



# Data Definition Language (DDL)

Modul Praktikum  
Sistem Manajemen  
Basis Data 2025

Tim Asisten  
Sistem Manajemen  
Basis Data 2025



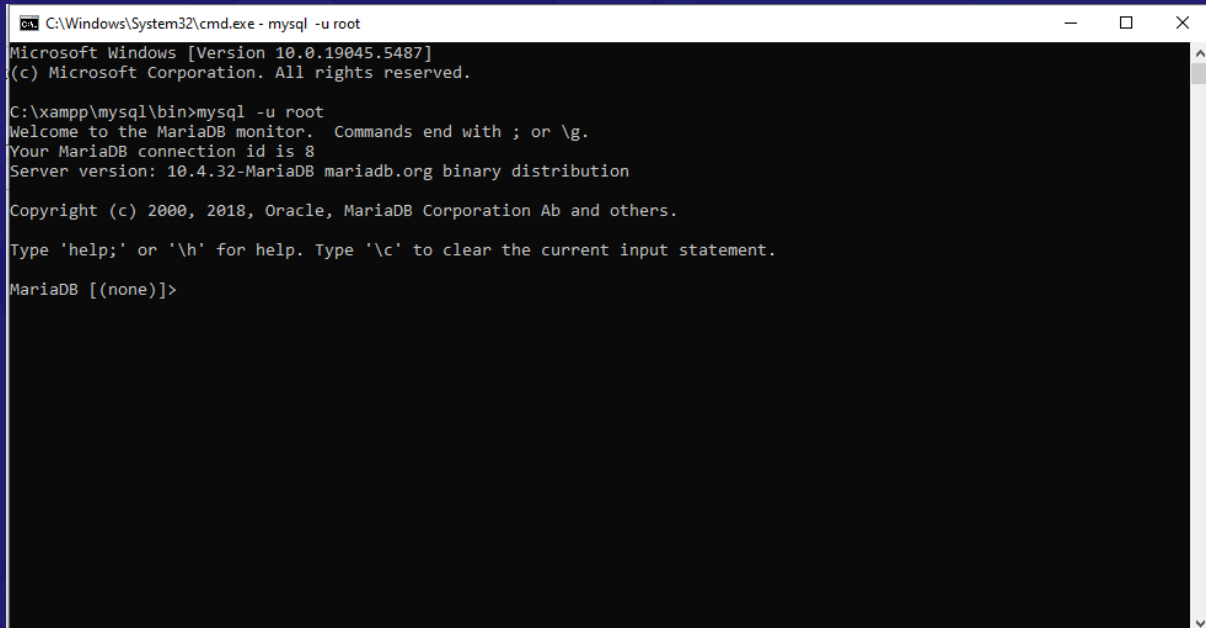
## MySQL (My Structured Query Language)

MySQL adalah RDBMS (Relational Database Management System) sumber terbuka yang menggunakan SQL untuk membuat dan mengelola database. Sebagai database relasional, MySQL menyimpan data dalam tabel yang terdiri dari baris dan kolom yang diorganisasikan ke dalam skema. Skema mendefinisikan bagaimana data diatur dan disimpan dan menggambarkan hubungan di antara berbagai tabel.

### Cara akses MySQL dari CLI

#### Pastikan MySQL pada XAMPP sudah aktif

- Navigasi ke Direktori MySQL kalian di command prompt, contoh: C:\xampp\mysql\bin
- Ketik syntax **mysql -u root** di command prompt



```
C:\Windows\System32\cmd.exe - mysql -u root
Microsoft Windows [Version 10.0.19045.5487]
(c) Microsoft Corporation. All rights reserved.

C:\xampp\mysql\bin>mysql -u root
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 8
Server version: 10.4.32-MariaDB mariadb.org binary distribution

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

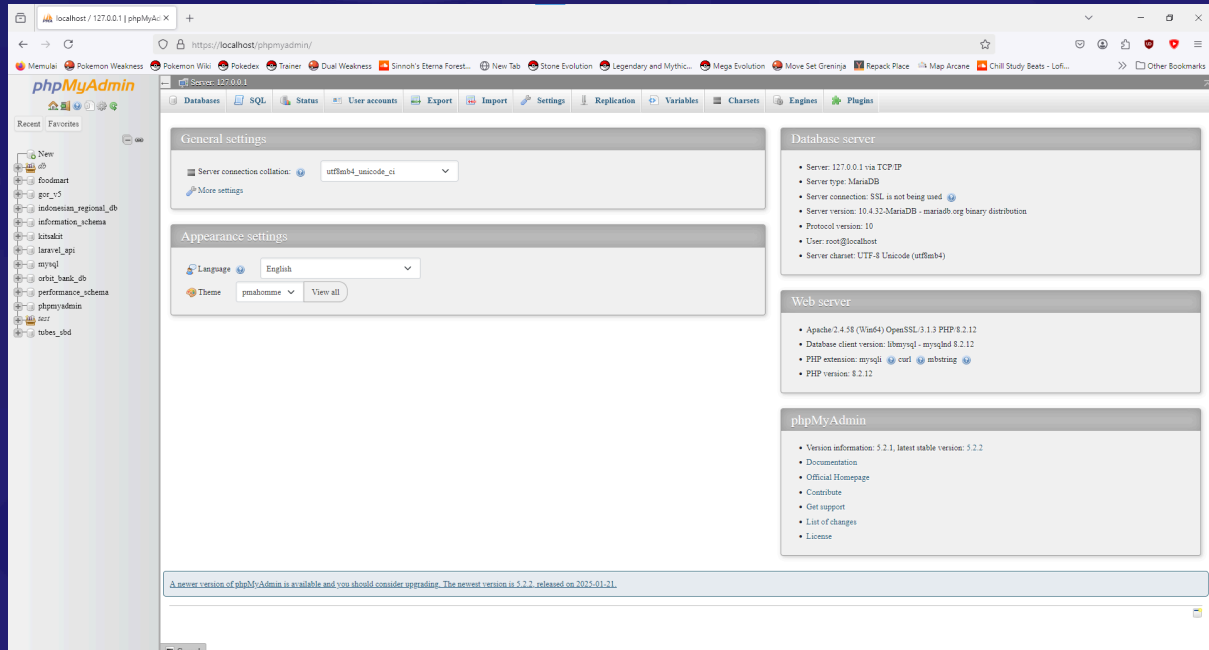
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]>
```

## Cara akses MySQL dari Website (PHPMyAdmin)

Pastikan server apache sudah aktif

- Ketik **localhost/phpmyadmin** di address bar browser kalian.



## DDL (Data Definition Language)

DDL digunakan untuk mendefinisikan struktur dan karakteristik dari objek-objek dalam basis data, seperti **tabel, indeks, dan tipe data**. Anggap seperti mendefinisikan tabel beserta kolomnya.

### Perintah-Perintah DDL:

CREATE	ALTER	DELETE
CREATE DATABASE	ALTER DATABASE	DROP DATABASE
CREATE FUNCTION	ALTER FUNCTION	DROP FUNCTION
CREATE PROCEDURE	ALTER PROCEDURE	DROP PROCEDURE
CREATE INDEX		DROP INDEX

CREATE TRIGGER		DROP TRIGGER
CREATE VIEW	ALTER VIEW	DROP VIEW
CREATE TABLE	ALTER/RENAME TABLE	DROP TABLE

### Tipe-Tipe Bilangan:

Tipe Data	Deskripsi
<b>TINYINT</b>	Digunakan untuk menyimpan data <b>bilangan bulat</b> (positif & negatif) yang <b>kecil</b> . Jangkauan: -128 s/d 127 Ukuran: 1 byte (8 bit)
<b>SMALLINT</b>	Digunakan untuk menyimpan data <b>bilangan bulat</b> (positif & negatif) yang <b>sedang</b> . Jangkauan: -8.388.608 s/d 8.388.607 Ukuran: 3 Byte (24 bit)
<b>INT</b>	Digunakan untuk menyimpan data <b>bilangan bulat</b> (positif & negatif) yang <b>cukup besar</b> . Jangkauan: -2.147.483.648 s/d -2.147.483.647 Ukuran: 4 Byte (32 bit)
<b>BIGINT</b>	Digunakan untuk menyimpan data <b>bilangan bulat</b> (positif & negatif) yang <b>sangat besar</b> . Jangkauan: $9,22 \times 10^{18}$ Ukuran: 8 Byte (64 bit)
<b>FLOAT</b>	Digunakan untuk menyimpan data bilangan <b>pecahan positif dan negatif dengan presisi tunggal</b> . Jangkauan: $-3.402823466E+38$ s/d $-1.175494351E-38$ , $0$ , $1.175494351E-38$ s/d $3.402823466E+38$ Ukuran: 4 byte (32 bit)
<b>DOUBLE/REAL</b>	Digunakan untuk menyimpan data bilangan pecahan positif dan negatif presisi ganda <b>ukuran sedang</b> .
<b>DECIMAL</b>	Digunakan untuk menyimpan data bilangan pecahan positif dan negatif yang <b>sangat besar</b> . Jangkauan: $-1.79...E+38$ s/d $-2.22...E-308$ , $0$ , $2.22...E-308$ s/d $1.79...E+308$



### Tipe-Tipe Data Jam, Tanggal, dan Hari:

Tipe Data	Deskripsi
DATE	Digunakan untuk menyimpan data <b>tanggal</b> . Jangkauan: 1001-01-01 s/d 9999-12-32 (YYYY-MM-DD) Ukuran: 3 byte
TIME	Digunakan untuk menyimpan data <b>waktu</b> . Jangkauan: -838:59:59 s/d 838:59:59 (HH:MM:SS) Ukuran: 3 Byte
DATETIME	Digunakan untuk menyimpan data <b>Tanggal BESERTA Waktu</b> . Jangkauan: 1000-01-01 00:00:00 s/d 9999-12-32 23:59:59 Ukuran: 8 Byte
TIMESTAMP	Pencatatan yang berisikan <b>kombinasi Tanggal dan Jam</b> saat tabel/data diakses. Jangkauan: 1970-01-01 00:00:01 s/d 2038:01:19 03:14:07 Ukuran: 4 Byte
YEAR	Digunakan untuk menyimpan data <b>Tahun</b> . Jangkauan: 1901 s/d 2155 Ukuran: 1 Byte

### Tipe-Tipe Data String

Tipe Data	Deskripsi
CHAR	Digunakan untuk menyimpan data <b>String berukuran Tetap (static)</b> . Jangkauan: 0 s/d 255 Karakter
VARCHAR	Digunakan untuk menyimpan data <b>String berukuran Dinamis</b> . Jangkauan: 0 s/d 255 karakter (versi 4.1), 0 s/d 65.535 (versi 5.0.3)

<b>TINYBLOB/TINYTEXT</b>	L+1 byte, dengan L<28. Tipe data <b>TEXT</b> atau <b>BLOB</b> dengan panjang max 255 karakter.
<b>MEDIUMBLOB/MEDIUMTEXT</b>	L+3 byte, dengan L<28. Tipe data <b>TEXT</b> atau <b>BLOB</b> dengan panjang max 1677215 karakter.
<b>LONGBLOB/LONGTEXT</b>	L+4 byte, dengan L<28. Tipe data <b>TEXT</b> atau <b>BLOB</b> dengan panjang max 4294967295 karakter.
<b>ENUM</b>	Digunakan untuk enumerasi (kumpulan data)
<b>SET</b>	Digunakan untuk combination (himpunan data) Jangkauan: < 255 string anggota

### Constraints

SQL Constraints digunakan untuk menentukan aturan data dalam table.

Constraints yang terdapat pada SQL:

- **NOT NULL:** Memastikan bahwa kolom tidak dapat memiliki nilai NULL, dan akan error jika NULL
- **UNIQUE:** Memastikan bahwa semua nilai dalam kolom berbeda, akan error jika ada yang sama
- **PRIMARY KEY:** Kombinasi dari NOT NULL dan UNIQUE, mengidentifikasi setiap baris dalam tabel secara unik. Anggap seperti NIK pada Manusia.
- **FOREIGN KEY:** Menghubungkan dua tabel, sehingga Mencegah tindakan yang akan merusak tautan antar tabel.
- **CHECK:** Memastikan bahwa nilai dalam kolom memenuhi kondisi tertentu. Jika nilai yang dimasukkan tidak memenuhi maka akan error.
- **DEFAULT:** Menetapkan nilai default untuk kolom jika tidak ada nilai yang ditentukan
- **CREATE INDEX:** Digunakan untuk membuat dan mengambil data dari database dengan sangat cepat

### Database

- Untuk menampilkan seluruh Database yang tersimpan:

```
SHOW databases;
```

- Cara Membuat Database Baru:

CREATE DATABASE [nama\_database]

Contoh: **CREATE DATABASE db\_toko\_baju;**

- Cara Menggunakan Database yang Tersimpan:

USE [nama\_database]

Contoh: **USE db\_toko\_baju;**

```
MariaDB [(none)]> USE db_toko_baju;
Database changed
MariaDB [db_toko_baju]>
```

- Menghapus Database yang Tersimpan:

DROP DATABASE [nama\_database]

Contoh: **DROP DATABASE db\_toko\_baju;**

```
MariaDB [(none)]> DROP DATABASE db_toko_baju;
Query OK, 0 rows affected (0.005 sec)

MariaDB [(none)]> USE db_toko_baju;
ERROR 1049 (42000): Unknown database 'db_toko_baju'
MariaDB [(none)]>
```

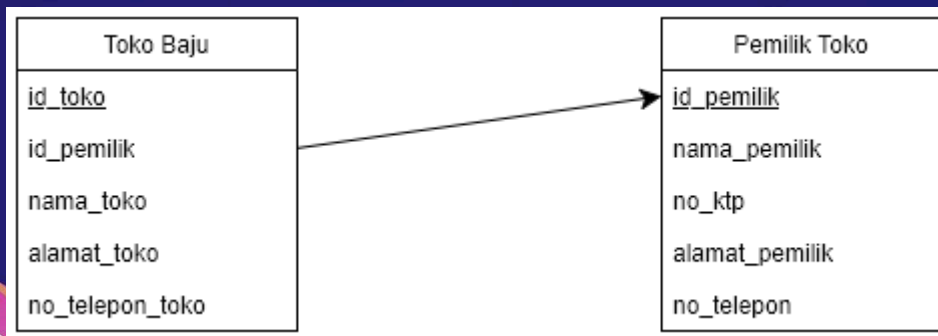
## Pembuatan Tabel

- Syntax Pembuatan Tabel:

```
CREATE TABLE [nama_tabel] (
  [nama_atribut1] [tipe_data] [constraint (opsional)],
  [nama_atribut1] [tipe_data] [constraint (opsional)],
  [nama_atribut1] [tipe_data] [constraint (opsional)],
  PRIMARY KEY ([nama_atribut])
  FOREIGN KEY ([nama_atribut]) REFERENCES [nama_tabel_relasi]
  ([nama_atribut]);
```

- Contoh:

Kita akan membuat 2 tabel ini pada database:



```
CREATE TABLE pemilik_toko(  
    id_pemilik int(11) NOT NULL AUTO_INCREMENT,  
    nama_pemilik varchar(255) NOT NULL,  
    no_ktp varchar(20) NOT NULL,  
    alamat_pemilik varchar(255) NOT NULL,  
    no_telepon varchar(10) NOT NULL,  
    PRIMARY KEY (id_pemilik)  
);
```

```
CREATE TABLE toko_baju(  
    id_toko int(11) NOT NULL AUTO_INCREMENT,  
    id_pemilik int(11) NOT NULL,  
    nama_toko varchar(255) NOT NULL,  
    alamat_toko varchar(255) NOT NULL,  
    no_telepon_toko varchar(10) NOT NULL,  
    PRIMARY KEY (id_toko),  
    FOREIGN KEY (id_pemilik) REFERENCES pemilik_toko (id_pemilik)  
);
```

- Hasil:

```
MariaDB [db_toko_baju]> show tables;  
+-----+  
| Tables_in_db_toko_baju |  
+-----+  
| pemilik_toko            |  
| toko_baju               |  
+-----+  
2 rows in set (0.000 sec)
```

## Modifikasi Tabel

Syntax Modifikasi Tabel:

```
ALTER TABLE [nama_tabel] [ADD | DROP | RENAME | MODIFY | CHANGE COLUMN]  
[kondisi]
```



- **Menambahkan Primary Key** pada Tabel:

```
ALTER TABLE [nama_tabel] ADD PRIMARY KEY ([nama_atribut]);
```

(Pastikan Atribut yang akan menjadi Primary Key sudah ada!)

- **Menambahkan Foreign Key** pada Tabel:

```
ALTER TABLE [nama_tabel] ADD FOREIGN KEY ([nama_atribut]) REFERENCES  
[nama_tabel_relasi] ([nama_atribut_relasi]);
```

- **Menghapus Primary Key**

Jika kalian ingat, pada pembuatan tabel kita memasukkan Constraint "AUTO INCREMENT" untuk atribut Primary Key. Jika kita ingin menghapus atribut tersebut, maka AUTO INCREMENT harus dihapuskan juga.

```
ALTER TABLE toko_baju modify id_toko int(11) NOT NULL
```

Dengan ini, Constraint id\_toko adalah **NOT NULL** saja.

Dilanjutkan dengan:

```
ALTER TABLE [nama_tabel] DROP primary key;
```

- **Menghapus ForeignKey pada Tabel**

Menghapus ForeignKey lebih ribet daripada PrimaryKey, karena nama Constraint yang dinamis. Oleh karena itu, kita harus mencari nama Constraint atribut dengan:

```
SHOW CREATE TABLE toko_baju;
```

```
MariaDB [db_toko_baju]> SHOW CREATE TABLE toko_baju;  
+-----+-----+  
| Table | Create Table  
+-----+-----+  
| toko_baju | CREATE TABLE `toko_baju` (  
  `id_toko` int(11) NOT NULL AUTO_INCREMENT,  
  `id_pemilik` int(11) NOT NULL,  
  `nama_toko` varchar(255) NOT NULL,  
  `alamat_toko` varchar(255) NOT NULL,  
  `no_telepon_toko` varchar(10) NOT NULL,  
  PRIMARY KEY (`id_toko`),  
  KEY `id_pemilik` (`id_pemilik`),  
  CONSTRAINT `toko_baju_ibfk_1` FOREIGN KEY (`id_pemilik`) REFERENCES `pemilik_toko` (`id_pemilik`)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci |  
+-----+-----+
```

Didapatilah Constraint ForeignKey: **toko\_baju\_ibfk\_1**

Selanjutnya kita hapus Constraint ForeignKey tersebut:

```
ALTER TABLE toko_baju DROP FOREIGN KEY toko_baju_ibfk_1;
```

Terakhir, kita drop INDEX nya.

```
ALTER TABLE toko_baju DROP INDEX id_pemilik;
```

- **Mengubah Nama Tabel**

```
ALTER TABLE [nama_tabel_lama] RENAME TO [nama_tabel_baru];
```

- **Menambahkan Kolom pada Tabel**

```
ALTER TABLE [nama_tabel] ADD [nama_kolom] [tipe_data] [constraint];
```

- **Mengubah Nama Kolom pada Tabel**

```
ALTER TABLE [nama_tabel] CHANGE [nama_atribut_lama]  
[nama_atribut_baru] [tipe_data] [constraint]
```

Contoh:

```
ALTER TABLE toko_baju CHANGE no_telepon_toko no_kontak_toko  
varchar(15) NOT NULL;
```

- **Menghapus Kolom pada Tabel**

```
ALTER TABLE [nama_tabel] DROP [nama_kolom];
```

- **Mengubah Tipe Data Kolom**

```
ALTER TABLE [nama_tabel] MODIFY [nama_kolom] [tipe_data];
```

- **Menghapus Tabel**

```
DROP TABLE [nama_tabel];
```

- **MENAMPILKAN TABLE YANG TERSIMPAN PADA DATABASE**

```
SHOW TABLES;
```

- **Melihat Struktur Tabel**

```
DESC [nama_tabel];
```

## Tugas

OTW ya tugasnya....

## Penutup

Kami mengucapkan terima kasih kepada Tuhan, Bangsa, dan Almamater. Yang telah berkontribusi dalam pembuatan modul ini. Semoga pengetahuan dan pengalaman yang diperoleh dapat memberikan manfaat yang signifikan bagi masa depan. Aamiin.

## Referensi

Asisten Praktikum Sistem Basis Data (2024). Modul Sistem Basis Data :  
Modul 3: SQL (DDL)