

# Definisi basis data

---

jika dibahasakan,basis adalah tempat atau wadah yang mempunyai aturan tertentu,dan data adalah kumpulan informasi yang sesuai fakta. Tetapi jika di definisikan **basis data** adalah kumpulan data yang berhubungan Secara logika disertai dengan deskripsi data.

# Peranan Basis Data

---

Perananan basis data sangat penting dalam pengolahan data,agar terstruktur.

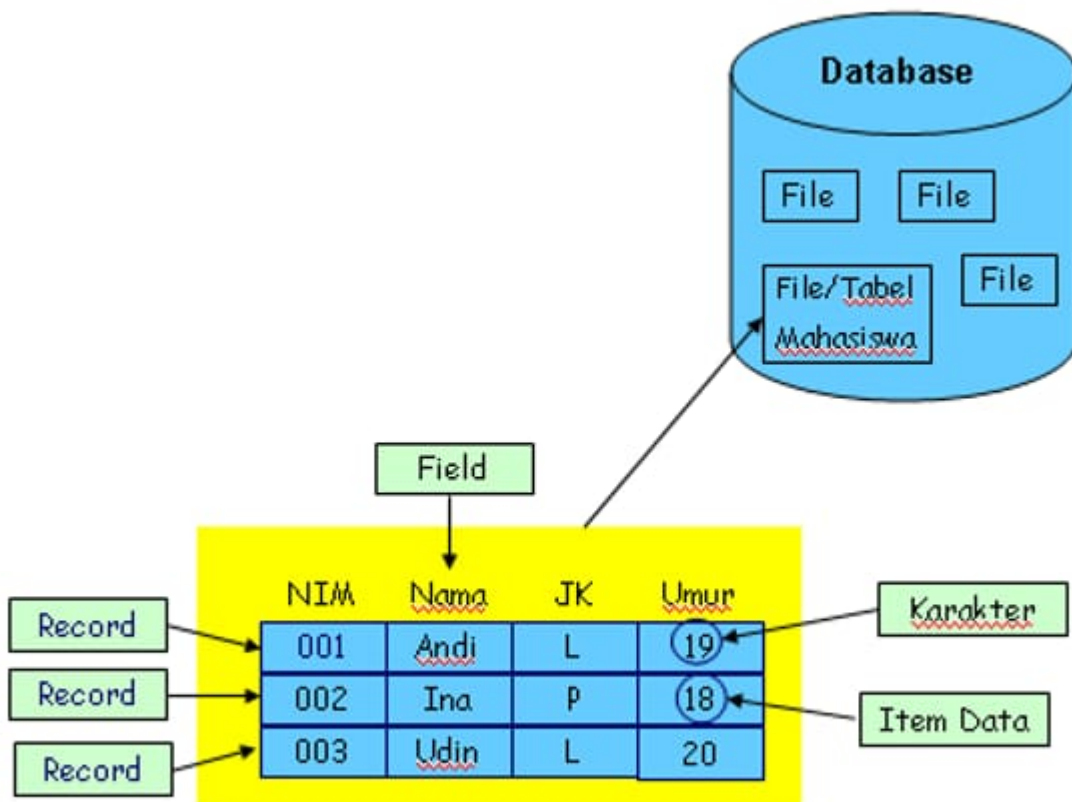
contoh yang bisa saya ambil di tempat *PKL* saya yaitu **PDAM** kota makassar,terdapat ruang server yang berisi **Data** identitas konsumen dari *Nama,Alamat,dan informasi* yang berhubungan antara **PDAM** dengan konsumen di seluruh kota makassar.

m

# Struktur Basis Data

---

Basis data di ibaratkan sebuah tabung yang didalamnya dapat mengolah sebuah file file dan juga tabel.pada gambar berikut terdapat ilustrasi dari basis data tersebut.



Dalam basis data terdapat juga yang dinamakan tabel, pada gambar di atas telah ditampilkan gambar tabel beserta struktur-struktur nya.

## PENJELASAN

- ◆ RECORD: **Record** adalah istilah dari sebuah baris pada tabel.
- ◆ FIELD : **Field** adalah istilah dari sebuah kolom pada tabel.
- ◆ KARAKTER : **karakter** adalah istilah dari sebuah nilai yang telah disikan pada sebuah tabel
- ◆ ITEM DATA: **Item data** adalah salah satu istilah dari nilai data sebuah tabel yang telah diisikan.

NO	NAMA	JK	KELAS
1	CHAIRIL ABIZALI	LAKI-LAKI	XI RPL 1
2	FACHRI RAMADHAN	MEN	XI RPL 1
3	RAYHAN JULI S (SETIA)	PRIA	XI RPL 1
4	NUR RAHMAT RAMADHAN	COWOK	XI RPL 1

## Info

Berikut adalah contoh pengaplikasian tabel yang telah dibuat.

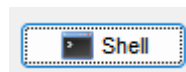
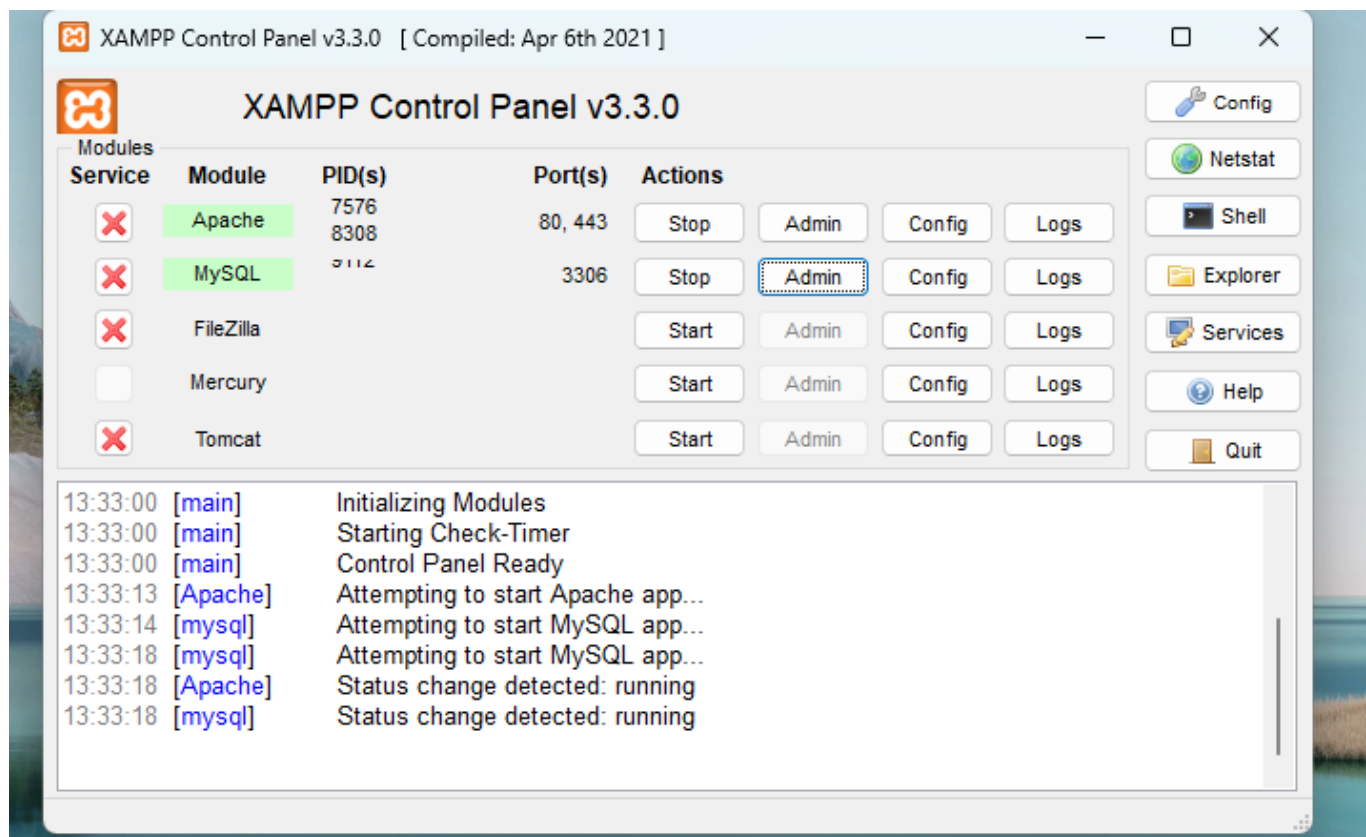
## DATA BASE

# Membuat data base

Data base adalah kumpulan informasi yang disimpan secara sistematis dalam komputer sehingga kemudian dapat diperiksa dengan mudah melalui program komputer tertentu.

Berikut adalah cara membuat data base:

- 1.masuk pada aplikasi xampp
- 2.pilih mode mysql lalu klik start
- 3.kemudian klik shell.



xampp-setup-storage

allow

mysql -u root -p

## Penerapan mysql

---

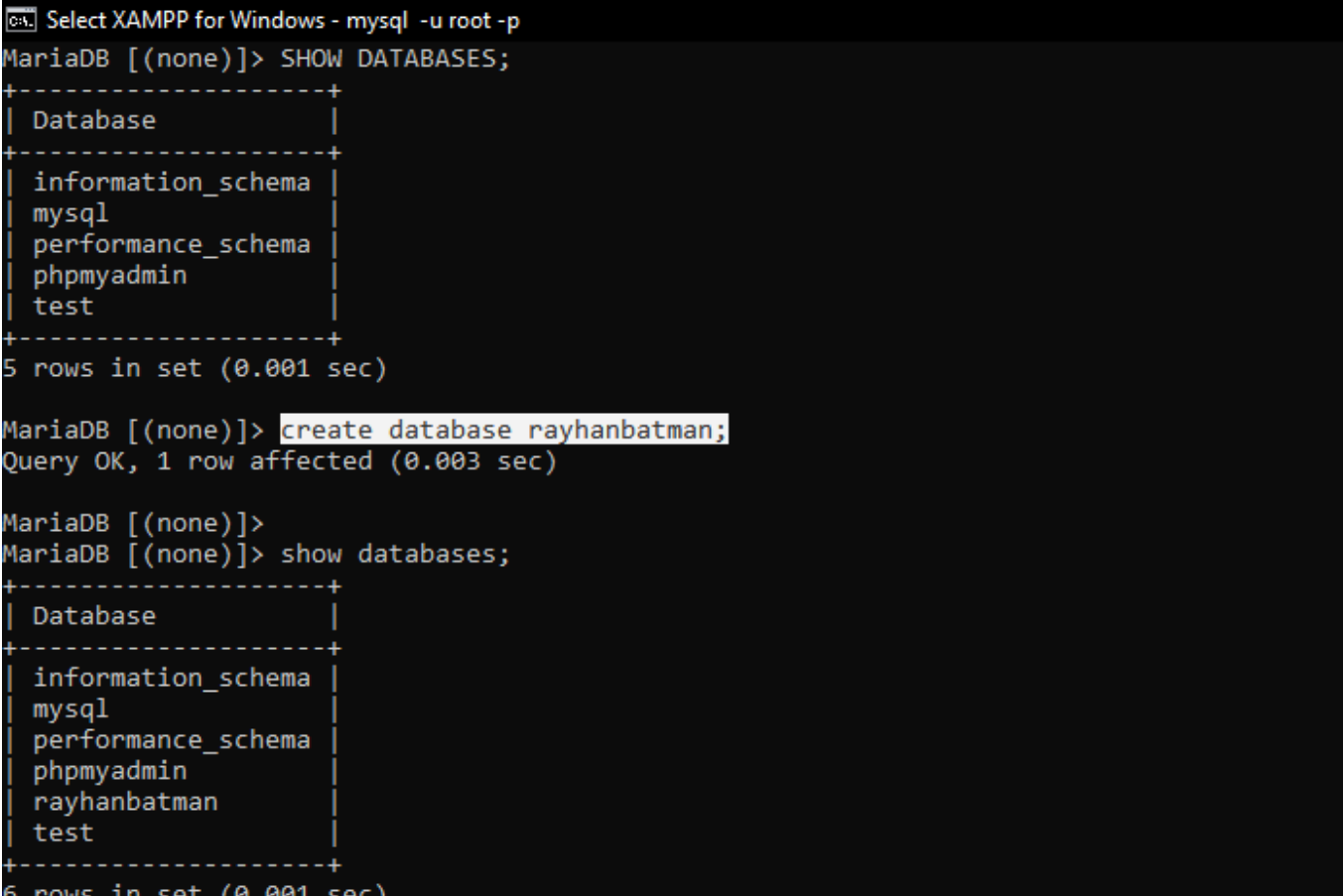
setelah pembuatan database telah selesai,kita dapat menerapkan mysql dengann menggunakan fitur adalah membuat,menghapus,melihat,dan menggunakan database.

- ♦ **Membuat Database**,dalam pembuatan database pada mysql,di awali dengan menggunakan query `CRATE DATABASE [NAMA DATABASE]` setelah pengetikan query database akan terbuat.

<<<<<< HEAD

## CONTOH GAMBAR:

---



```
C:\> Select XAMPP for Windows - mysql -u root -p
MariaDB [(none)]> SHOW DATABASES;
+-----+
| Database |
+-----+
| information_schema |
| mysql |
| performance_schema |
| phpmyadmin |
| test |
+-----+
5 rows in set (0.001 sec)

MariaDB [(none)]> create database rayhanbatman;
Query OK, 1 row affected (0.003 sec)

MariaDB [(none)]>
MariaDB [(none)]> show databases;
+-----+
| Database |
+-----+
| information_schema |
| mysql |
| performance_schema |
| phpmyadmin |
| rayhanbatman |
| test |
+-----+
6 rows in set (0.001 sec)
```

CONTOH GAMBAR:

```
C:\> Select XAMPP for Windows - mysql -u root -p
MariaDB [(none)]> SHOW DATABASES;
+-----+
| Database |
+-----+
| information_schema |
| mysql |
| performance_schema |
| phpmyadmin |
| test |
+-----+
5 rows in set (0.001 sec)

MariaDB [(none)]> create database rayhanbatman;
Query OK, 1 row affected (0.003 sec)

MariaDB [(none)]>
MariaDB [(none)]> show databases;
+-----+
| Database |
+-----+
| information_schema |
| mysql |
| performance_schema |
| phpmyadmin |
| rayhanbatman |
| test |
+-----+
6 rows in set (0.001 sec)
```

b. Pada bagian yang diblok pada gambar adalah sebuah perintah dengan *query* **create database rayhanbatman;** sebagai perintah mysql untuk membuat sebuah database dengan nama **rayhanbatman** pada mysql.

c. pada tabel kedua merupakan tampilan tabel yang telah berhasil membuat database dengan nama **rayhanbatman**.

- ♦ **Menghapus Database**, dalam penghapusan database pada mysql, diawali dengan menggunakan query **DROP DATABASE[NAMA DATABASE]** maka mysql akan melakukan penghapusan Database setelah perintah query ini.

## CONTOH GAMBAR:

---

```
C:\> Select XAMPP for Windows - mysql -u root -p
MariaDB [(none)]> SHOW DATABASES;
+-----+
| Database |
+-----+
| information_schema |
| mysql |
| performance_schema |
| phpmyadmin |
| rayhanbatman |
| test |
+-----+
6 rows in set (0.002 sec)

MariaDB [(none)]> DROP DATABASE RAYHANBATMAN;
Query OK, 0 rows affected (0.003 sec)

MariaDB [(none)]> SHOW DATABASES;
+-----+
| Database |
+-----+
| information_schema |
| mysql |
| performance_schema |
| phpmyadmin |
| test |
+-----+
5 rows in set (0.001 sec)
```

CONTOH GAMBAR:

```
C:\> Select XAMPP for Windows - mysql -u root -p
MariaDB [(none)]> SHOW DATABASES;
+-----+
| Database |
+-----+
| information_schema |
| mysql |
| performance_schema |
| phpmyadmin |
| rayhanbatman |
| test |
+-----+
6 rows in set (0.002 sec)

MariaDB [(none)]> DROP DATABASE RAYHANBATMAN;
Query OK, 0 rows affected (0.003 sec)

MariaDB [(none)]> SHOW DATABASES;
+-----+
| Database |
+-----+
| information_schema |
| mysql |
| performance_schema |
| phpmyadmin |
| test |
+-----+
5 rows in set (0.001 sec)
```

a. Pada tabel pertama merupakan tampilan pertama tabel berisikan nama nama database sebelum menghapus database dengan nama `rayhanbatman`

b. Pada bagian yang diblok pada gambar adalah sebuah perintah dengan *query* `drop database rayhanbatman;` sebagai perintah mysql untuk menghapus sebuah database dengan nama `rayhanbatman` pada mysql.

c. pada tabel kedua merupakan tampilan tabel yang telah berhasil menghapus database dengan nama `rayhanbatman`.

- ♦ **Menampilkan Database**, dalam perintah menampilkan database pada mysql, diawali dengan query `SHOW DATABASES;` maka perintah mysql akan menampilkan Databases setelah perintah query.

## CONTOH GAMBAR:

```

C:\> Select XAMPP for Windows - mysql -u root -p
MariaDB [(none)]> SHOW DATABASES;
+-----+
| Database |
+-----+
| information_schema |
| mysql |
| performance_schema |
| phpmyadmin |
| test |
+-----+
5 rows in set (0.001 sec)

MariaDB [(none)]> create database rayhanbatman;
Query OK, 1 row affected (0.003 sec)

MariaDB [(none)]>
MariaDB [(none)]> show databases;
+-----+
| Database |
+-----+
| information_schema |
| mysql |
| performance_schema |
| phpmyadmin |
| rayhanbatman |
| test |
+-----+
6 rows in set (0.001 sec)
```

*query*

`SHOW DATABASES;`

- ♦ **Menggunakan Database**,dalam perintah menggunakan database pada mysql,maka perintahnya menggunakan query `USE[NAMA_DATABASE]`

## CONTOH GAMBAR:

---

```
MariaDB [(none)]> use XI_RPL_1;
Database changed
MariaDB [XI_RPL_1]>
```

---

## CONTOH GAMBAR:

```
Ctrl Select XAMPP for Windows - mysql -u root -p
MariaDB [(none)]> SHOW DATABASES;
+-----+
| Database |
+-----+
| information_schema |
| mysql |
| performance_schema |
| phpmyadmin |
| test |
+-----+
5 rows in set (0.001 sec)

MariaDB [(none)]> create database rayhanbatman;
Query OK, 1 row affected (0.003 sec)

MariaDB [(none)]>
MariaDB [(none)]> show databases;
+-----+
| Database |
+-----+
| information_schema |
| mysql |
| performance_schema |
| phpmyadmin |
| rayhanbatman |
| test |
+-----+
6 rows in set (0.001 sec)
```

### query

`SHOW DATABASES;`

- ♦ **Menggunakan Database**,dalam perintah menggunakan database pada mysql,maka perintahnya menggunakan query `USE[NAMA_DATABASE]`

## CONTOH GAMBAR:

```
MariaDB [(none)]> use XI_RPL_1;
Database changed
MariaDB [XI_RPL_1]>
```



pada gambar di atas tertampilkan bahwa `query MariaDB [("none")]` adalah database yang sedang digunakan, kemudian setelah melakukan perintah dengan `query use XI_RPL_1` maka pada tampilan yang diblok dengan `query MariaDB [XI_RPL_1]>` menandakan bahwa user telah berada pada database `XI_RPL_1`

## Tipe Data

---

### Angka

---

- ♦ **int**: Adalah sebuah tipe data yang berfungsi untuk menyimpan data bilangan bulat.
- ♦ **Decimal**: adalah tipe data yang digunakan untuk menyimpan data bilangan desimal.
- ♦ **Float dan Double**: adalah tipe data yang digunakan untuk menyimpan data dengan presisi float. Double memiliki presisi lebih tinggi dibandingkan dengan Float.
- ♦ **Tinyint**, **Smallint**, **Mediumint**, dan **Bigint**: Beberapa tipe data tersebut memiliki fungsi yang sama dengan **int** yaitu menyimpan bilangan bulat, tetapi bedanya tipe data tersebut dapat menyimpan data dengan ukuran yang berbeda-beda. Contoh:

```
CREATE TABLE contoh_table (  
    id INT,  
    harga DECIMAL(10,2),  
    jumlah_barang TINYINT  
);
```

Pada contoh ini, ini adalah query untuk membuat table yang berisikan kolom id dengan bertipe data **int**, kolom harga yang menggunakan tipe data **Decimal** dengan kategori 10 digit dan 2 angka dibelakang koma, dan kolom **jumlah barang** yang menggunakan tipe data **tinyint**.

### Teks

---

- ♦ **CHAR(N)** Menyimpan data string karakter tetap dengan panjang jumlah N yang sudah ditentukan. Contoh: **CHAR(10)** akan menyimpan string dengan panjang tepat 10 karakter.
- ♦ **VARCHAR(N)**: Menyimpan string karakter dengan panjang variabel maksimal N. Misalnya, **VARCHAR(255)** dapat menyimpan string hingga 255 karakter, tetapi sebenarnya hanya menyimpan panjang yang diperlukan plus beberapa overhead.

- ♦ **TEXT**: Digunakan untuk menyimpan teks dengan panjang variabel, tanpa batasan panjang tertentu. Cocok untuk data teks yang panjangnya tidak terduga.
- ♦ **ENUM**: Memungkinkan Anda mendefinisikan set nilai yang mungkin dan membatasi kolom hanya dapat mengambil salah satu dari nilai tersebut.
- ♦ **SET**: Mirip dengan ENUM, namun dapat menyimpan satu atau lebih nilai dari himpunan yang telah ditentukan. Contoh :

```
CREATE TABLE contoh_tabel (
    nama CHAR(50),
    alamat VARCHAR(100),
    catatan TEXT,
    status ENUM('Aktif', 'Non-Aktif')
);
```

## Tanggal

- ♦ **Date** adalah tipe data yang menyimpan nilai tanggal dengan format **yyyy-mm-dd**.
  - ♦ **Time** adalah tipe data yang menyimpan nilai waktu dengan format **hh:mm:ss**.
  - ♦ **Datetime** adalah tipe data yang menyimpan dan menggabungkan nilai tanggal dan juga waktu dengan format **yyyy-mm-dd hh:mm:ss**.
  - ♦ **Timestamp** tipe data ini sama seperti DATETIME, tapi kelebihan nya yaitu dapat diatur otomatis saat data dimasukkan atau diubah
- Contoh:

```
CREATE TABLE ContohTabel (
    tanggal DATE,
    waktu TIME,
    datetimekolom DATETIME,
    timestampkolom TIMESTAMP
);
```

Dalam contoh ini, query tersebut adalah perintah membuat table yang memiliki kolom tanggal dengan tipe data **Date**, kolom waktu betipe data **Time**, kolom datetimekolom memiliki tipe data **Datetime**, dan kolom timestampkolom menggunakan tipe data **Timestamp**.

## Boolean

- ♦ **Boolean** adalah tipe data yang digunakan untuk menyimpan nilai boolean, yang bertujuan mencari kondisi benar atau salah. Representasi nilai benar adalah angka **1** dan nilai salah adalah angka **0**.

Contoh:

```
CREATE TABLE contohTabel (  
    title VARCHAR(255),  
    completed BOOLEAN  
);
```

Dalam contoh diatas, adalah query perintah untuk membuat table dengan memiliki kolom title yang menggunakan tipe data **Varchar** dengan menyimpan nilai string **255** karakter dan kolom completed yang menggunakan tipe data **Boolean** yang dimana kolom tersebut menyimpan nilai TRUE dan FALSE, dan jika direpresentasikan dalam bentuk angka yaitu **1** dan **0**.

## Tabel

---

### Buat tabel

---

Tabel merupakan **susunan data dalam baris dan kolom, atau mungkin dalam struktur yang lebih kompleks**. Tabel juga dapat digunakan pada *Basisdata* untuk membuat data menjadi lebih terstruktur.

```
CREATE TABLE pelanggan (  
    id_pelanggan INT(4) PRIMARY KEY NOT NULL,  
    nama_depan VARCHAR(25) NOT NULL,  
    nama_belakang VARCHAR(25) NOT NULL,  
    no_telp char(12) UNIQUE  
);
```

**id\_pelanggan** bertipe data **INT** untuk menyimpan data berupa bilangan bulat dengan panjang maksimum 4 digit, sebagai kunci utama (**PRIMARY KEY**) yang tidak boleh kosong (**NOT NULL**).

**nama\_depan** bertipe data **VARCHAR** untuk menyimpan data berupa karakter dengan panjang maksimum 25 karakter, tidak boleh kosong (**NOT NULL**).

**nama\_belakang** bertipe data **VARCHAR** untuk menyimpan data berupa karakter dengan

panjang maksimum 25 karakter, tidak boleh kosong (**NOT NULL**).

**no\_telp** bertipe data **CHAR** untuk menyimpan data berupa karakter (berbeda dengan VARCHAR tipe data ini menyimpan data terlepas dari panjang string yang sebenarnya) dengan panjang maksimum 12 karakter, dan harus unik (**UNIQUE**).

#### Info >

Setelah memasukkan query di atas, tekan ENTER lalu ketik **DESC [NAMA TABEL]**, kemudian tekan ENTER.

## tampilan struktur tabel

```
1 row in set (0.118 sec)

MariaDB [rental_hayril]> desc pelanggan;
+-----+-----+-----+-----+-----+-----+
| Field          | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| id             | int(4)        | NO   | PRI | NULL    |       |
| nama_depan     | varchar(25)   | NO   |     | NULL    |       |
| nama_belakang  | varchar(25)   | NO   |     | NULL    |       |
| no_tlp         | char(12)      | YES  | UNI | NULL    |       |
+-----+-----+-----+-----+-----+-----+
4 rows in set (0.337 sec)
```

- ♦ **id** adalah nama data base atau variabel untuk mengisi **id**, **INT(4)** adalah perintah untuk menambahkan data id pelanggan dengan menggunakan tipe data **INT** untuk menyimpan data berupa angka, dan angka 4 adalah jumlah data yang hanya dapat ditambahkan. **PRIMARY KEY NOT NULL** adalah perintah untuk membuat data menjadi terkunci yang artinya data tersebut wajib terisi.
- ♦ **nama\_depan** merupakan variabel untuk menambahkan data dengan inisialisasi **nama\_depan** dan **VARCHAR** sebagai tipe data untuk menyimpan data berupa karakter dan (25) adalah sebuah keterangan yang hanya dapat berupa 25 data, **NOT NULL** adalah sebagai perintah agar menegaskan bahwa data tersebut bisa saja tidak terisi.
- ♦ sama seperti nama depan tadi **nama\_belakang** merupakan variabel untuk menambahkan data dengan inisialisasi **nama\_belakang** dan **VARCHAR** sebagai tipe data untuk menyimpan data berupa karakter dan (25) adalah sebuah keterangan yang hanya dapat berupa 25 data, **NOT NULL** adalah sebagai perintah agar menegaskan bahwa data tersebut bisa saja tidak terisi.
- ♦ pada struktur **no\_telp** menggunakan variabel **no\_telp** sebagai tempat menyimpan data nomor telepon, **char(12)** tipe data char sebagai tempat menyimpan data berupa karakter, kemudian (12) sebagai keterangan bahwa data yang dapat diisi hanya dapat

berjumlah 12 data,dan **UNIQUE** adalah sebuah perintah bahwa data tersebut tidak boleh sama dengan data yang lainnya.

## menampilkan tabel

---

```
XAMPP for Windows - mysql -u root -p
MariaDB [rental_hayril]> show tables;
+-----+
| Tables_in_rental_hayril |
+-----+
| pelanggan               |
+-----+
1 row in set (0.118 sec)
```

**show tables;**

*query*

## QnA

---

1. Mengapa hanya kolom id\_pelanggan yang menggunakan constraint PRIMARY KEY?
2. Mengapa pada kolom no\_telp yang menggunakan data CHAR bukan VARCHAR?
3. Mengapa hanya kolom no\_telp yang menggunakan constraint UNIQUE?
4. Mengapa kolom no\_telp tidak memakai constraint NOT NULL sementara kolom lainnya menggunakan constraint tersebut?
5. Tuliskan perbedaan antara PRIMARY KEY dengan UNIQUE?

JAWABAN:

6. .Untuk membedakan id Pelanggan yang sama, mencegah duplikasi, dan mempermudah pencarian data.
7. Tipe data char menyimpan data dalam karakter panjang lebih efisien. pencarian pada kolom tipe data CHAR dapat lebih cepat
8. Karna no\_telp tidak ada yang sama semua pasti berbeda dan nilainya unik maka menggunakan constraints unique artinya data dalam tabel id\_telpon berbeda tidak ada yang sama.
9. Nomor telpon dianggap opsional. nomor telepon hanya menjadi wajib saat pengguna melakukan langkah-langkah tertentu, Anda mungkin tidak ingin mengharuskan

pengguna mengisinya pada tahap awal.

10. PRIMARY KEY untuk membedakan data yang sama dan hanya boleh 1 dan tidak boleh tidak ada.

Kalau UNIQUE sebuah kolom yang memiliki data yang berbeda atau tidak sama unique boleh 1,2,3 Dan seterusnya dan boleh tidak ada.

## INSERT

---

### INSERT 1 DATA

---

#### STRUKTUR

```
INSERT INTO [NAMA_TABEL]
VALUES (nilai1,nilai2,nilai3);
```

#### CONTOH

```
INSERT INTO pelanggan
VALUES (101,"hayril","abizali",0895710530505)
```

#### ANALISA:

**INSERT INTO** adalah perintah untuk memasukkan data kedalam tabel dengan nama **pelanggan** sebagai inisialisasi nama tabel yang ingin dimasukkan data. **VALUE** sebagai penentu nilai yang akan dimasukkan, angka *101* sebagai nilai untuk mengisi kolom pertama pada tabel **pelanggan** sebagai ID pelanggan, "*hayril*" sebagai data untuk mengisi tabel nama depan pada tabel **pelanggan**, "*abizali*" sebagai data untuk mengisi tabel nama belakang pada tabel **pelanggan**, "*0895710530505*" adalah data untuk mengisi tabel no telpon pada tabel **pelanggan**.

```
MariaDB [rental_hayril]> insert into pelanggan
-> values (101,"hayril","abizali",083012968947);
Query OK, 1 row affected (0.439 sec)

MariaDB [rental_hayril]>
```

### INSERT >1 DATA

---

#### STRUKTUR

```
INSERT INTO pelanggan
VALUES (101,"hayril","abizali",0895710530505),
      (102,"abd","rahman",0879634657909),
      (103,"rayhan","batman",0895477970909),
      (104,"fachri","ironman",0876542325675);
```

CONTOH:

```
Query OK, 1 row affected (0.439 sec)

MariaDB [rental hayril]> insert into pelanggan
-> values (102,"abd","rahman",0879634657909),
-> (103,"rayhan","batman",0895477970909), (104,"fachri","ironman",0876542325675);
Query OK, 3 rows affected (0.155 sec)
```

## SELECT

---

### SELECT SELURUH DATA

---

### STRUKTUR

---

```
SELECT * FROM [nama_table];
```

CONTOH:

---

```
SELECT * FROM pelanggan;
```

### HASIL

---

```

MariaDB [rental_hayril]> select * from pelanggan;
+-----+-----+-----+-----+
| id    | nama_depan | nama_belakang | no_tlp    |
+-----+-----+-----+-----+
| 101   | hayril     | abizali       | 83012968947 |
| 102   | abd        | rahman        | 879634657909 |
| 103   | rayhan     | batman        | 895477970909 |
| 104   | fachri     | ironman       | 876542325675 |
| 1234  | chairil    | abizali       | 895710530505 |
| 2231  | abd        | rahman        | 12346789101  |
| 2331  | nur        | rahmat        | 87654321231  |
| 4321  | rayhan     | batman        | 85341116306  |
+-----+-----+-----+-----+
8 rows in set (0.251 sec)

MariaDB [rental_hayril]>

```

## ANALISA:

- ◆ **SELECT** perintah untuk menampilkan data.
- ◆ **\*** adalah perintah untuk keseluruhan.
- ◆ **from** adalah perintah untuk pemanggilan dari tabel.
- ◆ **pelanggan** berfungsi untuk sebagai nama tabel yang dipanggil.

KESIMPULAN:

- ◆ **SELECT \* FROM pelanggan;** adalah query perintah untuk menampilkan data table secara keseluruhan.

## SELECT menyebutkan kolom

## STRUKTUR

```
SELECT [nama_kolom] from [nama_table];
```

## CONTOH

```
SELECT nama_depan from pelanggan;
```



```

MariaDB [rental_hayril]> select nama_depan from pelanggan;
+-----+
| nama_depan |
+-----+
| hayril     |
| abd        |
| rayhan     |
| fachri     |
| chairil    |
| abd        |
| nur        |
| rayhan     |
+-----+
8 rows in set (0.001 sec)

MariaDB [rental_hayril]>

```

## ANALISA:

- ◆ **SELECT** perintah untuk menampilkan data.
- ◆ **nama\_depan** adalah nama dari salah satu kolom pada tabel
- ◆ **from** adalah perintah untuk pemanggilan dari tabel.
- ◆ **pelanggan** berfungsi untuk sebagai nama tabel yang dipanggil.

## KESIMPULAN:

- ◆ SELECT nama\_depan FROM pelanggan; adalah query perintah untuk menampilkan seluruh data tabel pada kolom nama depan .

## KLAUSA WHERE

## STRUKTUR

```
SELECT [nama_kolom/] from [nama_table] where [kondisi];
```

## CONTOH

```
SELECT nama_depan from pelanggan where ID=104;
```

## HASIL

```
MariaDB [rental_hayril]> SELECT nama_depan from pelanggan where ID=104;
+-----+
| nama_depan |
+-----+
| fachri     |
+-----+
1 row in set (0.105 sec)

MariaDB [rental_hayril]>
```

## ANALISA:

---

- ♦ **SELECT** perintah untuk menampilkan data.
- ♦ **nama\_depan** adalah salah satu nama kolom pada tabel.
- ♦ **from** adalah perintah untuk pemanggilan dari tabel.
- ♦ **pelanggan** sebagai nama tabel yang ditujukan.
- ♦ **where** perintah yang berfungsi menentukan letak data yang dipilih.
- ♦ **ID=104** adalah letak data yang ingin di tampilkan.

## KESIMPULAN:

---

- ♦ **SELECT nama\_depan from pelanggan where ID=104;** adalah query perintah untuk menampilkan data tabel pada kolom nama depan pada data yang id 104.

## UPDATE

---

### STRUKTUR:

---

```
update [nama_tabel] set [nama_kolom]="data yang diubah" where
kodisi;
```

### CONTOH

---

```
update pelanggan set nama_depan="latto" where id=101;
```

## HASIL

---

```

MariaDB [rental_hayril]> select * from pelanggan;
+-----+-----+-----+-----+
| id   | nama_depan | nama_belakang | no_tlp |
+-----+-----+-----+-----+
| 101  | hayril    | abizali      | 83012968947 |
| 102  | abd       | rahman       | 879634657909 |
| 103  | rayhan    | batman       | 895477970909 |
| 104  | fachri    | ironman      | 876542325675 |
| 1234 | chairil   | abizali      | 895710530505 |
| 2231 | sijagaiki | sikatutui    | 12346789101  |
| 2331 | megumi    | rahmat       | 87654321231  |
+-----+-----+-----+-----+
7 rows in set (0.001 sec)

MariaDB [rental_hayril]> update pelanggan set nama_belakang="latto" where id=101;
Query OK, 1 row affected (0.043 sec)
Rows matched: 1  Changed: 1  Warnings: 0

MariaDB [rental_hayril]> select * from pelanggan;
+-----+-----+-----+-----+
| id   | nama_depan | nama_belakang | no_tlp |
+-----+-----+-----+-----+
| 101  | hayril    | latto        | 83012968947 |
| 102  | abd       | rahman       | 879634657909 |
| 103  | rayhan    | batman       | 895477970909 |
| 104  | fachri    | ironman      | 876542325675 |
| 1234 | chairil   | abizali      | 895710530505 |
| 2231 | sijagaiki | sikatutui    | 12346789101  |
| 2331 | megumi    | rahmat       | 87654321231  |
+-----+-----+-----+-----+
7 rows in set (0.001 sec)

MariaDB [rental_hayril]> 

```

## ANALISA:

- ♦ **update** adalah perintah query untuk melakukan update atau pembaharuan data.
- ♦ **pelanggan** sebagai nama tabel yang dipanggil.
- ♦ **set** adalah perintah untuk setting
- ♦ **nama\_depan** adalah sebagai nama kolom.
- ♦ **where** perintah yang berfungsi menentukan letak data yang dipilih.
- ♦ **ID=104** adalah letak data yang ingin dipilih.

## KESIMPULAN:

- ♦ Update pelanggan set nama\_depan="latto" where id=101; adalah perintah untuk memperbaharui data ber-id=101 pada kolom nama\_depan dengan data yang baru berupa "latto".

## DELETE

## STRUKTUR:

```
delete from pelanggan where kondisi;
```

## CONTOH:

```
delete from pelanggan where id=101;
```

## HASIL

```
MariaDB [rental_hayril]> select * from pelanggan;
+-----+-----+-----+-----+
| id    | nama_depan | nama_belakang | no_tlp    |
+-----+-----+-----+-----+
| 101   | hayril     | latto         | 83012968947 |
| 102   | abd        | rahman        | 879634657909 |
| 103   | rayhan     | batman        | 895477970909 |
| 104   | fachri     | ironman       | 876542325675 |
| 1234  | chairil    | abizali       | 895710530505 |
| 2231  | sijagaiki  | sikatutui     | 12346789101  |
| 2331  | megumi     | rahmat        | 87654321231  |
+-----+-----+-----+-----+
7 rows in set (0.001 sec)

MariaDB [rental_hayril]> delete from pelanggan where id=101;
Query OK, 1 row affected (0.099 sec)

MariaDB [rental_hayril]> select * from pelanggan;
+-----+-----+-----+-----+
| id    | nama_depan | nama_belakang | no_tlp    |
+-----+-----+-----+-----+
| 102   | abd        | rahman        | 879634657909 |
| 103   | rayhan     | batman        | 895477970909 |
| 104   | fachri     | ironman       | 876542325675 |
| 1234  | chairil    | abizali       | 895710530505 |
| 2231  | sijagaiki  | sikatutui     | 12346789101  |
| 2331  | megumi     | rahmat        | 87654321231  |
+-----+-----+-----+-----+
6 rows in set (0.001 sec)

MariaDB [rental_hayril]> _
```

## ANALISA:

- ♦ **DELETE** adalah perintah query yang berfungsi untuk menghapus data;
- ♦ **from** adalah perintah untuk pemanggilan dari tabel.
- ♦ **pelanggan** sebagai nama tabel yang ditujukan.
- ♦ **where** perintah yang berfungsi menentukan letak data yang dipilih.
- ♦ **ID=101** adalah letak data yang ingin dipilih.

## KESIMPULAN:

- ♦ delete from pelanggan where id=101; adalah perintah query untuk menghapus seluruh data yang ber-id=101.

## HAPUS TABEL

---

### STRUKTUR:

---

```
DROP TABLE [nama_table];
```

### CONTOH:

---

```
DROP TABLE pelanggan;
```

```
C:\> XAMPP for Windows - mysql -u root -p
MariaDB [rental_hayril]> show tables;
+-----+
| Tables_in_rental_hayril |
+-----+
| pelanggan               |
+-----+
1 row in set (0.040 sec)

MariaDB [rental_hayril]> DROP TABLE pelanggan;
Query OK, 0 rows affected (0.270 sec)

MariaDB [rental_hayril]> show tables;
Empty set (0.001 sec)

MariaDB [rental_hayril]>
```

### ANALISA:

---

- ♦ **DROP TABLE** adalah perintah untuk menghapus.
- ♦ **pelanggan** adalah sebuah nama tabel.

### KESIMPULAN:

---

- ♦ DROP TABLE pelanggan; adalah sebuah perintah untuk menghapus sebuah tabel yang bernama **pelanggan**.

## SELECT LANJUTAN

---

# AND

---

select AND ini akan menampilkan data dengan "nilai1" dan "nilai2".

## CONTOH:

---

```
select warna,pemilik FROM mobil WHERE warna="hitam" AND
pemilik="ibrahim";
```

## HASIL

---

```
MariaDB [rental_hayril]> select * from mobil;
+-----+-----+-----+-----+-----+-----+-----+
| id_pelanggan | no_plat | no_mesin | warna | pemilik | peminjam | harga_rental |
+-----+-----+-----+-----+-----+-----+-----+
| 1 | DD 2650 XY | ACX3568 | Hitam | Ibrahim | Afdal | 50000 |
| 2 | DD 2440 AX | BCS1120 | Merah | Ibrahim | Elia | 100000 |
| 3 | B 1611 QC | LSQ1112 | Silver | Baim | Anty | 50000 |
| 4 | DD 2901 JK | UQL1029 | Hitam | Ibe | NULL | 150000 |
| 5 | DD 2210 LS | CJH1011 | Hitam | Ibe | NULL | 100000 |
+-----+-----+-----+-----+-----+-----+-----+
5 rows in set (0.001 sec)

MariaDB [rental_hayril]> select warna,pemilik FROM mobil WHERE warna="hitam" AND pemilik="ibrahim";
+-----+-----+
| warna | pemilik |
+-----+-----+
| Hitam | Ibrahim |
+-----+-----+
1 row in set (0.001 sec)

MariaDB [rental_hayril]> _
```

# OR

---

select OR ini akan menampilkan data dengan "nilai1" atau "nilai2".

## CONTOH:

---

```
select warna,pemilik FROM mobil WHERE warna="hitam" OR
pemilik="ibrahim";
```

<<<<<< HEAD

## HASIL

---

```
C:\Windows\system32\cmd.exe - mysql -u root -p
MariaDB [rental_hayril]> select * from mobil;
+-----+-----+-----+-----+-----+-----+-----+
| id_pelanggan | no_plat   | no_mesin | warna | pemilik | peminjam | harga_rental |
+-----+-----+-----+-----+-----+-----+-----+
| 1 | DD 2650 XY | ACX3568 | Hitam | Ibrahim | Afdal    | 50000 |
| 2 | DD 2440 AX | BCS1120 | Merah | Ibrahim | Elia     | 100000 |
| 3 | B 1611 QC  | LSQ1112 | Silver | Baime   | Anty     | 50000 |
| 4 | DD 2901 JK | UQL1029 | Hitam | Ibe     | NULL     | 150000 |
| 5 | DD 2210 LS | CJH1011 | Hitam | Ibe     | NULL     | 100000 |
+-----+-----+-----+-----+-----+-----+-----+
5 rows in set (0.001 sec)

MariaDB [rental_hayril]> select warna,pemilik from mobil where warna="hitam" or pemilik="ibrahim";
+-----+-----+
| warna | pemilik |
+-----+-----+
| Hitam | Ibrahim |
| Hitam | Ibe     |
| Hitam | Ibe     |
+-----+-----+
3 rows in set (0.001 sec)

MariaDB [rental_hayril]> _
```

```
C:\Windows\system32\cmd.exe - mysql -u root -p
MariaDB [rental_hayril]> select * from mobil;
+-----+-----+-----+-----+-----+-----+-----+
| id_pelanggan | no_plat   | no_mesin | warna | pemilik | peminjam | harga_rental |
+-----+-----+-----+-----+-----+-----+-----+
| 1 | DD 2650 XY | ACX3568 | Hitam | Ibrahim | Afdal    | 50000 |
| 2 | DD 2440 AX | BCS1120 | Merah | Ibrahim | Elia     | 100000 |
| 3 | B 1611 QC  | LSQ1112 | Silver | Baime   | Anty     | 50000 |
| 4 | DD 2901 JK | UQL1029 | Hitam | Ibe     | NULL     | 150000 |
| 5 | DD 2210 LS | CJH1011 | Hitam | Ibe     | NULL     | 100000 |
+-----+-----+-----+-----+-----+-----+-----+
5 rows in set (0.001 sec)

MariaDB [rental_hayril]> select warna,pemilik from mobil where warna="hitam" or pemilik="ibrahim";
+-----+-----+
| warna | pemilik |
+-----+-----+
| Hitam | Ibrahim |
| Hitam | Ibe     |
| Hitam | Ibe     |
+-----+-----+
3 rows in set (0.001 sec)

MariaDB [rental_hayril]> _
```

||||| 1460ac171d437d06757ab7998eb78b7cac7da75b

## BETWEEN- AND

select between-and ini akan menampilkan data antara "nilai1" dan "nilai2".karena didukung dengan AND.

## CONTOH:

---

```
select * FROM mobil WHERE harga_rental BETWEEN 100000 AND 200000;
```

## HASIL

---

```
3 rows in set (0.001 sec)

MariaDB [rental_hayril]> SELECT * FROM mobil WHERE harga_rental BETWEEN 100000 AND 200000;
+-----+-----+-----+-----+-----+-----+-----+
| id_pelanggan | no_plat | no_mesin | warna | pemilik | peminjam | harga_rental |
+-----+-----+-----+-----+-----+-----+-----+
| 2 | DD 2440 AX | BCS1120 | Merah | Ibrahim | Elia | 100000 |
| 4 | DD 2901 JK | UQL1029 | Hitam | Ibe | NULL | 150000 |
| 5 | DD 2210 LS | CJH1011 | Hitam | Ibe | NULL | 100000 |
+-----+-----+-----+-----+-----+-----+-----+
3 rows in set (0.001 sec)

MariaDB [rental_hayril]>
```

## NOT BETWEEN

---

untuk NOT BETWEEN ini akan menampilkan yang bukan antara "nilai1" dan "nilai2".

## CONTOH:

---

```
SELECT * FROM mobil WHERE harga_rental NOT BETWEEN 100000 AND 200000;
```

## HASIL

---



```
C:\Windows\system32\cmd.exe - mysql -u root -p

+-----+-----+-----+-----+-----+-----+-----+
| 1 | DD 2650 XY | ACX3568 | Hitam | Ibrahim | Afdal | 50000 |
| 2 | DD 2440 AX | BC51120 | Merah | Ibrahim | Elia | 100000 |
| 3 | B 1611 QC | LSQ1112 | Silver | Baim | Anty | 50000 |
| 4 | DD 2901 JK | UQL1029 | Hitam | Ibe | NULL | 150000 |
| 5 | DD 2210 LS | CJH1011 | Hitam | Ibe | NULL | 100000 |
+-----+-----+-----+-----+-----+-----+-----+
5 rows in set (0.160 sec)

MariaDB [rental_hayril]> select * from mobil where harga_rental not between 100000 and 200000;
+-----+-----+-----+-----+-----+-----+-----+
| id_pelanggan | no_plat | no_mesin | warna | pemilik | peminjam | harga_rental |
+-----+-----+-----+-----+-----+-----+-----+
| 1 | DD 2650 XY | ACX3568 | Hitam | Ibrahim | Afdal | 50000 |
| 3 | B 1611 QC | LSQ1112 | Silver | Baim | Anty | 50000 |
+-----+-----+-----+-----+-----+-----+-----+
2 rows in set (0.056 sec)

MariaDB [rental_hayril]>
```

**<=**

untuk **<=** ini akan menampilkan "**data**" yang lebih kecil atau sama dengan "**nilai\_data**" yang telah ditentukan.

## CONTOH:

```
SELECT * FROM mobil WHERE harga_rental ≤ 50000;
```

## HASIL

```
MariaDB [rental_hayril]> SELECT * FROM mobil WHERE harga_rental <= 50000;
+-----+-----+-----+-----+-----+-----+-----+
| id_pelanggan | no_plat | no_mesin | warna | pemilik | peminjam | harga_rental |
+-----+-----+-----+-----+-----+-----+-----+
| 1 | DD 2650 XY | ACX3568 | Hitam | Ibrahim | Afdal | 50000 |
| 3 | B 1611 QC | LSQ1112 | Silver | Baim | Anty | 50000 |
+-----+-----+-----+-----+-----+-----+-----+
2 rows in set (0.044 sec)

MariaDB [rental_hayril]>
```

**>=**

untuk **>=** ini akan menampilkan "**data**" yang lebih besar atau sama dengan "**nilai\_data**" yang telah ditentukan.

## CONTOH:

```
SELECT * FROM mobil WHERE harga_rental ≥ 50000;
```

## HASIL

```
MariaDB [rental_hayril]> SELECT * FROM mobil WHERE harga_rental >= 50000;
+-----+-----+-----+-----+-----+-----+-----+
| id_pelanggan | no_plat | no_mesin | warna | pemilik | peminjam | harga_rental |
+-----+-----+-----+-----+-----+-----+-----+
| 1 | DD 2650 XY | ACX3568 | Hitam | Ibrahim | Afdal | 50000 |
| 2 | DD 2440 AX | BCS1120 | Merah | Ibrahim | Elia | 100000 |
| 3 | B 1611 QC | LSQ1112 | Silver | Baim | Anty | 50000 |
| 4 | DD 2901 JK | UQL1029 | Hitam | Ibe | NULL | 150000 |
| 5 | DD 2210 LS | CJH1011 | Hitam | Ibe | NULL | 100000 |
+-----+-----+-----+-----+-----+-----+-----+
5 rows in set (0.001 sec)

MariaDB [rental_hayril]>
```

## <> atau !=

untuk <> atau != ini akan menampilkan "**data**" yang tidak sama dengan "**nilai\_data**" yang telah ditentukan.

## CONTOH:

```
SELECT * FROM mobil WHERE harga_rental <> 50000;
```

## HASIL

```
C:\Windows\system32\cmd.exe - mysql -u root -p
MariaDB [rental_hayril]> SELECT * FROM mobil WHERE harga_rental <> 50000;
+-----+-----+-----+-----+-----+-----+-----+
| id_pelanggan | no_plat | no_mesin | warna | pemilik | peminjam | harga_rental |
+-----+-----+-----+-----+-----+-----+-----+
| 2 | DD 2440 AX | BCS1120 | Merah | Ibrahim | Elia | 100000 |
| 4 | DD 2901 JK | UQL1029 | Hitam | Ibe | NULL | 150000 |
| 5 | DD 2210 LS | CJH1011 | Hitam | Ibe | NULL | 100000 |
+-----+-----+-----+-----+-----+-----+-----+
3 rows in set (0.052 sec)

MariaDB [rental_hayril]> _
```