

## SELECT LANJUTAN + GROUP BY DAN HAVING + RELASI

### SELECT LANJUTAN

AND

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

STRUKTUR:

```
select [nama_kolom],[nama_kolom] FROM [nama_tabel] WHERE [nama_kolom]=["nama_data"] AND [nama_kolom]=["nama_data"];
```

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]> ■
```

ANALISIS:

- **SELECT** adalah sebuah perintah query yang tugas memilih.
- **WARNA** adalah sebuah nama kolom pada tabel mobil.
- **PEMILIK** adalah sebuah nama kolom juga pada tabel mobil.
- **FROM** adalah sebuah perintah query yang tugas nya untuk pemanggilan dari.
- **WHERE** adalah sebuah perintah query yang tugas untuk menentukan pemanggilan yang mana ditujukannya.
- **hitam** adalah nama data pada sebuah kolom
- **AND** adalah sebuah perintah query DAN menujukkan bahwa pemilihan data nya lebih dari 1.
- **ibrahim** adalah nama data pada sebuah kolom

KESIMPULAN:

query diatas adalah perintah untuk menampilkan sebuah data dari tabel mobil yang bertujuan pada kolom warna dengan data "hitam" dan kolom pemilik dengan data "ibrahim"

## OR

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

```
select [nama_kolom],[nama_kolom] FROM [nama_tabel] WHERE [nama_kolom]=["nama_data"] OR  
[nama_kolom]=["nama_data"];
```

CONTOH:

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

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 | 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" or pemilik="ibrahim";  
+-----+-----+  
| warna | pemilik |  
+-----+-----+  
| Hitam | Ibrahim |  
| Hitam | Ibe |  
| Hitam | Ibe |  
+-----+-----+  
3 rows in set (0.001 sec)  
  
MariaDB [rental_hayril]> ■
```

ANALISIS:

- **SELECT** adalah sebuah perintah query yang tugas memilih.
- **WARNA** adalah sebuah nama kolom pada tabel mobil.
- **PEMILIK** adalah sebuah nama kolom juga pada tabel mobil.
- **FROM** adalah sebuah perintah query yang tugas nya untuk pemanggilan dari.
- **WHERE** adalah sebuah perintah query yang tugas untuk menentukan pemanggilan yang mana ditujukannya.
- **hitam** adalah nama data pada sebuah kolom
- **OR** adalah sebuah perintah query ATAU menujukkan bahwa pemilihan data yang ditampilkan adalah salah satu dari syarat yang ditentukan.
- **ibrahim** adalah nama data pada sebuah kolom

KESIMPULAN:

query diatas adalah perintah untuk menampilkan sebuah data dari tabel mobil yang bertujuan pada

kolom warna dengan data "hitam" atau kolom pemilik dengan data "ibrahim" jadi hanya data hitam dan ibrahim lah yang akan tampil.

## BETWEEN- AND

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

```
select * FROM [nama_tabel] WHERE [nama_kolom] BETWEEN [nilai_data1] AND [nilai_data2];
```

CONTOH:

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

HASIL:

```
+----+----+----+----+----+----+----+
| 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]>
```

ANALISIS:

- **SELECT** adalah sebuah perintah query yang tugas memilih.
- **FROM** adalah sebuah perintah query yang tugas nya untuk pemanggilan dari.
- **WHERE** adalah sebuah perintah query yang tugas untuk menentukan pemanggilan yang mana ditujukannya.
- **MOBIL** adalah sebuah nama tabel.
- **BETWEEN** adalah sebuah perintah query ANTARA menujukkan bahwa pemilihan data yang ditampilkan adalah DI ANTARA salah satu data dari syarat yang ditentukan.
- **AND** adalah sebuah perintah query DAN menujukkan bahwa pemilihan data nya lebih dari 1.
- **harga\_rental** adalah sebuah nama kolom.
- **100000** adalah nama data pada sebuah kolom.
- **200000** adalah nama data pada sebuah kolom.

KESIMPULAN:

query diatas adalah perintah untuk menampilkan sebuah data dari tabel mobil yang bertujuan pada kolom **harga rental** dengan data antara "100000" dan data "200000" jadi hanya data dengan jumlah 100000 sampai dengan 200000 yang akan tampil.

## NOT BETWEEN

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

STRUKTUR:

```
SELECT * FROM [nama_tabel] WHERE [nama_kolom] NOT BETWEEN [nama_data] AND [nama_data];
```

CONTOH:

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

HASIL:

```
C:\Windows\system32\cmd.exe - mysql -u root -p
+-----+-----+-----+-----+-----+-----+-----+
| 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 (0.160 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]>
```

ANALISIS:

- **SELECT** adalah sebuah perintah query yang tugas memilih.
- **FROM** adalah sebuah perintah query yang tugas nya untuk pemanggilan dari.
- **WHERE** adalah sebuah perintah query yang tugas untuk menentukan pemanggilan yang mana ditujukannya.
- **MOBIL** adalah sebuah nama tabel.
- **NOT BETWEEN** adalah sebuah perintah query ANTARA menujukkan bahwa pemilihan data yang ditampilkan adalah TIDAK DI ANTARA salah satu data dari syarat yang ditentukan.
- **AND** adalah sebuah perintah query DAN menujukkan bahwa pemilihan data nya lebih dari 1.
- **harga\_rental** adalah sebuah nama kolom.
- **100000** adalah nama data pada sebuah kolom.
- **200000** adalah nama data pada sebuah kolom.

KESIMPULAN:

query diatas adalah perintah untuk menampilkan sebuah data dari tabel mobil yang bertujuan pada kolom **harga rental** dengan data tidak di antara "100000" dan data "200000" jadi hanya data dengan jumlah **100000** sampai dengan **200000** yang tidak akan tampil.

<=

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

STRUKTUR:

```
SELECT * FROM [nama_tabel] WHERE [nama_kolom] <= [nama_data];
```

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]>
```

ANALISIS:

- **SELECT** adalah sebuah perintah query yang tugas memilih.
- **FROM** adalah sebuah perintah query yang tugas nya untuk pemanggilan dari.
- **WHERE** adalah sebuah perintah query yang tugas untuk menentukan pemanggilan yang mana ditujukannya.
- **MOBIL** adalah sebuah nama tabel.
- **harga\_rental** adalah sebuah nama kolom.
- **500000** adalah nama data pada sebuah kolom.
- **<=** adalah perintah query yang betugas menentukan syarat *kurang dari atau sama dengan*.

KESIMPULAN:

query diatas adalah perintah untuk menampilkan sebuah data dari tabel mobil yang bertujuan pada kolom *harga rental* dengan data *KURANG DARI ATAU SAMA DENGAN 500000* jadi hanya data dengan jumlah kurang dari 500000 atau setara dengan 500000 yang akan ditampilkan.

**>=**

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

STRUKTUR:

```
SELECT * FROM [nama_tabel] WHERE [nama_kolom] >= [nama_data];
```

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]>

```

#### ANALISIS:

- **SELECT** adalah sebuah perintah query yang tugas memilih.
- **FROM** adalah sebuah perintah query yang tugas nya untuk pemanggilan dari.
- **WHERE** adalah sebuah perintah query yang tugas untuk menentukan pemanggilan yang mana ditujukannya.
- **MOBIL** adalah sebuah nama tabel.
- **harga\_rental** adalah sebuah nama kolom.
- **500000** adalah nama data pada sebuah kolom.
- **>=** adalah perintah query yang betugas menentukan syarat *lebih dari atau sama dengan*.

#### KESIMPULAN:

query diatas adalah perintah untuk menampilkan sebuah data dari tabel mobil yang bertujuhan pada kolom *harga rental* dengan data *LEBIH DARI ATAU SAMA DENGAN 500000* jadi hanya data dengan jumlah lebih dari 50000 atau setara dengan 50000 yang akan ditampilkan.

**<> atau !=**

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

#### STRUKTUR:

```
SELECT * FROM [nama_tabel] WHERE [nama_kolom] <> [nama_data];
```

#### 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]> -
```

ANALISIS:

- **SELECT** adalah sebuah perintah query yang tugas memilih.
- **FROM** adalah sebuah perintah query yang tugas nya untuk pemanggilan dari .
- **WHERE** adalah sebuah perintah query yang tugas untuk menentukan pemanggilan yang mana ditujukannya.
- **MOBIL** adalah sebuah nama tabel .
- **harga\_rental** adalah sebuah nama kolom .
- **500000** adalah nama data pada sebuah kolom .
- **<>** adalah perintah query yang betugas menentukan syarat *kurang dari atau lebih dari* .

KESIMPULAN: query diatas adalah perintah untuk menampilkan sebuah data dari tabel mobil yang bertujuan pada kolom *harga rental* dengan data \*LEBIH DARI ATAU KURANG DARI 500000 jadi hanya data dengan jumlah lebih dari 50000 atau kurang dari 500000 yang akan ditampilkan ( selain dari nilai 500000 yang akan ditampilkan).

## Tantangan

Untuk tantangan saya akan mengambil nama pemilik "baim" dengan cara memanggilnya dengan syarat nomor pelatnya yaitu "B 1611 QC" lalu hasilnya akan seperti berikut :

```
MariaDB [rental_hayril]> SELECT * FROM akun;
+-----+-----+-----+-----+
| id | nama | username | password |
+-----+-----+-----+-----+
|  1 | Githa | admin    | 12345  |
|  2 | Rani  | user     | 67890  |
|  3 | Qolby | new_user | 54321  |
+-----+-----+-----+
3 rows in set (0.002 sec)

MariaDB [rental_hayril]> SELECT nama FROM akun WHERE password="12345";
+-----+
| nama |
+-----+
| Githa |
+-----+
1 row in set (0.002 sec)
```

IN

STRUKTUR:

```
select * from nama_tabel where nama_kolom in ( "nama_data", "nama_data");
```

CONTOH:

```
select * from data_mobil where warna in ( "silver", "merah");
```

HASIL:

```
MariaDB [rental_hayril]> select * from data_mobil;
+-----+-----+-----+-----+-----+-----+
| id_mobil | 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 |
| 6 | DD 2812 UY | UYN1312 | HITAM | RAYHAN | NULL | 250000 |
+-----+-----+-----+-----+-----+-----+
6 rows in set (0.001 sec)
```

```
MariaDB [rental_hayril]> select * from data_mobil where warna in ("silver", "merah");
+-----+-----+-----+-----+-----+-----+
| id_mobil | no_plat | no_mesin | warna | pemilik | peminjam | harga_rental |
+-----+-----+-----+-----+-----+-----+
| 2 | DD 2440 AX | BCS1120 | Merah | Ibrahim | Elia | 100000 |
| 3 | B 1611 QC | LSQ1112 | Silver | Baim | Anty | 50000 |
+-----+-----+-----+-----+-----+-----+
2 rows in set (0.001 sec)
```

```
MariaDB [rental_hayril]> -
```

ANALISIS:

- **SELECT** adalah sebuah perintah query yang tugas memilih.
- **FROM** adalah sebuah perintah query yang tugas nya untuk pemanggilan dari.
- **WHERE** adalah sebuah perintah query yang tugas untuk menentukan pemanggilan yang mana ditujukannya.
- **DATA\_MOBIL** adalah sebuah nama tabel.
- **WARNA** adalah sebuah nama kolom.
- **SILVER DAN MERAH** adalah nama data pada sebuah kolom.
- **IN** adalah perintah query yang betugas menentukan seluruh data yang sesuai dengan kondisi ditentukan, menghindari penulisan berulang query dengan kondisi yang sama.

KESIMPULAN: query diatas adalah perintah untuk menampilkan sebuah data dari tabel data mobil yang bertujuan pada kolom **warna** dengan data yang hanya **silver dan merah** jadi hanya data silver dan merah yang akan ditampilkan ( tanpa menggunakan query dengan kondisi yang sama).

## IN+AND

STRUKTUR:

```
select * from data_mobil
-> where nama_kolom IN ( "nama_data", "nama_data")
-> AND nama_kolom = nama_data;
```

CONTOH:

```
select * from data_mobil
-> where warna IN ( "hitam","silver")
-> AND harga_rental = 50000;
```

HASIL:

```
1 row in set (0.103 sec)

MariaDB [rental_hayrill]> select * from data_mobil
-> where warna IN ("hitam","silver")
-> AND harga_rental = 50000;
+-----+-----+-----+-----+-----+
| id_mobil | 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 (0.001 sec)

MariaDB [rental_hayrill]> -
```

ANALISIS:

- **SELECT** adalah sebuah perintah query yang tugas memilih.
- **FROM** adalah sebuah perintah query yang tugas nya untuk pemanggilan dari.
- **WHERE** adalah sebuah perintah query yang tugas untuk menentukan pemanggilan yang mana ditujukannya.
- **DATA\_MOBIL** adalah sebuah nama tabel.
- **WARNA** adalah sebuah nama kolom.
- **HITAM DAN SILVER** adalah nama data pada sebuah kolom.
- **IN** adalah perintah query yang betugas menentukan seluruh data yang sesuai dengan kondisi ditentukan, menghindari penulisan berulang query dengan kondisi yang sama.
- **AND** adalah sebuah perintah query DAN menujukkan bahwa pemilihan data nya lebih dari 1
- **HARGA\_RENTAL** adalah sebuah nama kolom.
- **50000** adalah nama data pada sebuah kolom.

KESIMPULAN:query di atas adalah perintah untuk menampilkan sebuah data yang berkondisikan data warna hitam,silver dan memiliki harga rental 50000.

IN+OR

STRUKTUR:

```
select * from data_mobil
-> where nama_kolom IN ( "nama_data","nama_data")
-> OR nama_kolom = nama_data;
```

CONTOH:

```

    select * from data_mobil
-> where warna IN ( "hitam","silver")
-> OR harga_rental = 50000;

```

HASIL:

```

MariaDB [rental_hayril]> select * from data_mobil
-> where warna in ("hitam","silver")
-> or harga_rental =50000;
+-----+-----+-----+-----+-----+-----+-----+
| id_mobil | 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 |
|      4 | DD 2901 JK | UQL1029 | Hitam | Ibe | NULL | 150000 |
|      5 | DD 2210 LS | CJH1011 | Hitam | Ibe | NULL | 100000 |
|      6 | DD 2812 UY | UYN1312 | HITAM | RAYHAN | NULL | 250000 |
+-----+-----+-----+-----+-----+-----+-----+
5 rows in set (0.001 sec)

```

ANALISIS:

- **SELECT** adalah sebuah perintah query yang tugas memilih.
- **FROM** adalah sebuah perintah query yang tugas nya untuk pemanggilan dari.
- **WHERE** adalah sebuah perintah query yang tugas untuk menentukan pemanggilan yang mana ditujukannya.
- **DATA\_MOBIL** adalah sebuah nama tabel.
- **WARNA** adalah sebuah nama kolom.
- **HITAM DAN SILVER** adalah nama data pada sebuah kolom.
- **IN** adalah perintah query yang betugas menentukan seluruh data yang sesuai dengan kondisi ditentukan, menghindari penulisan berulang query dengan kondisi yang sama.
- **OR** adalah sebuah perintah query ATAU menujukkan bahwa pemilihan dari salah satu data
- **HARGA\_RENTAL** adalah sebuah nama kolom.
- **50000** adalah nama data pada sebuah kolom.

KESIMPULAN:query di atas adalah perintah untuk menampilkan sebuah data yang berkondisikan data warna hitam,silver atau yang memiliki harga rental 50000.

## IN+AND+OPERATOR

STRUKTUR:

```

select * from nama_tabel
-> where nama_kolom in ( "nama_data","nama_data")
-> and nama_kolom > nama_data;
```
```
mysql
select * from nama_tabel
-> where nama_kolom in ( "nama_data","nama_data")
-> and nama_kolom < nama_data;

```

CONTOH:

```
select * from data_mobil
-> where warna in ( "hitam","silver")
-> and harga_rental > 50000;
```

```
select * from data_mobil
-> where warna in ( "hitam","silver")
-> and harga_rental < 100000;
```

HASIL:

```
+-----+
| id_mobil | no_plat   | no_mesin | warna | pemilik | peminjam | harga_rental |
+-----+
|      4  | DD 2901 JK | UQL1029  | Hitam | Ibe     | NULL      | 150000    |
|      5  | DD 2210 LS | CJH1011  | Hitam | Ibe     | NULL      | 100000    |
|      6  | DD 2812 UY | UYN1312  | HITAM | RAYHAN | NULL      | 250000    |
+-----+
3 rows in set (0.001 sec)

MariaDB [rental_hayril]> select * from data_mobil
-> where warna in ("hitam","silver")
-> and harga_rental < 100000;
+-----+
| id_mobil | 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 (0.001 sec)

MariaDB [rental_hayril]>
```

ANALISIS:

- **SELECT** adalah sebuah perintah query yang tugas memilih.
- **FROM** adalah sebuah perintah query yang tugas nya untuk pemanggilan dari.
- **WHERE** adalah sebuah perintah query yang tugas untuk menentukan pemanggilan yang mana ditujukannya.
- **DATA\_MOBIL** adalah sebuah nama tabel.
- **WARNA** adalah sebuah nama kolom.
- **HITAM DAN SILVER** adalah nama data pada sebuah kolom.
- **IN** adalah perintah query yang betugas menentukan seluruh data yang sesuai dengan kondisi ditentukan, menghindari penulisan berulang query dengan kondisi yang sama.
- **DAN** adalah sebuah perintah query DAN menujukkan bahwa kondisi yang ditentukan lebih dari 1
- **HARGA\_RENTAL** adalah sebuah nama kolom.
- **>** adalah perintah lebih dari data yang ditentukan.
- **<** adalah perintah kurang dari data yang ditentukan
- **50000** adalah nama data pada sebuah kolom.
- **100000** adalah nama data pada sebuah kolom.

## KESIMPULAN:

1. query pertama di atas adalah perintah untuk menampilkan kondisi dengan data warna hitam,silver dan yang memiliki harga rental lebih dari 50000.
2. query kedua diatas adalah perintah untuk menampilkan kondisi dengan data warna hitam,silver dan yang memiliki harga rental kurang dari 100000.

## LIKE

### 1. MENCARI AWALAN

STRUKTUR:

```
select * from nama_tabel  
-> where nama_kolom like 'nama_pola';
```

CONTOH:

```
select * from data_mobil  
-> where pemilik like 'ib%';
```

HASIL:

```
XAMPP for Windows - mysql -u root -p  
MariaDB [rental_hayril]> select * from data_mobil  
-> where pemilik like 'ib%';  
+-----+-----+-----+-----+-----+-----+-----+  
| id_mobil | no_plat | no_mesin | warna | pemilik | peminjam | harga_rental |  
+-----+-----+-----+-----+-----+-----+-----+  
| 1 | DD 2650 XY | ACX3568 | Hitam | Ibrahim | Afdal | 50000 |  
| 4 | DD 2901 JK | UQL1029 | Hitam | Ibe | NULL | 150000 |  
| 5 | DD 2210 LS | CJH1011 | Hitam | Ibe | NULL | 100000 |  
+-----+-----+-----+-----+-----+-----+-----+  
3 rows in set (0.128 sec)  
  
MariaDB [rental_hayril]>
```

ANALISIS:

- **SELECT** adalah sebuah perintah query yang tugas memilih.
- **FROM** adalah sebuah perintah query yang tugas nya untuk pemanggilan dari.
- **WHERE** adalah sebuah perintah query yang tugas untuk menentukan pemanggilan yang mana ditujukannya.
- **DATA\_MOBIL** adalah sebuah nama tabel.
- **PEMILIK** adalah sebuah nama kolom.
- **LIKE** adalah perintah query yang betugas menentukan hasil berdasarkan pola teks tertentu.
- **IB%** adalah sebuah pola awalan tertentu yang ingin ditampilkan.

KESIMPULAN: query diatas adalah perintah untuk menampilkan data pada kolom pemilik yang berpolo ib% artinya data dengan awalan ib pada kolom pemilik yang akan ditampilkan.

## 2. MENCARI AKHIRAN

STRUKTUR:

```
select * from nama_tabel  
-> where nama_kolom like 'nama_pola';
```

CONTOH:

```
select * from data_mobil  
-> where pemilik like '%M';
```

HASIL:

```
MariaDB [rental_hayril]> select * from data_mobil where pemilik like '%M';  
+-----+-----+-----+-----+-----+-----+-----+  
| id_mobil | 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 |  
+-----+-----+-----+-----+-----+-----+-----+  
3 rows in set (0.001 sec)  
  
MariaDB [rental_hayril]>
```

ANALISIS:

- **SELECT** adalah sebuah perintah query yang tugas memilih.
- **FROM** adalah sebuah perintah query yang tugas nya untuk pemanggilan dari.
- **WHERE** adalah sebuah perintah query yang tugas untuk menentukan pemanggilan yang mana ditujukannya.
- **DATA\_MOBIL** adalah sebuah nama tabel.
- **PEMILIK** adalah sebuah nama kolom.
- **LIKE** adalah perintah query yang betugas menentukan hasil berdasarkan pola teks tertentu.
- **%M** adalah sebuah pola akhiran tertentu yang ingin ditampilkan.

KESIMPULAN: query diatas adalah perintah untuk menampilkan data pada kolom pemilik yang berpola %m artinya data dengan akhiran M pada kolom pemilik yang akan ditampilkan.

## 3. MENCARI AWALAN DAN AKHIRAN

STRUKTUR:

```
select * from nama_tabel  
-> where nama_kolom like 'nama_pola';
```

CONTOH:

```
select * from data_mobil  
-> where pemilik like 'B%M';
```

HASIL:

```

MariaDB [rental_hayrill]> select * from data_mobil
    -> where pemilik like 'b%m';
+-----+-----+-----+-----+-----+-----+
| id_mobil | no_plat | no_mesin | warna | pemilik | peminjam | harga_rental |
+-----+-----+-----+-----+-----+-----+
|      3 | B 1611 QC | LSQ1112 | Silver | Baim   | Anty     |      50000 |
+-----+-----+-----+-----+-----+-----+
1 row in set (0.001 sec)

MariaDB [rental_hayrill]> 

```

ANALISIS:

- **SELECT** adalah sebuah perintah query yang tugas memilih.
  - **FROM** adalah sebuah perintah query yang tugas nya untuk pemanggilan dari.
  - **WHERE** adalah sebuah perintah query yang tugas untuk menentukan pemanggilan yang mana ditujukannya.
  - **DATA\_MOBIL** adalah sebuah nama tabel.
  - **PEMILIK** adalah sebuah nama kolom.
  - **LIKE** adalah perintah query yang betugas menentukan hasil berdasarkan pola teks tertentu.
  - **B%M** adalah sebuah pola awalan dan akhiran tertentu yang ingin ditampilkan.
- KESIMPULAN: query diatas adalah perintah untuk menampilkan data pada kolom pemilik yang berpola b%m artinya data dengan awalan B dan akhiran M pada kolom pemilik yang akan ditampilkan.

#### 4. MENCARI BERDASARKAN TOTAL KARAKTER

STRUKTUR:

```

select * from nama_tabel
    -> where nama_kolom like 'nama_pola';

```

CONTOH:

```

select * from data_mobil
    -> where pemilik like 'i__';

```

HASIL:

```

XAMPP for Windows - mysql -u root -p
MariaDB [(none)]> use rental_hayrill;
Database changed
MariaDB [rental_hayrill]> select * from data_mobil
    -> where pemilik like 'i__';
+-----+-----+-----+-----+-----+-----+
| id_mobil | no_plat | no_mesin | warna | pemilik | peminjam | harga_rental |
+-----+-----+-----+-----+-----+-----+
|      4 | DD 2901 JK | UQL1029 | Hitam | Ibe    | NULL      | 1500000 |
|      5 | DD 2210 LS | CJH1011 | Hitam | Ibe    | NULL      | 1000000 |
+-----+-----+-----+-----+-----+-----+
2 rows in set (0.001 sec)

```

ANALISIS:

- **SELECT** adalah sebuah perintah query yang tugas memilih.

- **FROM** adalah sebuah perintah query yang tugas nya untuk pemanggilan dari.
- **WHERE** adalah sebuah perintah query yang tugas untuk menentukan pemanggilan yang mana ditujukannya.
- **DATA\_MOBIL** adalah sebuah nama tabel.
- **PEMILIK** adalah sebuah nama kolom.
- **LIKE** adalah perintah query yang betugas menentukan hasil berdasarkan pola teks tertentu.
- '**I\_\_**' adalah pola karakter tertentu yang ingin ditampilkan.

KESIMPULAN:

query adalah perintah untuk menampilkan data pada kolom pemilik yang berawalan i dan memiliki 2 karakter yang lainnya( hanya menampilkan nama pemilik dengan 3 kata dan berawalan i).

## 5. KOMBINASI

STRUKTUR:

```
SELECT * FROM NAMA_TABEL
-> WHERE NAMA_KOLOM LIKE 'NAMA_POLA';
```

CONTOH:

```
SELECT * FROM DATA_MOBIL
-> WHERE PEMILIK LIKE '__R%';
```

HASIL:

```
XAMPP for Windows - mysql -u root -p
MariaDB [rental_hayril]> SELECT * FROM DATA_MOBIL;
+-----+-----+-----+-----+-----+-----+
| id_mobil | 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 |
| 6 | DD 2812 UY | UVN1312 | HITAM | RAYHAN | NULL | 250000 |
+-----+-----+-----+-----+-----+-----+
6 rows in set (0.001 sec)

MariaDB [rental_hayril]> SELECT * FROM DATA_MOBIL
-> WHERE PEMILIK LIKE '__R%';
+-----+-----+-----+-----+-----+-----+
| id_mobil | no_plat | no_mesin | warna | pemilik | peminjam | harga_rental |
+-----+-----+-----+-----+-----+-----+
| 1 | DD 2650 XY | ACX3568 | Hitam | Ibrahim | Afdal | 50000 |
+-----+-----+-----+-----+-----+-----+
1 row in set (0.001 sec)

MariaDB [rental_hayril]>
```

ANALISIS:

- **SELECT** adalah sebuah perintah query yang tugas memilih.
- **FROM** adalah sebuah perintah query yang tugas nya untuk pemanggilan dari.

- **WHERE** adalah sebuah perintah query yang tugas untuk menentukan pemanggilan yang mana ditujukannya.
- **DATA\_MOBIL** adalah sebuah nama tabel.
- **PEMILIK** adalah sebuah nama kolom.
- **LIKE** adalah perintah query yang betugas menentukan hasil berdasarkan pola teks tertentu.
- '**\_R%**' adalah pola karakter tertentu dari salah satu data yang ingin ditampilkan.

KESIMPULAN:

query adalah perintah untuk menampilkan data pada kolom pemilik yang huruf ketiganya adalah R .

## 6. NOT LIKE

STRUKTUR:

```
SELECT * FROM NAMA_TABEL WHERE NAMA_KOLOM NOT LIKE 'NAMA_POLA'
```

CONTOH:

```
SELECT * FROM DATA_MOBIL WHERE PEMINJAM NOT LIKE 'A%';
```

HASIL:

```
1 row in set (0.001 sec)

MariaDB [rental_hayril]> SELECT * FROM DATA_MOBIL WHERE PEMINJAM NOT LIKE 'A%';
+-----+-----+-----+-----+-----+
| id_mobil | no_plat | no_mesin | warna | pemilik | peminjam | harga_rental |
+-----+-----+-----+-----+-----+
|      2 | DD 2440 AX | BCS1120 | Merah | Ibrahim | Elia     |      100000 |
+-----+-----+-----+-----+-----+
1 row in set (0.001 sec)

MariaDB [rental_hayril]>
```

ANALISIS:

- **SELECT** adalah sebuah perintah query yang tugas memilih.
- **FROM** adalah sebuah perintah query yang tugas nya untuk pemanggilan dari.
- **WHERE** adalah sebuah perintah query yang tugas untuk menentukan pemanggilan yang mana ditujukannya.
- **DATA\_MOBIL** adalah sebuah nama tabel.
- **PEMINJAM** adalah sebuah nama kolom.
- **NOT LIKE** adalah perintah query yang betugas menentukan hasil yang tidak sesuai berdasarkan pola teks tertentu.
- '**A%**' adalah pola yang tidak boleh ditampilkan.

KESIMPULAN:

query adalah perintah untuk menampilkan data pada kolom peminjam yang tidak memiliki huruf A pada datanya.

## NULL & NOT NULL

## 1. MENCARI DATA KOSONG

STRUKTUR:

```
SELECT * FROM NAMA_TABEL WHERE NAMA_KOLOM IS NOT NULL;
```

CONTOH:

```
SELECT * FROM DATA_MOBIL WHERE PEMINJAM IS NOT NULL;
```

HASIL:

```
1 row in set (0.001 sec)

MariaDB [rental_hayril]> SELECT * FROM DATA_MOBIL WHERE PEMINJAM IS NULL;
+-----+-----+-----+-----+-----+-----+-----+
| id_mobil | no_plat | no_mesin | warna | pemilik | peminjam | harga_rental |
+-----+-----+-----+-----+-----+-----+-----+
|     4 | DD 2901 JK | UQL1029 | Hitam | Ibe    | NULL      | 150000   |
|     5 | DD 2210 LS | CJH1011 | Hitam | Ibe    | NULL      | 100000   |
|     6 | DD 2812 UY | UYN1312 | HITAM | RAYHAN | NULL      | 250000   |
+-----+-----+-----+-----+-----+-----+-----+
3 rows in set (0.115 sec)

MariaDB [rental_hayril]> -
```

ANALISIS:

- **SELECT** adalah sebuah perintah query yang tugas memilih.
- **FROM** adalah sebuah perintah query yang tugas nya untuk pemanggilan dari.
- **WHERE** adalah sebuah perintah query yang tugas untuk menentukan pemanggilan yang mana ditujukannya.
- **DATA\_MOBIL** adalah sebuah nama tabel.
- **PEMINJAM** adalah sebuah nama kolom.
- **IS NULL** adalah perintah query yang betugas menampilkan suatu kolom yang memiliki data.

KESIMPULAN:

query adalah perintah untuk menampilkan tabel yang pada kolom peminjamnya tidak memiliki data atau NULL.

## 2. NOT NULL

STRUKTUR:

```
SELECT * FROM NAMA_TABEL WHERE NAMA_KOLOM IS NOT NULL;
```

CONTOH:

```
SELECT * FROM DATA_MOBIL WHERE PEMINJAM IS NOT NULL;
```

HASIL:

```

3 rows in set (0.115 sec)

MariaDB [rental_hayril]> SELECT * FROM DATA_MOBIL WHERE PEMINJAM IS NOT NULL;
+-----+-----+-----+-----+-----+-----+
| id_mobil | 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      | 1000000  |
|      3  | B 1611 QC   | LSQ1112  | Silver   | Baim      | Anty      |      50000  |
+-----+-----+-----+-----+-----+-----+
3 rows in set (0.001 sec)

MariaDB [rental_hayril]> .

```

ANALISIS:

- **SELECT** adalah sebuah perintah query yang tugas memilih.
- **FROM** adalah sebuah perintah query yang tugas nya untuk pemanggilan dari.
- **WHERE** adalah sebuah perintah query yang tugas untuk menentukan pemanggilan yang mana ditujukannya.
- **DATA\_MOBIL** adalah sebuah nama tabel.
- **PEMINJAM** adalah sebuah nama kolom.
- **IS NOT NULL** adalah perintah query yang betugas menampilkan suatu kolom yang memiliki nilai .

KESIMPULAN:

query adalah perintah untuk menampilkan tabel yang pada kolom peminjamnya terdapat sebuah data atau NOT NULL.

## ORDER BY LIMIT

### 1. ASC

STRUKTUR:

```
SELECT * FROM NAMA_TABEL ORDER BY NAMA_KOLOM ASC;
```

CONTOH:

```
SELECT * FROM DATA_MOBIL ORDER BY PEMILIK ASC;
```

HASIL:

```

MariaDB [rental_hayril]> select * from data_mobil order by pemilik asc;
+-----+-----+-----+-----+-----+-----+
| id_mobil | no_plat      | no_mesin | warna   | pemilik    | peminjam  | harga_rental |
+-----+-----+-----+-----+-----+-----+
|      2  | DD 2440 AX  | BCS1120  | Merah   | Ibrahim    | Elia      | 1000000  |
|      3  | B 1611 QC   | LSQ1112  | Silver   | Baim      | Anty      |      50000  |
|      4  | DD 2901 JK   | UQL1029  | Hitam   | Ibe       | NULL      | 1500000  |
|      5  | DD 2210 LS   | CJH1011  | Hitam   | Ibe       | NULL      | 1000000  |
|      1  | DD 2650 XY  | ACX3568  | Hitam   | Ibrahim    | Afdal     |      50000  |
|      6  | DD 2812 UY   | UYN1312  | HITAM   | RAYHAN    | NULL      | 2500000  |
+-----+-----+-----+-----+-----+-----+
6 rows in set (0.003 sec)

```

ANALISIS:

- **SELECT** adalah sebuah perintah query yang tugas memilih.
- **FROM** adalah sebuah perintah query yang tugas nya untuk pemanggilan dari.
- **ORDER BY** adalah sebuah perintah query yang untuk mengurutkan pemanggilan berdasarkan nilai.
- **DATA\_MOBIL** adalah sebuah nama tabel.
- **PEMINJAM** adalah sebuah nama kolom.
- **ASC** adalah perintah query yang betugas menampilkan suatu data secara berurutan ke atas.

KESIMPULAN:

query adalah perintah untuk menampilkan tabel dengan data pada kolom peminjam sesuai dengan alfabet dari A-Z.

## 2. DESC

STRUKTUR:

```
SELECT * FROM NAMA_TABEL ORDER BY NAMA_KOLOM DESC;
```

CONTOH:

```
SELECT * FROM DATA_MOBIL ORDER BY PEMILIK DESC;
```

HASIL:

```
MariaDB [rental_hayril]> select * from data_mobil order by pemilik desc;
+-----+-----+-----+-----+-----+-----+-----+
| id_mobil | no_plat | no_mesin | warna | pemilik | peminjam | harga_rental |
+-----+-----+-----+-----+-----+-----+-----+
| 6 | DD 2812 UY | UYN1312 | HITAM | RAYHAN | NULL | 250000 |
| 1 | DD 2650 XY | ACX3568 | Hitam | Ibrahim | Afdal | 50000 |
| 4 | DD 2901 JK | UQL1029 | Hitam | Ibe | NULL | 150000 |
| 5 | DD 2210 LS | CJH1011 | Hitam | Ibe | NULL | 100000 |
| 3 | B 1611 QC | LSQ1112 | Silver | Baim | Anty | 50000 |
| 2 | DD 2440 AX | BCS1120 | Merah | Ibrahim | Elia | 100000 |
+-----+-----+-----+-----+-----+-----+-----+
6 rows in set (0.001 sec)

MariaDB [rental_hayril]> -
```

ANALISIS:

- **SELECT** adalah sebuah perintah query yang tugas memilih.
- **FROM** adalah sebuah perintah query yang tugas nya untuk pemanggilan dari.
- **ORDER BY** adalah sebuah perintah query yang untuk mengurutkan pemanggilan berdasarkan nilai.
- **DATA\_MOBIL** adalah sebuah nama tabel.
- **PEMINJAM** adalah sebuah nama kolom.
- **DESC** adalah perintah query yang betugas menampilkan suatu data secara berurutan ke bawah .

KESIMPULAN:

query di atas adalah perintah untuk menampilkan tabel dengan data pada kolom peminjam sesuai dengan alfabet dari Z-A.

## 3. LIMIT

STRUKTUR:

```
SELECT * FROM NAMA_TABEL ORDER BY NAMA_KOLOM ASC;
```

CONTOH:

```
SELECT * FROM DATA_MOBIL ORDER BY PEMILIK ASC;
```

HASIL:

```
MariaDB [rental_hayril]> select * from data_mobil order by pemilik asc;
+-----+-----+-----+-----+-----+-----+
| id_mobil | no_plat | no_mesin | warna | pemilik | peminjam | harga_rental |
+-----+-----+-----+-----+-----+-----+
|      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 |
|      1 | DD 2650 XY | ACX3568 | Hitam | Ibrahim | Afdal | 50000 |
|      6 | DD 2812 UY | UYN1312 | HITAM | RAYHAN | NULL | 250000 |
+-----+-----+-----+-----+-----+-----+
6 rows in set (0.003 sec)
```

ANALISIS:

- **SELECT** adalah sebuah perintah query yang tugas memilih.
- **FROM** adalah sebuah perintah query yang tugas nya untuk pemanggilan dari.
- **ORDER BY** adalah sebuah perintah query yang untuk mengurutkan pemanggilan berdasarkan nilai.
- **DATA\_MOBIL** adalah sebuah nama tabel.
- **PEMINJAM** adalah sebuah nama kolom.
- **ASC** adalah perintah query yang betugas menampilkan suatu data secara berurutan ke atas.

KESIMPULAN:

query adalah perintah untuk menampilkan tabel dengan data pada kolom peminjam sesuai dengan alfabet dari A-Z.

### 3. LIMIT

STRUKTUR:

```
SELECT * FROM NAMA_TABEL WHERE NAMA_KOLOM="NAMA_DATA" ORDER BY NAMA_KOLOM ASC LIMIT 2;
```

CONTOH:

```
SELECT * FROM DATA_MOBIL WHERE WARNA="HITAM" ORDER BY HARGA_RENTAL ASC LIMIT 2;
```

HASIL:

```
MariaDB [rental_hayril]> SELECT * FROM DATA_MOBIL WHERE WARNA="HITAM" ORDER BY HARGA_RENTAL ASC LIMIT 2;
+-----+-----+-----+-----+-----+-----+
| id_mobil | no_plat | no_mesin | warna | pemilik | peminjam | harga_rental |
+-----+-----+-----+-----+-----+-----+
|      1 | DD 2650 XY | ACX3568 | Hitam | Ibrahim | Afdal | 50000 |
|      5 | DD 2210 LS | CJH1011 | Hitam | Ibe | NULL | 100000 |
+-----+-----+-----+-----+-----+-----+
2 rows in set (0.001 sec)

MariaDB [rental_hayril]> -
```

#### ANALISIS:

- **SELECT** adalah sebuah perintah query yang tugas memilih.
- **FROM** adalah sebuah perintah query yang tugas nya untuk pemanggilan dari.
- **ORDER BY** adalah sebuah perintah query yang untuk mengurutkan pemanggilan berdasarkan nilai.
- **DATA\_MOBIL** adalah sebuah nama tabel.
- **WHERE** adalah sebuah perintah query yang tugas untuk menentukan pemanggilan yang mana ditujukannya.
- **PEMINJAM & WARNA** adalah sebuah nama kolom.
- **ASC** adalah perintah query yang betugas menampilkan suatu data secara berurutan ke atas.
- **HITAM** adalah nama data pada sebuah kolom.
- **LIMIT 2** adalah perintah untuk membatasi jumlah baris yang ingin ditampilkan.

#### KESIMPULAN:

Query di atas adalah perintah untuk menampilkan tabel dengan data hitam pada kolom warna yang sesuai dengan alfabet A-Z tapi ditampilkan hanya 2 baris .

## DISTINCT

#### STRUKTUR:

```
SELECT DISTINCT( NAMA_KOLOM) FROM NAMA_TABEL;
```

#### CONTOH:

```
SELECT DISTINCT( PEMILIK) FROM DATA_MOBIL;
```

#### HASIL:

```
MariaDB [rental_hayril]> select distinct(pemilik) from data_mobil;
+-----+
| pemilik |
+-----+
| Ibrahim |
| Ibrahim |
| Baim    |
| Ibe     |
| RAYHAN  |
+-----+
```

#### ANALISIS:

- **SELECT** adalah perintah query yang tugas nya memilih.
- **DISTINCT** adalah perintah untuk mengembalikan nilai unik dari hasil query.
- **PEMILIK** adalah sebuah nama kolom.
- **FROM** adalah sebuah pemanggilan dari data yang ditujukan.
- **DATA\_MOBIL** adalah sebuah nama tabel.

#### KESIMPULAN:

query di atas perintah ini akan mengembalikan setiap nilai yang berbeda dari kolom "PEMILIK", tanpa adanya duplikasi. Ini berguna untuk melihat daftar pemilik mobil yang berbeda yang terdapat dalam tabel tersebut.

## CONCAT, CONCAT\_WS , AS

### 1. CONCAT

STRUKTUR:

```
SELECT CONCAT( NAMA_TABEL,NAMA_KOLOM) FROM NAMA_TABEL;
```

CONTOH:

```
SELECT CONCAT( PEMILIK,WARNA) FROM DATA_MOBIL;
```

HASIL:

```
MariaDB [rental_hayril]> SELECT CONCAT(PEMILIK,WARNA) FROM data_mobil;
+-----+
| CONCAT(PEMILIK,WARNA) |
+-----+
| IbrahimHitam
| IbrahimMerah
| BaimSilver
| IbeHitam
| IbeHitam
| RAYHANHITAM
+-----+
6 rows in set (0.109 sec)
```

ANALISIS:

- **SELECT** adalah perintah query yang tugas nya memilih.
- **CONCAT** adalah perintah untuk menggabungkan sebuah kolom untuk ditampilkan.
- **PEMILIK & WARNA** adalah sebuah nama kolom.
- **FROM** adalah sebuah pemanggilan dari data yang ditujukan.
- **DATA\_MOBIL** adalah sebuah nama tabel.

KESIMPULAN:

query di atas adalah perintah untuk menggabungkan kolom pemilik dan warna untuk ditampilkan

### 2. CONCAT\_WS

STRUKTUR:

```
SELECT CONCAT_WS ( "PEMISAH",NAMA_KOLOM,NAMA_KOLOM,NAMA_KOLOM) FROM NAMA_TABEL;
```

CONTOH:

```
SELECT CONCAT_WS ( "-",NO_PLAT,NO_MESIN,ID_MOBIL) FROM DATA_MOBIL;
```

HASIL:

```
MariaDB [rental_hayril]> SELECT CONCAT_WS ("-",NO_PLAT,NO_MESIN,ID_MOBIL) FROM DATA_MOBIL;
+-----+
| CONCAT_WS ("-",NO_PLAT,NO_MESIN,ID_MOBIL) |
+-----+
| DD 2650 XY-ACX3568-1 |
| DD 2440 AX-BCS1120-2 |
| B 1611 QC- LSQ1112-3 |
| DD 2901 JK-UQL1029-4 |
| DD 2210 LS-CJH1011-5 |
| DD 2812 UY-UYN1312-6 |
+-----+
6 rows in set (0.109 sec)
```

ANALISIS:

- **SELECT** adalah perintah query yang tugas nya memilih.
- **CONCAT\_WS** adalah perintah untuk menggabungkan nilai nilai sebuah kolom dan memiliki pemisah tertentu untuk ditampilkan.
- **NO\_PLAT,NO\_MESIN,ID\_MOBIL** adalah sebuah nama kolom.
- **"-"** adalah pemisah dari kolom yang di satukan.
- **FROM** adalah sebuah pemanggilan dari data yang ditujukan.
- **DATA\_MOBIL** adalah sebuah nama tabel.

KESIMPULAN:

query di atas adalah perintah untuk menggabungkan kolom no\_plat,no\_mesin,id\_mobil menggunakan **"-** sebagai pemisah untuk ditampilkan.

### 3. AS

STRUKTUR:

```
SELECT CONCAT_WS ( "PENGHUBUNG",NAMA_KOLOM,NAMA_KOLOM) FROM NAMA_TABEL;
```

CONTOH:

```
SELECT CONCAT_WS ( "+",PEMILIK,PEMINJAM) FROM DATA_MOBIL;
```

HASIL:

```
MariaDB [rental_hayril]> SELECT CONCAT_WS ("+",PEMILIK,PEMINJAM) AS COLLAB FROM DATA_MOBIL;
+-----+
| COLLAB |
+-----+
| Ibrahim+Afdal |
| Ibrahim+Elia |
| Baim+Anty |
| Ibe |
| Ibe |
| RAYHAN |
+-----+
6 rows in set (0.134 sec)

MariaDB [rental_hayril]>
```

ANALISIS:

- **SELECT** adalah perintah query yang tugas nya memilih.

- **CONCAT\_WS** adalah perintah untuk menggabungkan nilai nilai sebuah kolom dan memiliki pemisah tertentu untuk ditampilkan.
- **PEMILIK,PEMINJAM** adalah sebuah nama kolom.
- "+" adalah penghubung dari kolom yang di satukan.
- **FROM** adalah sebuah pemanggilan dari data yang ditujukan.
- **DATA\_MOBIL** adalah sebuah nama tabel.

KESIMPULAN:

query di atas adalah perintah untuk menggabungkan kolom pemilik dan peminjam menggunakan "+" sebagai penghubung untuk ditampilkan.

## VIEW

### 1. MEMBUAT TABEL VIRTUAL

STRUKTUR:

```
CREATE VIEW NAMA_TABEL_VIRTUAL AS
SELECT NAMA_KOLOM,NAMA_KOLOM,NAMA_KOLOM,NAMA_KOLOM
FROM NAMA_TABEL
WHERE NAMA_KOLOM="NAMA_DATA";
```

CONTOH:

```
CREATE VIEW INFO_NO_PLAT AS
SELECT ID_MOBIL,NO_PLAT,PEMILIK,PEMINJAM
FROM DATA_MOBIL
WHERE PEMILIK="IBRAHIM";
```

HASIL:

```
MariaDB [rental_hayril]> create view info_no_plat as
    -> select id_mobil,no_plat,pemilik,peminjam
    -> from data_mobil
    -> where pemilik="ibrahim";
Query OK, 0 rows affected (0.229 sec)

MariaDB [rental_hayril]> show table;
ERROR 1064 (42000): You have an error in your SQL syntax
MariaDB [rental_hayril]> show tables;
+-----+
| Tables_in_rental_hayril |
+-----+
| data_mobil               |
| info_no_plat             |
+-----+
2 rows in set (0.001 sec)

MariaDB [rental_hayril]>
```

ANALISIS:

- **SELECT** adalah perintah query yang tugas nya memilih.
- **CREATE VIEW** adalah perintah untuk membuat sebuah tabel virtual

- **ID\_MOBIL,NO\_PLAT,PEMILIK,PEMINJAM** adalah sebuah nama kolom.
  - **FROM** adalah sebuah pemanggilan dari data yang ditujukan.
  - **DATA\_MOBIL** adalah sebuah nama tabel.
  - **WHERE** adalah sebuah perintah query yang tugas untuk menentukan pemanggilan yang mana ditujukannya.
  - **IBRAHIM** adalah nama data pada suatu kolom
- KESIMPULAN:  
query diatas adalah perintah untuk membuat sebuah tabel virtual yang ditampilkan adalah kolom id\_mobil,no\_plat,pemilik,peminjam yang sesuai dengan data ibrahim pada kolom pemilik.

## 2. MENAMPILKAN TABEL VIRTUAL

STRUKTUR:

```
SELECT * FROM NAMA_TABEL_VITUAL;
```

CONTOH:

```
SELECT * FROM INFO_NO_PLAT;
```

HASIL:

```
MariaDB [rental_hayril]> select * from info_no_plat;
+-----+-----+-----+-----+
| id_mobil | no_plat | pemilik | peminjam |
+-----+-----+-----+-----+
| 1 | DD 2650 XY | Ibrahim | Afdal |
+-----+-----+-----+
1 row in set (0.143 sec)

MariaDB [rental_hayril]> -
```

ANALISIS:

- **SELECT** adalah perintah query yang tugas nya memilih.
  - **FROM** adalah perintah query dari.
  - **INFO\_NO\_PLAT** adalah sebuah nama tabel virtual yang telah dibuat
- KESIMPULAN:  
query di atas adalah perintah untuk menampilkan tabel virtual yang dibuat tadi.

## 3. MENGHAPUS TABEL VIRTUAL

STRUKTUR:

```
DROP VIEW NAMA_TABEL_VITUAL;
```

CONTOH:

```
DROP VIEW INFO_NO_PLAT;
```

HASIL:

```

MariaDB [rental_hayril]> show tables;DROP VIEW INFO_NO_PLAT;
+-----+
| Tables_in_rental_hayril |
+-----+
| data_mobil              |
| info_no_plat             |
+-----+
2 rows in set (0.001 sec)

Query OK, 0 rows affected (0.133 sec)

MariaDB [rental_hayril]> DROP VIEW INFO_NO_PLAT;
ERROR 4092 (42S02): Unknown VIEW: 'rental_hayril.info_no_plat'
MariaDB [rental_hayril]> show tables;
+-----+
| Tables_in_rental_hayril |
+-----+
| data_mobil              |
+-----+
1 row in set (0.001 sec)

MariaDB [rental_hayril]>

```

ANALISIS:

query di atas adalah perintah untuk menghapus tabel yang telah dibuat tadi.

## Tantangan

### Pertama

Contoh Query:

```

create view
-> mobil_tanpa_peminjam as
-> select no_plat,peminjam
-> from mobil
-> where peminjam is null;

```

Hasil

```

ERROR 1118 (42S02): Table 'rental_raihan.mobil_tanpa_peminjam' doesn't exist
MariaDB [rental_raihan]> select * from mobil_tanpa_peminjam;
+-----+-----+
| no_plat | peminjam |
+-----+-----+
| DD 2901 JK | NULL    |
| DD 2210 LS | NULL    |
+-----+-----+
2 rows in set (0.005 sec)

MariaDB [rental_raihan]>

```

## Analisis:

Perintah query MySQL di atas membuat sebuah view dengan nama mobil\_tanpa\_peminjam. Berikut adalah analisisnya:

- *Pembuatan View:*

Perintah CREATE VIEW digunakan untuk membuat sebuah view baru dalam basis data.

- *Data yang Dimuat:*

View ini akan memuat data dari tabel mobil dengan hanya memilih kolom no\_plat dan peminjam.

- *Kondisi Pemilihan Data:*

Data yang dimuat dalam view ini akan dipilih berdasarkan kondisi WHERE peminjam IS NULL.

Artinya, hanya data mobil yang tidak memiliki peminjam ( peminjamnya NULL) yang akan dimuat dalam view i

## Kesimpulan:

Detik CREATE VIEW mobil\_tanpa\_peminjam AS SELECT no\_plat, peminjam FROM mobil WHERE peminjam IS NULL; membuat sebuah view baru yang menampilkan informasi tentang mobil-mobil yang saat ini tidak memiliki peminjam atau ( null). Ini memungkinkan akses mudah dan terstruktur ke data tentang mobil-mobil yang tersedia untuk dipinjamkan, serta mendukung pemantauan dan analisis terkait status peminjaman mobil.

## Kedua

### Penjelasan:

UPDATE mobil: adalah perintah untuk memperbarui data dalam tabel yang disebut mobil.

- SET peminjaman = NULL : menetapkan nilai kolom peminjam menjadi NULL.
- WHERE peminjam= 'Afdal' : adalah klausa WHERE yang membatasi update hanya pada baris-baris dimana nilai kolom peminjam adalah 'Afdal'. Maksudnya perubahan hanya akan berlaku untuk baris-baris yang memiliki peminjam dengan nama 'Afdal'.
- SELECT \*: adalah perintah untuk memilih semua kolom dari view atau tabel.
- FROM mobil\_tanpa\_peminjam: Menunjukkan bahwa data diambil dari view yang disebut "mobil\_tanpa\_peminjam", yang telah dibuat sebelumnya.

### Query

```
UPDATE mobil
-> SET peminjaman = NULL
-> WHERE peminjaman = 'Afdal';
```

### Hasil:

| id_pelanggan | no_plat    | no_mesin | warna  | pemilik | peminjam | harga_rental |
|--------------|------------|----------|--------|---------|----------|--------------|
| 1            | DD 2650 XY | ACX3568  | Hitam  | Ibrahim | NULL     | 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     | Rahmat   | 150000       |
| 5            | DD 2210 LS | CJH1011  | Hitam  | Ibe     | NULL     | 100000       |
| 6            | DD 4539 QW | SLA 1121 | PUTIH  | Ibe     | NULL     | 200000       |
| 7            | B 4532 H   | RPL2122  | Hitam  | Valen   | NULL     | 100000       |

7 rows in set (0.003 sec)

Kesimpulan:

Pernyataan `UPDATE` digunakan untuk memperbarui nilai kolom dalam tabel. Dalam pernyataan SQL ini, tabel yang diubah adalah "mobil".

Pernyataan `SET peminjaman = NULL` berarti kita ingin mengubah nilai kolom "peminjaman" menjadi NULL untuk baris-baris yang memenuhi kondisi tertentu.

Kondisi tersebut dinyatakan dengan `WHERE peminjaman = 'Afdal'`, yang berarti kita hanya ingin mengupdate baris-baris di mana kolom "peminjaman" memiliki nilai 'Afdal'. Hal ini bertujuan untuk membatasi pembaruan hanya pada data yang memiliki nilai peminjaman yang sama dengan 'Afdal'.

Dengan demikian, kesimpulan dari pernyataan SQL ini adalah bahwa kita ingin mengubah nilai kolom "peminjaman" menjadi NULL hanya untuk baris-baris di mana kolom "peminjaman" memiliki nilai 'Afdal'.

## Ketiga

Berikan Kesimpulan mengapa tabel Virtual 1,2 ini dibuat?

View dapat digunakan untuk menyaring data yang sesuai dengan kriteria tertentu, seperti menampilkan entri yang memiliki nilai NULL pada kolom tertentu atau mengubah salah satu data peminjam menjadi NULL. memberikan pandangan yang jelas tentang mobil yang tersedia untuk disewakan atau yang belum dipinjam.

Dengan membuat view dapat membatasi akses ke data sensitif atau kolom tertentu dari tabel yang mungkin tidak perlu diakses oleh semua pengguna.

Dengan membuat view untuk kueri yang sering digunakan, Anda dapat menghindari pengulangan kode SQL yang sama di beberapa tempat dalam aplikasi atau prosedur penyimpanan.

## AGREGASI

### 1. MENGHITUNG TOTAL

STRUKTUR:

```
SELECT SUM( NAMA_KOLOM) AS NAMA_KOLOM FROM NAMA_TABEL
```

CONTOH:

```
SELECT SUM( HARGA_RENTAL) AS TOTAL_HARGA FROM DATA_MOBIL;
```

HASIL:

```
MariaDB [rental_hayril]> SELECT SUM(HARGA_RENTAL) AS TOTAL_HARGA FROM DATA_MOBIL;
+-----+
| TOTAL_HARGA |
+-----+
|    700000 |
+-----+
1 row in set (0.117 sec)
```

ANALISIS:

- **SELECT** adalah perintah untuk memilih suatu kolom
- **SUM** adalah perintah untuk menghitung semua total jumlah
- **HARGA\_RENTAL** adalah sebuah nama kolom
- **AS** adalah perintah untuk memberikan nama lain dari perintah tersebut.
- **TOTAL\_HARGA** nama lain dari kolom pada perintah tersebut
- **DATA\_MOBIL** adalah sebuah nama tabel.

KESIMPULAN:

query di atas adalah perintah untuk menghitung total data dari kolom harga rental dan diubah nama nya menjadi total harga

## 2. MENGHITUNG JUMLAH BARIS

STRUKTUR:

```
SELECT COUNT( NAMA_KOLOM) AS NAMA_KOLOM FROM NAMA_TABEL
```

CONTOH:

```
SELECT COUNT( PEMILIK) AS JUMLAH PEMILIK FROM DATA_MOBIL;
```

HASIL:

```
MariaDB [rental_hayril]> SELECT COUNT(PEMILIK) AS TOTAL PEMILIK FROM DATA_MOBIL;
+-----+
| TOTAL PEMILIK |
+-----+
|      6 |
+-----+
1 row in set (0.001 sec)
```

ANALISIS:

- **SELECT** adalah perintah untuk memilih suatu kolom
- **COUNT** adalah perintah untuk menghitung semua jumlah baris.
- **JUMLAH PEMILIK** adalah sebuah nama kolom
- **AS** adalah perintah untuk memberikan nama lain dari perintah tersebut.
- **PEMILIK** nama lain dari kolom pada perintah tersebut
- **DATA\_MOBIL** adalah sebuah nama tabel.

KESIMPULAN:

query di atas adalah perintah untuk menghitung jumlah data pada kolom pemilik.

### 3. MENGHITUNG NILAI TERENDAH

STRUKTUR:

```
SELECT MIN( NAMA_KOLOM) AS NAMA_KOLOM FROM NAMA_TABEL
```

CONTOH:

```
SELECT MIN( HARGA_RENTAL) AS MINIMUM FROM DATA_MOBIL;
```

HASIL:

```
MariaDB [rental_hayril]> SELECT MIN(HARGA_RENTAL) AS MINIMUM FROM DATA_MOBIL;
+-----+
| MINIMUM |
+-----+
| 50000 |
+-----+
1 row in set (0.001 sec)
```

ANALISIS:

- **SELECT** adalah perintah untuk memilih suatu kolom
- **MIN** adalah perintah untuk menghitung jumlah data yang paling terkecil.
- **HARGA\_RENTAL** adalah sebuah nama kolom
- **AS** adalah perintah untuk memberikan nama lain dari perintah tersebut.
- **MINIMUM** nama lain dari kolom pada perintah tersebut
- **DATA\_MOBIL** adalah sebuah nama tabel.

KESIMPULAN:

query di atas adalah perintah untuk menghitung jumlah data terkecil dari kolom harga rental dan ditampilkan dengan kolom baru yaitu MINIMUM.

### 4. MENGHITUNG NILAI TERTINGGI

STRUKTUR:

```
SELECT MAX( NAMA_KOLOM) AS NAMA_KOLOM FROM NAMA_TABEL
```

CONTOH:

```
SELECT MAX( HARGA_RENTAL) AS MAXIMUM FROM DATA_MOBIL;
```

HASIL:

```
MariaDB [rental_hayril]> SELECT MAX(HARGA_RENTAL) AS MAXIMUM FROM DATA_MOBIL;
+-----+
| MAXIMUM |
+-----+
| 250000 |
+-----+
1 row in set (0.001 sec)
```

ANALISIS:

- **SELECT** adalah perintah untuk memilih suatu kolom
- **MAX** adalah perintah untuk menghitung data yang paling tinggi.
- **HARGA\_RENTAL** adalah sebuah nama kolom
- **AS** adalah perintah untuk memberikan nama lain dari perintah tersebut.
- **MAXIMUM** nama lain dari kolom pada perintah tersebut
- **DATA\_MOBIL** adalah sebuah nama tabel.

KESIMPULAN:

query di atas adalah perintah untuk menghitung jumlah data tertinggi dari kolom harga rental dan ditampilkan dengan kolom baru yaitu MAXIMUM.

## 5. MENAMPILKAN NILAI RATA RATA

STRUKTUR:

```
SELECT AVG( NAMA_KOLOM) AS NAMA_KOLOM FROM NAMA_TABEL
```

CONTOH:

```
SELECT AVG( HARGA_RENTAL) AS RERATA FROM DATA_MOBIL;
```

HASIL:

```
MariaDB [rental_hayril]> SELECT SUM(HARGA_RENTAL) AS TOTAL_HARGA FROM DATA_MOBIL;
+-----+
| TOTAL_HARGA |
+-----+
|    700000 |
+-----+
1 row in set (0.117 sec)
```

ANALISIS:

- **SELECT** adalah perintah untuk memilih suatu kolom
- **AVG** adalah perintah untuk menghitung jumlah rata rata
- **HARGA\_RENTAL** adalah sebuah nama kolom
- **AS** adalah perintah untuk memberikan nama lain dari perintah tersebut.
- **RERATA** nama lain dari kolom pada perintah tersebut
- **DATA\_MOBIL** adalah sebuah nama tabel.

KESIMPULAN:

query di atas adalah perintah untuk menghitung rata rata jumlah data dari kolom harga rental dan ditampilkan dengan kolom baru yaitu RERATA.

## Relasi

### Data Tabel Pegawai

Showing rows 0 - 8 (9 total, Query took 0.0004 seconds.)

SELECT \* FROM `pegawai`

Profiling [ Edit inline ] [ Edit ] [ Explain SQL ] [ Create PHP code ] [ Refresh ]

Show all | Number of rows: 25 ▾ Filter rows: Search this table Sort by key: None ▾

Extra options

|                          | ← T → |      | IdPeg | Nama | NamaKel | JK     | Alamat    | Kota | Telp           | NoCab                    |
|--------------------------|-------|------|-------|------|---------|--------|-----------|------|----------------|--------------------------|
| <input type="checkbox"/> |       | Edit |       |      | 10107   | Emya   | Salsalina | P    | Jl.Suci 78     | Bandung 022-555768 C101  |
| <input type="checkbox"/> |       | Edit |       |      | 10176   | Diah   | Wahyuni   | P    | Jl.Maluku 56   | Bandung 022-555934 C101  |
| <input type="checkbox"/> |       | Edit |       |      | 10246   | Dian   | Anggraini | P    | Jl.Mawar 5     | Semarang 024-555102 C103 |
| <input type="checkbox"/> |       | Edit |       |      | 10252   | Antoni | Irawan    | L    | Jl.A.Yani 15   | Jakarta 021-555888 C102  |
| <input type="checkbox"/> |       | Edit |       |      | 10307   | Erik   | Andrian   | L    | Jl.Manggis 155 | Semarang 024-555236 C103 |
| <input type="checkbox"/> |       | Edit |       |      | 10314   | Ayu    | Rahmadani | P    | Jl.Malaka 342  | Jakarta 021-555098 C102  |
| <input type="checkbox"/> |       | Edit |       |      | 10324   | Martin | Susanto   | L    | Jl.Bima 51     | Jakarta 021-555785 C102  |
| <input type="checkbox"/> |       | Edit |       |      | 10407   | Rio    | Gunawan   | L    | Jl.Melati 356  | Surabaya 031-555231 C104 |
| <input type="checkbox"/> |       | Edit |       |      | 10415   | Susan  | Sumantri  | P    | Jl.Pahlawan 24 | Surabaya 031-555120 C104 |

↑  Check all With selected: Edit Copy Delete Export

## Perubahan Struktur Table

Before

| #                        | Name             | Type           | Collation          | Attributes | Null | Default | Comments | Extra | Action |  |
|--------------------------|------------------|----------------|--------------------|------------|------|---------|----------|-------|--------|--|
| <input type="checkbox"/> | 1 <b>IdPeg</b>   | int(10)        |                    |            | No   | None    |          |       |        |  |
| <input type="checkbox"/> | 2 <b>Nama</b>    | varchar(25)    | utf8mb4_general_ci |            | No   | None    |          |       |        |  |
| <input type="checkbox"/> | 3 <b>NamaKel</b> | varchar(25)    | utf8mb4_general_ci |            | No   | None    |          |       |        |  |
| <input type="checkbox"/> | 4 <b>JK</b>      | enum('L', 'P') | utf8mb4_general_ci |            | No   | None    |          |       |        |  |
| <input type="checkbox"/> | 5 <b>Alamat</b>  | varchar(25)    | utf8mb4_general_ci |            | No   | None    |          |       |        |  |
| <input type="checkbox"/> | 6 <b>Kota</b>    | varchar(25)    | utf8mb4_general_ci |            | No   | None    |          |       |        |  |
| <input type="checkbox"/> | 7 <b>Telp</b>    | varchar(15)    | utf8mb4_general_ci |            | No   | None    |          |       |        |  |
| <input type="checkbox"/> | 8 <b>NoCab</b>   | varchar(10)    | utf8mb4_general_ci |            | No   | None    |          |       |        |  |

After

✓ Table pegawai has been altered successfully.

```
ALTER TABLE `pegawai` CHANGE `Nama` `Nama` TEXT CHARACTER SET utf8mb4 COLLATE utf8mb4_general_ci NOT NULL;
```

[ Edit inline ] [ Edit ] [ Create PHP code ]

Table structure Relation view

| # | Name    | Type           | Collation          | Attributes | Null | Default | Comments | Extra | Action            |
|---|---------|----------------|--------------------|------------|------|---------|----------|-------|-------------------|
| 1 | IdPeg   | int(10)        |                    |            | No   | None    |          |       | Change  Drop More |
| 2 | Nama    | text           | utf8mb4_general_ci |            | No   | None    |          |       | Change  Drop More |
| 3 | NamaKel | varchar(25)    | utf8mb4_general_ci |            | No   | None    |          |       | Change  Drop More |
| 4 | JK      | enum('L', 'P') | utf8mb4_general_ci |            | No   | None    |          |       | Change  Drop More |
| 5 | Alamat  | varchar(25)    | utf8mb4_general_ci |            | No   | None    |          |       | Change  Drop More |
| 6 | Kota    | varchar(25)    | utf8mb4_general_ci |            | No   | None    |          |       | Change  Drop More |
| 7 | Telp    | varchar(15)    | utf8mb4_general_ci |            | No   | None    |          |       | Change  Drop More |
| 8 | NoCab   | varchar(10)    | utf8mb4_general_ci |            | No   | None    |          |       | Change  Drop More |

## Perubahan Data Tabel Pegawai & Cabang

### Pegawai

Before

Showing rows 0 - 8 (9 total, Query took 0.0004 seconds.)

```
SELECT * FROM `pegawai`
```

Profiling [ Edit inline ] [ Edit ] [ Explain SQL ] [ Create PHP code ] [ Refresh ]

Show all | Number of rows: 25 Filter rows: Search this table Sort by key: None

Extra options

|                          |  |  |  | IdPeg | Nama   | NamaKel   | JK | Alamat         | Kota     | Telp       | NoCab |
|--------------------------|--|--|--|-------|--------|-----------|----|----------------|----------|------------|-------|
| <input type="checkbox"/> |  |  |  | 10107 | Emya   | Salsalina | P  | Jl.Suci 78     | Bandung  | 022-555768 | C101  |
| <input type="checkbox"/> |  |  |  | 10176 | Diah   | Wahyuni   | P  | Jl.Maluku 56   | Bandung  | 022-555934 | C101  |
| <input type="checkbox"/> |  |  |  | 10246 | Dian   | Anggraini | P  | Jl.Mawar 5     | Semarang | 024-555102 | C103  |
| <input type="checkbox"/> |  |  |  | 10252 | Antoni | Irawan    | L  | Jl.A.Yani 15   | Jakarta  | 021-555888 | C102  |
| <input type="checkbox"/> |  |  |  | 10307 | Erik   | Andrian   | L  | Jl.Manggis 155 | Semarang | 024-555236 | C103  |
| <input type="checkbox"/> |  |  |  | 10314 | Ayu    | Rahmadani | P  | Jl.Malaka 342  | Jakarta  | 021-555098 | C102  |
| <input type="checkbox"/> |  |  |  | 10324 | Martin | Susanto   | L  | Jl.Bima 51     | Jakarta  | 021-555785 | C102  |
| <input type="checkbox"/> |  |  |  | 10407 | Rio    | Gunawan   | L  | Jl.Melati 356  | Surabaya | 031-555231 | C104  |
| <input type="checkbox"/> |  |  |  | 10415 | Susan  | Sumantri  | P  | Jl.Pahlawan 24 | Surabaya | 031-555120 | C104  |

Check all With selected:

After

|  | <input type="checkbox"/> |
|--|--------------------------|--------------------------|--------------------------|--------------------------|--------------------------|--------------------------|--------------------------|--------------------------|--------------------------|--------------------------|
|  |                          |                          | IdPeg                    | Nama                     | NamaKel                  | JK                       | Alamat                   | Kota                     | Telp                     | NoCab                    |
|  | <input type="checkbox"/> |                          |                          |                          | 10107                    | Rahmat                   | Salsalina                | P                        | Jl.Suci 78               | Bandung 022-555768 C101  |
|  | <input type="checkbox"/> |                          |                          |                          | 10176                    | Fahri                    | Wahyuni                  | P                        | Jl.Maluku 56             | Bandung 022-555934 C101  |
|  | <input type="checkbox"/> |                          |                          |                          | 10246                    | Dian                     | Anggraini                | P                        | Jl.Mawar 5               | Semarang 024-555102 C103 |
|  | <input type="checkbox"/> |                          |                          |                          | 10252                    | Antoni                   | Irawan                   | L                        | Jl.A.Yani 15             | Jakarta 021-555888 C102  |
|  | <input type="checkbox"/> |                          |                          |                          | 10307                    | Erik                     | Andrian                  | L                        | Jl.Manggis 155           | Semarang 024-555236 C103 |
|  | <input type="checkbox"/> |                          |                          |                          | 10314                    | Ayu                      | Rahmadani                | P                        | Jl.Malaka 342            | Jakarta 021-555098 C102  |
|  | <input type="checkbox"/> |                          |                          |                          | 10324                    | Martin                   | Susanto                  | L                        | Jl.Bima 51               | Jakarta 021-555785 C102  |
|  | <input type="checkbox"/> |                          |                          |                          | 10407                    | Rio                      | Gunawan                  | L                        | Jl.Melati 356            | Surabaya 031-555231 C104 |
|  | <input type="checkbox"/> |                          |                          |                          | 10415                    | Susan                    | Sumantri                 | P                        | Jl.Pahlawan 24           | Surabaya 031-555120 C104 |

## Cabang

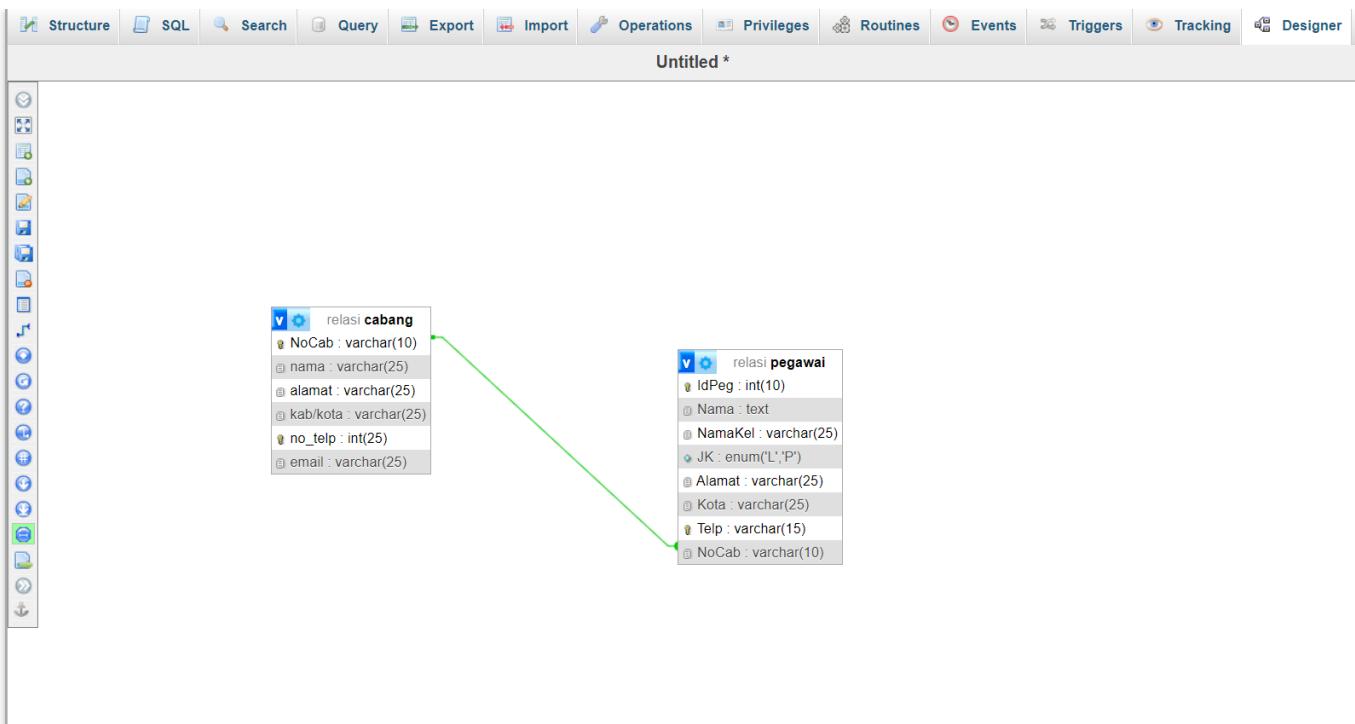
Before

|  | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | NoCab | nama    | alamat           | kab/kota | no_telp   | email               |
|--|--------------------------|--------------------------|--------------------------|--------------------------|-------|---------|------------------|----------|-----------|---------------------|
|  | <input type="checkbox"/> |                          |                          |                          | C101  | Rahmat  | Jl Korban        | Makassar | 812489410 | nurrahmat@gmail.com |
|  | <input type="checkbox"/> |                          |                          |                          | C102  | Fachri  | Jl Singa         | Makassar | 815672432 | fachri@gmail.com    |
|  | <input type="checkbox"/> |                          |                          |                          | C103  | Hayril  | Jl Muh Jufri     | Makassar | 892503945 | hayi@gmail.com      |
|  | <input type="checkbox"/> |                          |                          |                          | C104  | Bom Bom | Jl Sungai Preman | Makassar | 825671234 | sir@gmail.com       |

After

|  | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | NoCab | nama    | alamat           | kab/kota | no_telp   | email           |
|--|--------------------------|--------------------------|--------------------------|--------------------------|-------|---------|------------------|----------|-----------|-----------------|
|  | <input type="checkbox"/> |                          |                          |                          | C101  | Nasir   | Jl Regge         | Makassar | 781263812 | nasir@gmail.com |
|  | <input type="checkbox"/> |                          |                          |                          | C102  | Hansar  | Jl Maccini       | Makassar | 815672432 | ansar@gmail.com |
|  | <input type="checkbox"/> |                          |                          |                          | C103  | Hayril  | Jl Muh Jufri     | Makassar | 892503945 | hayi@gmail.com  |
|  | <input type="checkbox"/> |                          |                          |                          | C104  | Bom Bom | Jl Sungai Preman | Makassar | 825671234 | sir@gmail.com   |

## Hasil Relasi Pegawai & Cabang



## Query & Hasil

The screenshot shows the MySQL Workbench interface with the 'Foreign key constraints' section selected. A constraint named 'pegawai\_ibfk\_1' is listed with the following properties: Action: Drop, On Delete: CASCADE, On Update: CASCADE. The Foreign key constraint (INNODB) is defined between the 'Database' (relasi) and 'Table' (cabang) columns, both labeled 'NoCab'.

## Group BY and Having

1. tampilkan jumlah data mobil dan kelompok kan berdasarkan warna nya sesuai dengan tabel mobil kalian.

### Struktur Query

```
select data 2,COUNT(Data 1) AS nama_sementara FROM nama_tabel GROUP BY data 2;
```

### Query

```
SELECT warna,
COUNT(id_mobil) AS jumlah_warna
FROM mobil GROUP BY warna;
```

### Hasil

```

MariaDB [rental_rahmat]> SELECT warna,
-> COUNT(id_mobil) AS jumlah_warna
-> FROM mobil GROUP BY warna;
+-----+-----+
| warna | jumlah_warna |
+-----+-----+
| Hitam | 3 |
| Merah | 1 |
| Silver | 1 |
+-----+-----+
3 rows in set (0.006 sec)

```

## Analisis

`SELECT warna` nama kolom yang akan diambil dari tabel `mobil`.`COUNT(id_mobil)` AS `jumlah_warna`, `COUNT` digunakan untuk menghitung jumlah baris oleh `GROUP BY`. Di sini, `COUNT(id_mobil)` menghitung jumlah mobil untuk setiap warna, `jumlah_warna`. `FROM mobil` ini adalah nama tabel yang akan digunakan. `GROUP BY warna;` digunakan untuk mengelompokkan hasil query berdasarkan nilai dalam kolom warna. semua baris yang memiliki warna yang sama akan dikelompokkan, dan `COUNT(id_mobil)` akan dihitung.

## kesimpulan

Kesimpulannya, Query ini menghitung jumlah mobil berdasarkan warna dalam tabel `data_mobil`.

2. berdasarkan query ini tampilkan yang lebih BESAR dari 3 atau sama dengan 3 pemilik mobil nya

## Struktur Query

```

select data 3,COUNT(id_mobil) AS nama_sementara from nama_tabel GROUP BY data 3 HAVING
COUNT(data 1) >= 3;

```

## Query

```

select pemilik,COUNT(id_mobil) AS jumlah_mobil from mobil GROUP BY pemilik HAVING
COUNT(id_mobil) >= 3;

```

## Hasil

```

MariaDB [rental_rahmat]> select pemilik,COUNT(id_mobil) AS jumlah_mobil from mobil GROUP BY pemilik HAVING COUNT(id_mobil) >= 3;
+-----+-----+
| pemilik | jumlah_mobil |
+-----+-----+
| Rahmat | 3 |
+-----+-----+
1 row in set (0.005 sec)

```

---

## Analisis

```
select pemilik adalah nama kolomnya yang digunakan, count(id_mobil) digunakan untuk menghitung jumlah baris yang memiliki nilai pada kolom id_mobil AS jumlah_mobil Alias yang memberikan untuk nama sementara pada hasil dari count(id_mobil) jadi hasilnya akan diberi nama jumlah_mobil. from mobil adalah nama tabelnya. GROUP BY pemilik kolom yang digunakan untuk di kelompokkan. HAVING COUNT(id_mobil) >= 3 , HAVING digunakan untuk menetapkan kondisi pada hasil pengelompokan GROUP BY .(id_mobil) >= 3 Kondisi yang harus dipenuhi. Hanya grup dengan jumlah id_mobil yang dihitung lebih besar atau sama dengan 3 yang akan ditampilkan dalam hasil akhir.
```

## kesimpulan

Kesimpulannya, query SQL ini digunakan untuk menemukan pemilik yang memiliki setidaknya tiga mobil.

3.tampilkan semua pemilik dengan jumlah mobilnya yang memiliki atau sama dengan 3 mobil

## Struktur query

```
SELECT nama_data,COUNT(nama_data) AS nama_sementara FROM nama_tabel GROUP BY nama_data;
```

## Query

```
SELECT pemilik,  
COUNT(id_mobil) AS jumlah_mobil  
FROM mobil GROUP BY pemilik;
```

## Hasil

```
MariaDB [rental_rahamat]> SELECT pemilik,  
-> COUNT(id_mobil) AS jumlah_mobil  
-> FROM mobil GROUP BY pemilik;  
+-----+-----+  
| pemilik | jumlah_mobil |  
+-----+-----+  
| Ibe | 2 |  
| Rahmat | 3 |  
+-----+-----+  
2 rows in set (0.002 sec)
```

---

## Analisis

```
SELECT pemilik ini adalah nama kolomnya, count(id_mobil) digunakan untuk menghitung jumlah baris dalam kolom id_mobil untuk setiap grup. Fungsi ini menghitung berapa banyak mobil yang dimiliki oleh setiap pemilik. AS jumlah_mobil memberikan nama sementara pada hasil dari COUNT(id_mobil), sehingga hasilnya akan diberi nama JUMLAH_MOBIL. FROM mobil ini adalah nama tabelnya. GROUP BY pemilik digunakan untuk mengelompokkan hasil query satu atau lebih kolom pemilik.
```

## kesimpulan

Kesimpulannya, Query ini mengelompokkan data berdasarkan pemilik mobil dan menghitung jumlah mobil yang dimiliki setiap pemilik.

4. berdasarkan query yang ada pada praktikum 5 bagian 7 tampilkan data pada table mobil dengan mengelompokkan berdasarkan pemiliknya. hitung menggunakan sum total pendapatan pemilik berdasarkan harga rental

## Struktur Query

```
select data 3, SUM(data 5) AS nama_sementara from nama_tabel GROUP BY data 3;
```

## Query

```
SELECT pemilik,  
SUM(harga_rental) AS total_pendapatan  
FROM mobil GROUP BY pemilik;
```

## Hasil

```
MariaDB [rental_rahamat]> SELECT pemilik,  
-> SUM(harga_rental) AS total_pendapatan  
-> FROM mobil GROUP BY pemilik;  
+-----+  
| pemilik | total_pendapatan |  
+-----+  
| Ibe    |      250000 |  
| Rahmat |      200000 |  
+-----+  
2 rows in set (0.006 sec)
```

---

## Analisis

```
Select pemilik ini adalah nama kolomnya, SUM(harga_rental) digunakan untuk menghitung total nilai dari kolom harga_rental untuk setiap grup pemilik. AS total_pendapatan memberikan nama sementara pada hasil dari SUM(harga_rental), sehingga hasilnya akan diberi nama
```

`total_pendapatan`.`FROM mobil` adalah nama tabelnya, `GROUP BY pemilik` memastikan bahwa hasil query dikelompokkan berdasarkan pemilik mobil, sehingga setiap baris hasil akan menunjukkan pemilik serta total pendapatan rental mereka.

## kesimpulan

Kesimpulannya, Query ini memberikan informasi tentang total pendapatan rental mobil yang diperoleh oleh masing-masing pemilik mobil.

5. Berdasarkan praktikum 5 query no 8 tampilkan jumlah pemasukan pemilik berdasarkan harga rental kelompokkan berdasarkan pemiliknya dan seleksi yang total pemasukannya atau harga rentalnya mencapai lebih besar atau sama dengan 300k

## Struktur Query

```
select data_mobil, SUM(data_mobil) AS nama_sementara from nama_tabel GROUP BY data_mobil  
HAVING SUM(data_mobil) >= 300000;
```

## Query

```
SELECT pemilik,  
SUM(harga_rental) AS total_pemasukan  
FROM mobil GROUP BY pemilik  
HAVING SUM(harga_rental) >= 300000;
```

## Hasil

```
MariaDB [rental_rahamat]> SELECT pemilik,  
-> SUM(harga_rental) AS total_pemasukan  
-> FROM mobil GROUP BY pemilik  
-> HAVING SUM(harga_rental) >= 300000;  
+-----+  
| pemilik | total_pemasukan |  
+-----+  
| Valen   |          600000 |  
+-----+  
1 row in set (0.002 sec)
```

---

## Analisis

`SELECT pemilik` nama kolom yang akan di gunakan, `SUM(harga_rental)` digunakan untuk menghitung total nilai dari kolom `harga_rental` untuk setiap grup pemilik. Fungsi ini menjumlahkan pendapatan dari rental mobil untuk setiap pemilik. `AS total_pemasukan` memberikan nama sementara pada hasil

dari `SUM(harga_rental)`, sehingga hasilnya akan diberi nama `total_pemasukan`. `From mobil` ini adalah nama tabelnya. `GROUP BY pemilik` pengelompokan Setiap nilai unik di kolom pemilik akan membentuk satu grup. `HAVING SUM(harga_rental) >= 300000` Mengelompokkan data berdasarkan kolom pemilik, sehingga setiap pemilik membentuk satu grup serta, menyertakan pemilik yang memiliki total pendapatan dari rental mobil sebesar `300,000` atau lebih.

## kesimpulan

Kesimpulannya, Query ini digunakan untuk mencari pemilik mobil yang memiliki total pendapatan dari rental mobil sebesar 300,000 atau lebih.

6. Berdasarkan praktikum 6 no 12 tampilkan rata rata pemasukan pemilik mobil kelompokkan berdasarkan pemiliknya

## Struktur Query

```
select nama_data, AVG(nama_data) AS nama_sementara from nama_tabel GROUP BY nama_data;
```

## Query

```
SELECT pemilik,  
AVG(harga_rental) AS total_pemasukan  
FROM mobil GROUP BY pemilik;
```

## Hasil

```
MariaDB [rental_rahamat]> SELECT pemilik,  
    -> AVG(harga_rental) AS total_pemasukan  
    -> FROM mobil GROUP BY pemilik;  
+-----+  
| pemilik | total_pemasukan |  
+-----+  
| Ibe     |      125000.0000 |  
| Rahmat  |      66666.6667 |  
| Valen   |      300000.0000 |  
+-----+  
3 rows in set (0.005 sec)
```

---

## Analisis

`SELECT pemilik` nama kolomnya, `AVG(harga_rental)` menghitung nilai rata-rata dari kolom `harga_rental` untuk setiap kolom `pemilik`. `AS total_pemasukan` memberikan nama sementara pada hasil dari `AVG(harga_rental)`, sehingga hasilnya akan diberi nama `rata_rata_pendapatan`. `from mobil` nama tabelnya, `GROUP BY pemilik` `GROUP BY` digunakan untuk mengelompokkan data berdasarkan kolom `pemilik`. Setiap nilai unik di kolom `pemilik` akan membentuk satu grup.

Data akan dikelompokkan berdasarkan pemilik mobil, sehingga rata-rata `harga_rental` dihitung untuk setiap `pemilik`.

## kesimpulan

Kesimpulannya, Query ini digunakan untuk menghitung rata-rata pendapatan dari rental mobil untuk setiap pemilik

7. Berdasarkan praktikum 5 no 16 tampilkan pemasukan terbesar dan pemasukan terkecil kelompokan berdasarkan pemiliknya dan seleksi data pemilik yg tampil atau memiliki jumlah mobil lebih besar dari 1.

## Struktur

```
select nama_data, MAX(nama_data) AS nama_sementara, MIN(nama_data) AS nama_sementara from  
nama_tabel GROUP BY nama_data HAVING COUNT(nama_data) >= 1;
```

## Query

```
SELECT pemilik,  
MAX(harga_rental) AS pemasukan_terbesar,  
MIN(harga_rental) AS pemasukan_terkecil  
FROM mobil GROUP BY pemilik  
HAVING COUNT(id_mobil) > 1;
```

## Hasil

```
MariaDB [rental_rahamat]> SELECT pemilik,  
-> MAX(harga_rental) AS pemasukan_terbesar,  
-> MIN(harga_rental) AS pemasukan_terkecil  
-> FROM mobil GROUP BY pemilik  
-> HAVING COUNT(id_mobil) > 1;  
+-----+-----+-----+  
| pemilik | pemasukan_terbesar | pemasukan_terkecil |  
+-----+-----+-----+  
| Ibe    |      150000 |      100000 |  
| Rahmat |      100000 |       50000 |  
| Valen  |      450000 |      150000 |  
+-----+-----+-----+  
3 rows in set (0.006 sec)
```

## Analisis

`SELECT pemilik` nama kolomnya, `max(harga_rental)` menghitung pendapatan terbesar dari rental mobil untuk setiap pemilik. `AS pemasukan_terbesar` memberikan nama sementara pada hasil dari

```
MAX(harga_rental), MIN(harga_rental)
```

## kesimpulan

Kesimpulannya, kita bisa melihat daftar pemilik, pemasukan terbesar, dan pemasukan terkecil.