari table

Untuk menampilkan isi table dengan option-option tertentu. Misalnya untuk menampilkan seluruh isi table digunakan :

```
SELECT * FROM nama_tbl
```

Untuk menampilkan kolom-kolom tertentu saja :

```
SELECT kolom1, kolom2, ... FROM nama_tbl
```

Untuk menampilkan isi suatu kolom dengan kondisi tertentu

```
SELECT kolom1 FROM nama_tbl WHERE kolom2=isikolom
```

- Modifikasi struktur table

Dapat digunakan untuk mengganti nama table atau mengubah strukturnya seperti manambah kolom atau indeks, menghapus kolom atau indeks, mengubah tipe kolom dsb. Sintaks umum :

```
ALTER TABLE nama_tbl action
```

Untuk menambah kolom baru di tempat tertentu dapat menggunakan :

```
ALTER TABLE nama_tbl
```

```
ADD kolom_baru type() definisi
```

Untuk menambah kolom\_baru bertipe integer setelah kolom1 digunakan :

```
ALTER TABLE nama_tbl
ADD kolom_baru INT NOT NULL AFTER kolom1
```

Untuk menambah indeks baru pada table tertentu baik yang unik ataupun yang biasa:

```
ALTER TABLE nama_tbl ADD INDEX nama_index (nama_kolom)
ALTER TABLE nama_tbl ADD UNIQUE nama_indeks (nama_kolom)
ALTER TABLE nama_tbl ADD PRIMARY KEY nama_indeks (nama_kolom)
```

Untuk mengubah nama kolom dan definisinya, misalnya mengubah nama kolom\_baru dengan tipe integer menjadi new\_kolom dengan tipe char dengan lebar 30 digunakan:

```
ALTER TABLE nama_tbl
CHANGE kolom_baru new_kolom CHAR(30) NOT NULL
```

Untuk menghapus suatu kolom dan seluruh atributnya, misal menghapus kolom1

:

```
ALTER TABLE nama_tbl DROP kolom1
```

Untuk menghapus indeks baik yang unik ataupun yang biasa digunakan :

```
ALTER TABLE nama_tbl DROP nama_index
ALTER TABLE nama_tbl DROP PRIMARY KEY
```

- **Modifikasi informasi dalam table.**

Untuk menambah record atau baris baru dalam table, sintaksnya :

```
INSERT INTO nama_tbl (nama_kolom) VALUES (isi_kolom)
atau
INSERT INTO nama_tbl SET nama_kolom=isi_kolom
```

Misalnya untuk menambah dua baris pada table profil dengan isi nama = deden & ujang dan isi umur = 17 & 18 adalah :

```
INSERT INTO profil (nama,umur) VALUES (deden,17), (ujang,18)
atau
INSERT INTO profil SET nama=deden, umur=17
INSERT INTO profil SET nama=ujang, umur=18
```

Untuk memodifikasi record atau baris yang sudah ada yang bersesuaian dengan suatu kolom. Misalnya untuk mengubah umur deden menjadi 18 pada contoh di atas dapat digunakan sintaks :

```
UPDATE profil SET umur=18
WHERE nama=deden
```

Untuk menghapus record atau baris tertentu dalam suatu table. Misalnya untuk menghapus baris yang ada nama ujang digunakan sintaks :

```
DELETE FROM profil WHERE
nama=ujang
```

Jika WHERE tidak disertakan maka semua isi dalam table profil akan terhapus.

## 2. Tipe-tipe data MySQL

MySQL dapat mengetahui beberapa tipe data antara lain :

- **Data Numerik**

MySQL dapat menerima masukan berupa angka-angka yang dibagi atas integer (angka tanpa pecahan) dan floating-point (angka dengan pecahan).

MySQL juga mengerti notasi scientific yaitu integer atau floating-point yang diikuti tanda 'e' atau 'E', tanda '+' atau '-'. Misalnya angka 1.34E+12 atau 3.23e-5.

- **Data Karakter/String**

Merupakan deretan huruf yang membentuk kata yang diapit oleh tanda petik (") atau tanda petik ganda ("").

- Data Waktu  
Merupakan data yang berisi tanggal (date) dan jam (time) misalnya "2001-10-15" untuk tanggal dengan format YYYY-MM-DD dan "12:45:15" untuk jam dengan format hh:mm:ss.
- Data kosong (NULL)  
NULL berarti kosong atau tidak diisi data atau bisa juga berarti data yang tidak jelas, data yang hilang ataupun yang lainnya.

## 2.1 Tipe-tipe kolom MySQL

Setiap table yang dibuat dalam database selalu terdiri atas kolom-kolom. Ketika anda membuatnya dengan perintah CREATE TABLE, anda harus menentukan tipe masing-masing kolom. Tiap tipe kolom memiliki karakteristik berikut :

- Jenis harga apa yang dapat diisikan
- Berapa banyak ruang yang dapat menampung harga tersebut
- Bagaimana harga dari tipe tersebut dibandingkan dan disaring
- Apakah tipe tersebut boleh mengisi dengan NULL atau tidak
- Apakah tipe tersebut boleh diindeks atau tidak

Secara garis besar kolom MySQL terbagi menjadi tiga tipe yaitu :

- Tipe kolom Numerik
- Tipe kolom Karakter/String
- Tipe kolom Waktu

Selanjutnya akan dijelaskan dengan singkat anggota-anggotanya.

## 2.2 Tipe Numerik

Tipe ini untuk harga integer dan floating-point. Untuk integer kolom haruslah PRIMARY KEY atau indeks yang unik jika ia diberi atribut AUTO\_INCREMENT (dapat otomatis mengurutkan angka). Jika diberikan atribut UNSIGNED berarti angka tidak boleh negatif. Sedangkan atribut ZEROFILL menandakan bahwa angka diawali dengan angka nol.

- TINYINT  
Berarti integer dengan range yang sangat kecil yaitu  $-2^7$  sampai  $2^7-1$  atau 0 sampai  $2^8-1$  jika UNSIGNED. Atribut yang dibolehkan adalah



AUTO\_INCREMENT, UNSIGNED, dan ZEROFILL. Harga default adalah NULL jika bisa atau 0 jika NOT NULL dengan penyimpanan 1 byte.

- SMALLINT

Berarti integer dengan range yang kecil yaitu  $-2^{15}$  sampai  $2^{15}-1$  atau 0 sampai  $2^{16}-1$  jika UNSIGNED. Atribut yang dibolehkan adalah AUTO\_INCREMENT, UNSIGNED, dan ZEROFILL. Harga default adalah NULL jika bisa atau 0 jika NOT NULL dengan penyimpanan 2 byte.

- MEDIUMINT

Berarti integer dengan range yang sangat kecil yaitu  $-2^{23}$  sampai  $2^{23}-1$  atau 0 sampai  $2^{24}-1$  jika UNSIGNED. Atribut yang dibolehkan adalah AUTO\_INCREMENT, UNSIGNED, dan ZEROFILL. Harga default adalah NULL jika bisa atau 0 jika NOT NULL dengan penyimpanan 3 byte.

- INT

Berarti integer dengan range yang normal yaitu  $-2^{31}$  sampai  $2^{31}-1$  atau 0 sampai  $2^{32}-1$  jika UNSIGNED. Atribut yang dibolehkan adalah AUTO\_INCREMENT, UNSIGNED, dan ZEROFILL. Harga default adalah NULL jika bisa atau 0 jika NOT NULL dengan penyimpanan 4 byte.

- BIGINT

Berarti integer dengan range yang sangat kecil yaitu  $-2^{63}$  sampai  $2^{63}-1$  atau 0 sampai  $2^{64}-1$  jika UNSIGNED. Atribut yang dibolehkan adalah AUTO\_INCREMENT, UNSIGNED, dan ZEROFILL. Harga default adalah NULL jika bisa atau 0 jika NOT NULL dengan penyimpanan 8 byte.

- FLOAT

Berarti floating-point dengan range kecil yaitu antara  $\pm 1.175494351E-38$  sampai  $\pm 3.402823466E+38$  serta dengan single presisi. Atribut yang dibolehkan adalah ZEROFILL. Harga default NULL jika bisa atau 0 jika NOT NULL. Tempat penyimpanan 4 byte.

- DOUBLE

Berarti floating-point dengan range besar yaitu antara  $\pm 2.22507385072014E-308$  sampai  $\pm 1.7976931348623157E308$  serta dengan

double presisi. Atribut yang dibolehkan adalah ZEROFILL. Harga default NULL jika bisa atau 0 jika NOT NULL. Tempat penyimpanan 8 byte.

- DECIMAL(M,D) atau NUMERIC(M,D)  
Berarti floating-point yang tersimpan sebagai string (1 byte untuk setiap digit, tanda desimal, atau tanda '-'). Range harga sama seperti DOUBLE. Atribut yang dibolehkan adalah ZEROFILL. Tempat penyimpanan sebesar M byte. Jika D diisi 0 berarti tidak punya nilai desimal.

### 2.3 Tipe String

- CHAR(M)  
Karakter dengan panjang 0 sampai M byte. Atribut yang dibolehkan adalah BINARY. Harga default adalah NULL jika bisa atau " " jika NOT NULL. Tempat penyimpanan M byte.
- VARCHAR  
Variabel karakter dengan panjang 0 sampai M byte. Atribut yang dibolehkan adalah BINARY. Harga default adalah NULL jika bisa atau " " jika NOT NULL. Tempat penyimpanan M+1 byte.
- TINYTEXT  
Teks berukuran kecil dengan panjang 0 sampai  $2^8-1$  byte. Harga default adalah NULL jika bisa atau " " jika NOT NULL. Tempat penyimpanan sebanyak panjang harga plus 1 byte.
- TEXT  
Teks yang normal dengan panjang 0 sampai  $2^{16}-1$  byte. Harga default adalah NULL jika bisa atau " " jika NOT NULL. Tempat penyimpanan sebanyak panjang harga plus 2 byte.
- MEDIUMTEXT  
Teks berukuran sedang dengan panjang 0 sampai  $2^{24}-1$  byte. Harga default adalah NULL jika bisa atau " " jika NOT NULL. Tempat penyimpanan sebanyak panjang harga plus 3 byte.
- LONGTEXT

Teks berukuran besar dengan panjang 0 sampai  $2^{32}-1$  byte. Harga default adalah NULL jika bisa atau " " jika NOT NULL. Tempat penyimpanan sebanyak panjang harga plus 4 byte.

- ENUM("harga1","harga2",...)

Berarti kolom hanya boleh diisi dengan salah satu dari harga yang ada. Harga default adalah NULL jika bisa atau harga1 jika NOT NULL. Tempat penyimpanan adalah 1 byte untuk enumerasi dengan anggota 1 sampai 255 dan 2 byte untuk enumerasi dengan anggota 256 sampai 65535.

- SET("harga1","harga2",...)

Berarti kolom boleh dikosongi atau diisi dengan beberapa harga dari daftar harga yang ada. Harga default adalah NULL jika bisa atau " " jika NOT NULL. Tempat penyimpanan adalah 1 byte untuk set dengan anggota 1 sampai 8, 2 byte untuk set dengan anggota 9 sampai 16, 3 byte untuk set dengan anggota 17 sampai 24, 4 byte untuk set dengan anggota 25 sampai 32, atau 8 byte untuk set dengan anggota 33 sampai 64.

## 2.4 Tipe Waktu

- DATE

Untuk kolom tanggal dengan format YYYY-MM-DD dan range antara "1000-01-01" sampai "9999-12-31". Harga default adalah NULL jika bisa atau "0000-00-00" jika NOT NULL. Tempat penyimpanan 3 byte.

- TIME

Untuk kolom jam dengan format hh:mm:ss atau -hh:mm:ss untuk harga negatif. Range harga antara "-838:59:59" sampai "838:59:59". Harga default adalah NULL jika bisa atau "00:00:00" jika NOT NULL. Tempat penyimpanan 3 byte.

- DATETIME

Gabungan antar hari dan jam dengan format YYYY-MM-DD hh:mm:ss dan range antar "1000-01-01 00:00:00" sampai "9999-12-31 23:59:59". Harga

default adalah NULL jika bisa atau "0000-00-00 00:00:00" jika NOT NULL.

Tempat penyimpanan 8 byte.

- **TIMESTAMP**

Hampir sama dengan DATETIME tapi dengan format YYYYMMDDhhmmss dan range antara 19700101000000 sampai suatu saat di tahun 2037. Harga default adalah hari dan jam saat itu. Tempat penyimpanan 4 byte.

- **YEAR**

Untuk kolom tahun dengan format YYYY dan range antara 1900 sampai 2155. Harga default adalah NULL jika bisa atau 0000 jika NOT NULL. Tempat penyimpanan 3 byte.