

# 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 menunjukkan 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".

## STRUKTUR

---

```
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 | 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]>

```

## 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 ditujukkannya.
- ◆ **hitam** adalah nama data pada sebuah kolom
- ◆ **OR** adalah sebuah perintah query ATAU menunjukkan 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.

# STRUKTUR

---

```
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:

---

```
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]>
```

## 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 ditujukkannya.
- ♦ **MOBIL** adalah sebuah nama tabel.
- ♦ **BETWEEN** adalah sebuah perintah query ANTARA menunjukkan bahwa pemilihan data yang ditampilkan adalah DI ANTARA salah satu data dari syarat yang ditentukan.
- ♦ **AND** adalah sebuah perintah query DAN menunjukkan 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  
+-----+-----+-----+-----+-----+-----+-----+-----+  
| 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 ditujukkannya.
- ♦ **MOBIL** adalah sebuah nama tabel.
- ♦ **NOT BETWEEN** adalah sebuah perintah query ANTARA menunjukkan bahwa pemilihan data yang ditampilkan adalah TIDAK DI ANTARA salah satu data dari syarat yang ditentukan.
- ♦ **AND** adalah sebuah perintah query DAN menunjukkan 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 bertugas 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 bertugas menentukan syarat *lebih dari atau sama dengan*.

## KESIMPULAN:

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

```
C:\XAMPP for Windows - MySQL -u root -p
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 (0.001 sec)

MariaDB [rental_hayril]> SELECT nama from akun WHERE password="12345";
+-----+
| nama |
+-----+
| Githa |
+-----+
1 row in set (0.001 sec)

MariaDB [rental_hayril]> _
```

## 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 bertugas 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_hayril]> 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_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 bertugas menentukan seluruh data yang sesuai dengan kondisi ditentukan, menghindari penulisan berulang query dengan kondisi yang sama.
- ♦ **AND** adalah sebuah perintah query DAN menunjukkan 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 berondisikan 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 bertugas menentukan seluruh data yang sesuai dengan kondisi ditentukan, menghindari penulisan berulang query dengan kondisi yang sama.
- ♦ **OR** adalah sebuah perintah query ATAU menunjukkan 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:

```
3 rows in set (0.001 sec)

MariaDB [rental_hayril]> select * from data_mobil
  -> where warna in ("hitam","silver")
  -> and harga_rental > 50000;
+-----+-----+-----+-----+-----+-----+-----+
| 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 bertugas menentukan seluruh data yang sesuai dengan kondisi ditentukan, menghindari penulisan berulang query dengan kondisi yang sama.
- ♦ **DAN** adalah sebuah perintah query DAN menunjukkan 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:

---

```
Cat. 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 bertugas 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 berpola 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 bertugas 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_hayril]> 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_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 bertugas 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

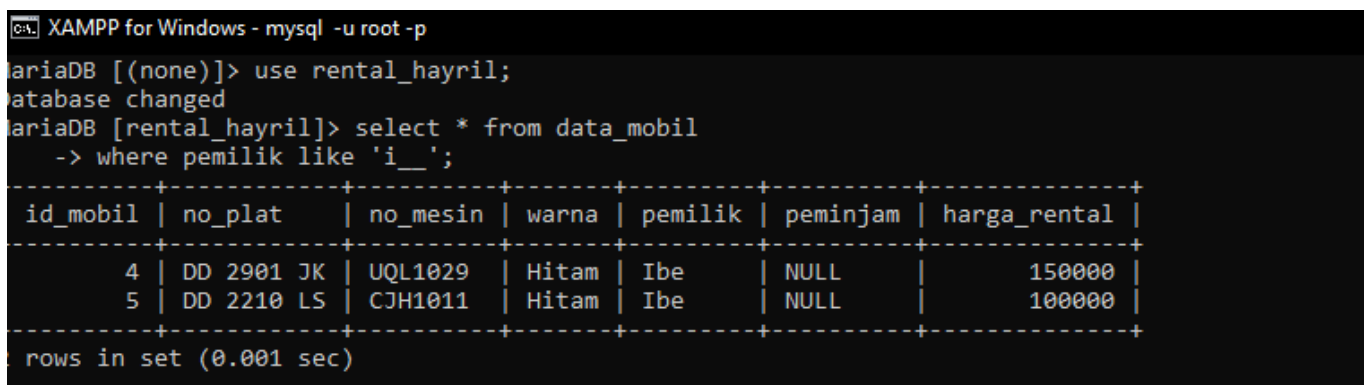
### STRUKTUR:

```
select * from nama_tabel
→ where nama_kolom like 'nama_pola';
```

### CONTOH:

```
select * from data_mobil
→ where pemilik like 'i__';
```

HASIL:



```

C:\> XAMPP for Windows - mysql -u root -p
MariaDB [(none)]> use rental_hayril;
Database changed
MariaDB [rental_hayril]> 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 | 150000 |
| 5 | DD 2210 LS | CJH1011 | Hitam | Ibe | NULL | 100000 |
+-----+-----+-----+-----+-----+-----+-----+
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 bertugas 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:

---

```

C:\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 | UYN1312 | 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 bertugas 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 bertugas 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 bertugas 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 | 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 ditujukkannya.
- ♦ **DATA\_MOBIL** adalah sebuah nama tabel.
- ♦ **PEMINJAM** adalah sebuah nama kolom.
- ♦ **IS NOT NULL** adalah perintah query yang bertugas 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	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 bertugas 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 bertugas 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 bertugas 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 bertugas 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 ("PENHUBUNG",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 synta
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_VIRTUAL;
```

## 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_VIRTUAL;
```

## 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.

## 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.

## Tantangan Table Virtual

1. buat kan table virtual yang peminjamnya tidak ada dan tampilkan datanya

### QUERY:

```
CREATE VIEW mobil_Tanpa_peminjam AS Select no_plat,peminjam From mobil WHERE peminjam IS NULL ;
```

### HASIL:

```
ERROR 1146 (42S02): Table 'rental_hayril.mobil' doesn't exist
MariaDB [rental_hayril]> CREATE VIEW mobil_Tanpa_peminjam AS Select no_plat,peminjam From data_mobil WHERE peminjam IS NULL ;
Query OK, 0 rows affected (0.133 sec)

MariaDB [rental_hayril]> select * from mobil_tanpa_peminjam;
+-----+-----+
| no_plat | peminjam |
+-----+-----+
| DD 2901 JK | NULL |
| DD 2210 LS | NULL |
| DD 2812 UY | NULL |
+-----+-----+
3 rows in set (0.006 sec)

MariaDB [rental_hayril]> _
```

### ANALISIS:

1. **CREATE VIEW mobil\_Tanpa\_peminjam AS** : Perintah ini digunakan untuk membuat view baru dengan nama **mobil\_Tanpa\_peminjam**.
2. **SELECT no\_plat, peminjam FROM mobil WHERE peminjam IS NULL;** : Query ini akan mengambil nilai dari kolom **no\_plat** dan **peminjam** dari tabel **mobil** dimana nilai **peminjam** adalah NULL.

### KESIMPULAN:

Perintah **CREATE VIEW mobil\_Tanpa\_peminjam AS** digunakan untuk membuat view baru dengan nama **mobil\_Tanpa\_peminjam**. View ini akan menampilkan semua data mobil yang tidak memiliki peminjam. Query **SELECT no\_plat, peminjam FROM mobil**



`WHERE peminjam IS NULL;` akan mengambil nilai dari kolom `no_plat` dan `peminjam` dari tabel `mobil` dimana nilai `peminjam` adalah NULL.

2. ganti salah satu data peminjam dari table mobil dengan nilai null dan tampilkan isi data pada table virtual hasil nya akan tiga data yang tampil

## QUERY:

```
UPDATE mobil SET peminjam = NULL where peminjam = 'Elia' ;
```

## 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	NULL	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]>
```

## ANALISIS:

1. `UPDATE mobil` : Perintah ini digunakan untuk mengupdate data dalam tabel `mobil`.
2. `SET peminjam = NULL` : Bagian ini menentukan bahwa nilai kolom `peminjam` akan diubah menjadi NULL.
3. `WHERE peminjam = 'Elia'` : Kondisi ini membatasi pengubahan hanya untuk baris-baris dimana nilai kolom `peminjam` adalah 'Elia'.

## KESIMPULAN:

perintah `UPDATE mobil SET peminjam = NULL WHERE peminjam = 'Elia'` akan mengubah nilai kolom `peminjam` menjadi NULL hanya untuk baris-baris di mana nilai kolom `peminjam` adalah 'Elia' dalam tabel `mobil`

3. berikan kesimpulan mengapa table virtual ini di buat.

## KESIMPULAN:

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.

## Relasi Tabel

### Data Tabel Pegawai

Browse

Structure

SQL

Search

Insert

Export

Import

Privileges

Operations

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

<

## Update Struktur Table

### Before

localhost/phpmyadmin/index.php?route=/table/...

Server: 127.0.0.1 » Database: relasi\_tabel » Table: pegawai

Table structure Relation view

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

Check all With selected: Browse Change Drop Primary Unique Index Spatial Fulltext

Print Propose table structure Move columns Normalize

Add 1 column(s) after NoCab Go

Indexes

Console

2:20 AM 6/26/2024

After

localhost/phpmyadmin/index.php?route=/table/...

Server: 127.0.0.1 » Database: relasi\_tabel » Table: pegawai

✓ Table pegawai has been altered successfully.

```
ALTER TABLE `pegawai` CHANGE `NamaKel` `NamaKel` 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
2	Nama	varchar(25)	utf8mb4_general_ci		No	None			Change
3	NamaKel	text	utf8mb4_general_ci		No	None			Change
4	JK	enum('L', 'P', ' ', '')	utf8mb4_general_ci		No	None			Change
5	Alamat	varchar(25)	utf8mb4_general_ci		No	None			Change
6	Kota	varchar(25)	utf8mb4_general_ci		No	None			Change
7	Telp	varchar(15)	utf8mb4_general_ci		No	None			Change
8	NoCab	varchar(10)	utf8mb4_general_ci		No	None			Change

Check all With selected: Browse Change Drop Primary Unique Index Spatial Fulltext

Console

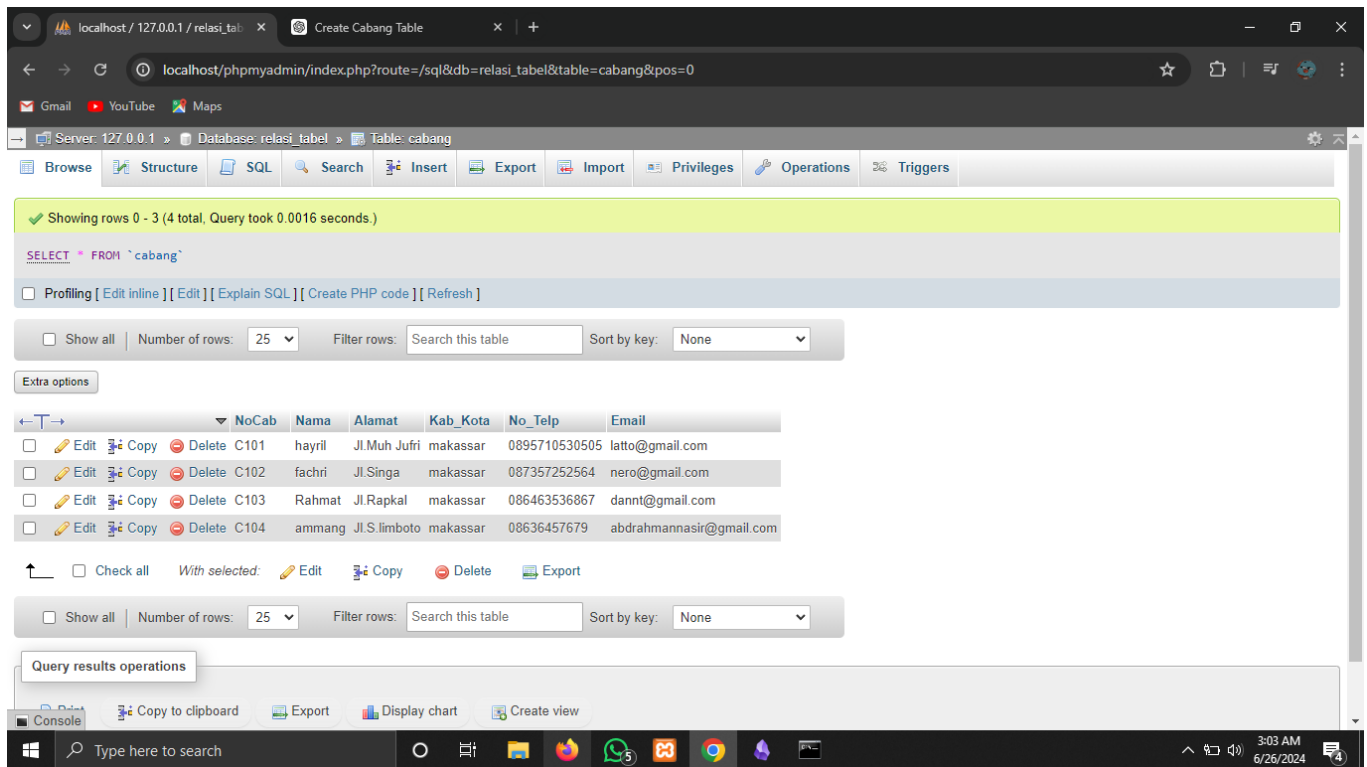
2:23 AM 6/26/2024

## Update Tabel Pegawai dan Cabang

### Pegawai

### Before





## After

Extra options

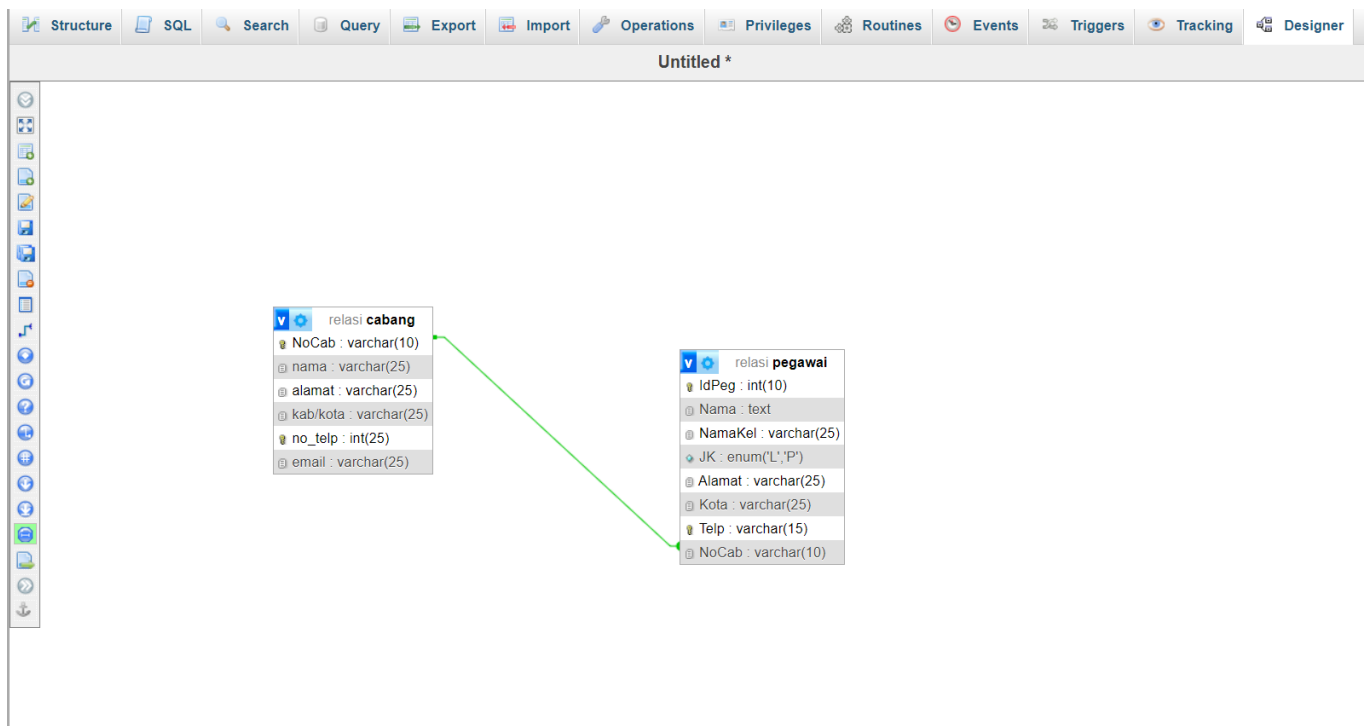
NoCab	Nama	Alamat	Kab_Kota	No_Telp	Email
C101	hayril	Jl.Muh Jufri	makassar	0895710530505	latto@gmail.com
C102	fachri	Jl.Singa	makassar	087357252564	nero@gmail.com
C103	rahmat	Jl.Rapkal	makassar	086463536867	dannt@gmail.com
C104	abd rahman	Jl.S.limboto	makassar	08636457679	abdrahmannasir@gmail.com

Check all With selected: Edit Copy Delete Export

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

## Relasi Tabel Pegawai dan Cabang





## Query

Table structure Relation view

Foreign key constraints

Actions	Constraint properties	Column	Foreign key constraint (INNODB)		
			Database	Table	Column
<input type="button" value="Drop"/>	pegawai_ibfk_1 ON DELETE <input type="button" value="CASCADE"/> ON UPDATE <input type="button" value="CASCADE"/>	NoCab	relasi	cabang	NoCab

+ Add column

## Group By and Having

1. Tampilkan Jumlah data mobil dan kelompok berdasarkan warna nya sesuai dengan tabel mobil kalian

## Struktur

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

## Contoh

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

## Hasil

---

```
rows in set (0.000 sec)  
MariaDB [rental_hayril]> SELECT warna,COUNT(id_mobil) AS jumlah_warna FROM data_mobil GROUP BY warna;  
+-----+-----+  
| warna | jumlah_warna |  
+-----+-----+  
| HITAM |          5 |  
| Merah |          1 |  
| Silver |          1 |  
+-----+-----+  
3 rows in set (0.001 sec)
```

## Analisis

---

- ♦ `SELECT warna, COUNT(id_mobil) AS jumlah_warna` : Memilih kolom `warna` dan menghitung jumlah baris `id_mobil` untuk setiap warna.
- ♦ `FROM mobil` : Menentukan tabel yang ingin di tampilkan, yaitu `data_mobil`.
- ♦ `GROUP BY warna` : Mengelompokkan hasil berdasarkan kolom `warna`.

## Kesimpulan

---

Query di atas berfungsi untuk mengelompokkan hasil sesuai dengan jumlah mobil berdasarkan warnanya.

## 2. Berdasarkan Query in tampilkan yang lebih besar dari 3 atau sama dengan 3 pemilik mobilnya

---

### Struktur

---

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

## Contoh

---

```
select pemilik,COUNT(id_mobil) AS jumlah_mobil from data_mobil GROUP  
BY pemilik HAVING COUNT(id_mobil) ≥ 3;
```



# Hasil

---

```
MariaDB [rental_hayril]> select pemilik,COUNT(id_mobil) AS jumlah_mobil from data_mobil GROUP BY pemilik HAVING COUNT(id_mobil) >= 3;
Empty set (0.002 sec)

MariaDB [rental_hayril]> select pemilik,COUNT(id_mobil) AS jumlah_mobil from data_mobil GROUP BY pemilik HAVING COUNT(id_mobil) >= 3;
+-----+-----+
| pemilik | jumlah_mobil |
+-----+-----+
| Ibe     | 3             |
+-----+-----+
1 row in set (0.002 sec)

MariaDB [rental_hayril]>
```

## Analisis

---

- ◆ **SELECT** : Perintah untuk memilih kolom yang akan ditampilkan dalam hasil query.
- ◆ **pemilik** : Kolom yang menyimpan informasi tentang pemilik mobil.
- ◆ **COUNT(id\_mobil) AS jumlah\_mobil** : Menghitung jumlah baris dalam setiap grup yang memiliki nilai **id\_mobil** dan memberikan alias **jumlah\_mobil** pada hasil hitungan tersebut.
- ◆ **FROM** : Menentukan tabel sumber data yang digunakan dalam query.
- ◆ **data\_mobil** : Nama tabel yang berisi data mobil termasuk informasi tentang pemilik dan id mobil.
- ◆ **GROUP BY** : Mengelompokkan hasil query berdasarkan kolom tertentu.
- ◆ **pemilik** : Kolom yang digunakan untuk mengelompokkan data.
- ◆ **HAVING** : Klausul yang digunakan untuk menyaring grup berdasarkan hasil agregat.
- ◆ **COUNT(id\_mobil) ≥ 3** : Kondisi penyaringan yang hanya menyertakan grup di mana jumlah **id\_mobil** adalah 3 atau lebih.

## Kesimpulan

---

Query di atas adalah perintah untuk mengelompokkan nama pemilik yang memiliki jumlah mobil lebih atau sama dengan 3.

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

---

## Struktur

---

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

## Contoh

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

## Hasil

```
1 row in set (0.002 sec)  
MariaDB [rental_hayril]> SELECT pemilik, COUNT(id_mobil) AS jumlah_mobil FROM data_mobil GROUP BY pemilik;  
+-----+-----+  
| pemilik | jumlah_mobil |  
+-----+-----+  
| Baim    | 1            |  
| Hayril  | 2            |  
| Ibe     | 3            |  
| Ibrahim | 1            |  
+-----+-----+  
4 rows in set (0.003 sec)  
MariaDB [rental_hayril]>
```

## Analisis

- ♦ **SELECT** : adalah perintah untuk memilih kolom yang akan ditampilkan
- ♦ **pemilik** : adalah nama kolom yang telah dipilih
- ♦ **COUNT(id\_mobil) AS jumlah\_mobil** : Menghitung jumlah mobil yang dimiliki oleh setiap pemilik dan memberikan alias **jumlah\_mobil** pada hasil hitungan tersebut.
- ♦ **FROM data\_mobil** : adalah perintah untuk mengambil data dari tabel **data\_mobil**.
- ♦ **GROUP BY pemilik** : adalah perintah untuk mengelompokkan hasil berdasarkan kolom **pemilik**.

## Kesimpulan

Query di atas adalah perintah untuk mengelompokkan data pemilik mobil dan menghitung jumlah mobil yang dimiliki setiap **pemilik**

**4. Berdasarkan Query yang ada pada praktikum 5 bagian 7 tampilkan data pada tabel mobil dengan**

# mengelompokkan SUM TOTAL pendapat pemilik berdasarkan harga rental

---

## Struktur

---

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

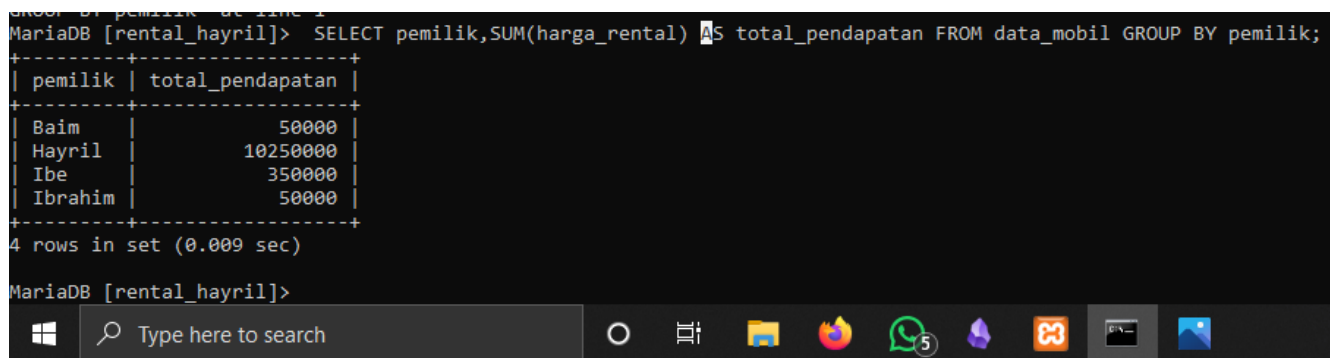
## Contoh

---

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

## Hasil

---



```
MariaDB [rental_hayril]> SELECT pemilik,SUM(harga_rental) AS total_pendapatan FROM data_mobil GROUP BY pemilik;  
+-----+-----+  
| pemilik | total_pendapatan |  
+-----+-----+  
| Baim    | 50000            |  
| Hayril  | 10250000         |  
| Ibe     | 350000           |  
| Ibrahim | 50000            |  
+-----+-----+  
4 rows in set (0.009 sec)  
  
MariaDB [rental_hayril]>
```

## Analisis

---

- ♦ **SELECT** : adalah perintah untuk memilih kolom yang akan ditampilkan
- ♦ **pemilik** : adalah nama kolom yang telah dipilih
- ♦ **SUM(harga\_rental)** : Fungsi perhitungan yang menjumlahkan nilai **harga\_rental** dalam setiap grup.
- ♦ **AS total\_pendapatan** : adalah perintah untuk membuat kolom hasil penjumlahan, membuat hasil lebih mudah dibaca.
- ♦ **FROM data\_mobil** : adalah perintah untuk mengambil data dari tabel **data\_mobil**.
- ♦ **GROUP BY pemilik** : adalah perintah untuk mengelompokkan hasil berdasarkan kolom **pemilik**.

## Kesimpulan

---

Query di atas adalah perintah untuk menjumlahkan setiap data `harga_rental` dan mengelompokkan dalam hasil nya dalam kolom `hasil_pendapatan`.

**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

```
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 data_mobil GROUP BY pemilik  
HAVING SUM(harga_rental) ≥ 300000;
```

## Hasil

```
MariaDB [rental_hayril]> SELECT pemilik,SUM(harga_rental) AS total_pemasukan FROM data_mobil GROUP BY pemilik HAVING SUM(harga_rental) >= 300000;  
+-----+-----+  
| pemilik | total_pemasukan |  
+-----+-----+  
| Hayril  | 10250000        |  
| Ibe     | 350000          |  
+-----+-----+  
2 rows in set (0.034 sec)  
  
MariaDB [rental_hayril]>
```

## Analisis

- ♦ **SELECT** : adalah perintah untuk memilih kolom yang akan ditampilkan

- ♦ **pemilik** : adalah nama kolom yang telah dipilih
- ♦ **SUM(harga\_rental)** : Fungsi perhitungan yang menjumlahkan nilai **harga\_rental** dalam setiap grup.
- ♦ **AS total\_pendapatan** : adalah perintah untuk membuat kolom hasil penjumlahan, membuat hasil lebih mudah dibaca.
- ♦ **FROM data\_mobil** : adalah perintah untuk mengambil data dari tabel **data\_mobil**.
- ♦ **GROUP BY pemilik** : adalah perintah untuk mengelompokkan hasil berdasarkan kolom **pemilik**.
- ♦ **HAVING SUM(harga\_rental) ≥ 300000;** : adalah perintah untuk mengelompokkan dan menyeleksi data kolom **harga\_rental** yang jumlah nya lebih atau sama dengan 300.000

## Kesimpulan

---

Query di atas adalah perintah untuk menghitung jumlah **harga\_rental** setiap **pemilik** dan mengelompokkan dalam kolom **total\_pemasukan** yang jumlahnya lebih atau sama dengan 300.000

## 6. Berdasarkan Praktikum 6 NO.12 Tampilkan Rata Rata pemasukkan pemilik mobil kelompokkan berdasarkan pemiliknya

---

### Struktur

---

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

### Contoh

---

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

### Hasil

---

```
MariaDB [rental_hayril]> SELECT pemilik,AVG(harga_rental) AS total_pemasukan FROM data_mobil GROUP BY pemilik;
```

pemilik	total_pemasukan
Baim	50000.0000
Hayril	5125000.0000
Ibe	116666.6667
Ibrahim	50000.0000

```
4 rows in set (0.034 sec)

MariaDB [rental_hayril]> _
```

## Analisis

- ♦ **SELECT** : adalah perintah untuk memilih kolom yang akan ditampilkan
- ♦ **pemilik** : adalah nama kolom yang telah dipilih
- ♦ **AVG(harga\_rental)** : Fungsi perhitungan yang menghitung nilai rata-rata **harga\_rental** dalam setiap grup.
- ♦ **AS total\_pemasukan** : adalah perintah untuk membuat kolom hasil penjumlahan, membuat hasil lebih mudah dibaca.
- ♦ **FROM data\_mobil** : adalah perintah untuk mengambil data dari tabel **data\_mobil**.
- ♦ **GROUP BY pemilik** : adalah perintah untuk mengelompokkan hasil berdasarkan kolom **pemilik**.

## Kesimpulan

Query di atas adalah perintah untuk menghitung nilai rata rata setiap data **harga\_rental** dan mengelompokkan dalam hasil nya dalam kolom **hasil\_pemasukan**.

**7. Berdasarkan Praktikum 5 NO.16 tampilkan pemasukan terbesar dan pemasukan terkecil kelompokkan berdasarkan pemiliknya dan seleksi data pemilik yang tampil atau 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;
```

## Contoh

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

## Hasil

```
4 rows in set (0.034 sec)  
MariaDB [rental_hayril]> SELECT pemilik, MAX(harga_rental) AS pemasukan_terbesar, MIN(harga_rental) AS pemasukan_terkecil FROM data_mobil GROUP BY pemilik HAVING COUNT(id_mobil) > 1;  
+-----+-----+-----+  
| pemilik | pemasukan_terbesar | pemasukan_terkecil |  
+-----+-----+-----+  
| Hayril  | 1000000            | 250000             |  
| Ibe     | 150000             | 100000             |  
+-----+-----+-----+  
2 rows in set (0.047 sec)  
MariaDB [rental_hayril]>
```

## Analisis

- ♦ **SELECT** : adalah perintah untuk memilih kolom yang akan ditampilkan
- ♦ **pemilik** : adalah nama kolom yang telah dipilih
- ♦ **MAX(harga\_rental)** : Fungsi perhitungan yang menghitung nilai terbesar **harga\_rental** dalam setiap grup.
- ♦ **MIN(harga\_rental)** : Fungsi perhitungan yang menghitung nilai terkecil **harga\_rental** dalam setiap grup.
- ♦ **GROUP BY pemilik** : adalah perintah untuk mengelompokkan hasil berdasarkan kolom **pemilik**.
- ♦ **HAVING** : Klausul yang digunakan untuk menyaring grup berdasarkan hasil agregat.
- ♦ **COUNT(id\_mobil) > 1** : Kondisi penyaringan yang hanya menyertakan grup di mana jumlah **id\_mobil** adalah 1 .

## Kesimpulan