

AND

Penjelasan

1. `select` query yang digunakan untuk menampilkan masukan dari `insert`
2. `warna, pemilik` merupakan nama kolom dari mobil
3. `from` query yang digunakan untuk memberi tanda bahwa tabel mana yang akan di tampilkan
4. `where` query yang digunakan untuk memberikan sebuah kondisi
5. `warna='hitam' and pemilik='ibrahim'` merupakan sebuah kondisi untuk query dan `and` digunakan untuk memberikan syarat yang keduanya harus di penuhi

Struktur Query

```
SELECT kolom1,kolom2 FROM [nama_tabel] WHERE kolom1="nilai1" AND  
kolom2="nilai2";
```

Contoh Query

```
SELECT warna,pemilik FROM mobil WHERE warna="HITAM" AND pemilik="REZKY";
```

Hasil

```
Administrator: XAMPP for Windows - mysql -u root -p
MariaDB [rental_alya]> SELECT warna,pemilik FROM mobil WHERE warna="HITAM" AND pemilik="REZKY"
;
+-----+-----+
| warna | pemilik |
+-----+-----+
| HITAM | REZKY   |
+-----+-----+
1 row in set (0.000 sec)

MariaDB [rental_alya]> _
```

Kesimpulan

Jika ingin menampilkan data yang telah di seleksi dengan cara memberikan syarat yang semuanya harus di penuhi kalian bisa menggunakan query dengan struktur `select kolom1,kolom2 from nama_table where kolom1='nilai_kolom1' and kolom2='nilai_kolom2';`

OR

Penjelasan

1. `select` query yang digunakan untuk menampilkan masukan dari `insert`
2. `warna,pemilik` merupakan nama kolom dari mobil
3. `from` query yang digunakan untuk memberi tanda bahwa tabel mana yang akan di tampilak
4. `where` query yang digunakan untuk memberikan sebuah kondisi
5. `warna='hitam' or pemilik='ibrahim'` merupakan sebuah kondisi untuk query dan `or` digunakan untuk memberikan syarat yang salah satunya harus di penuhi

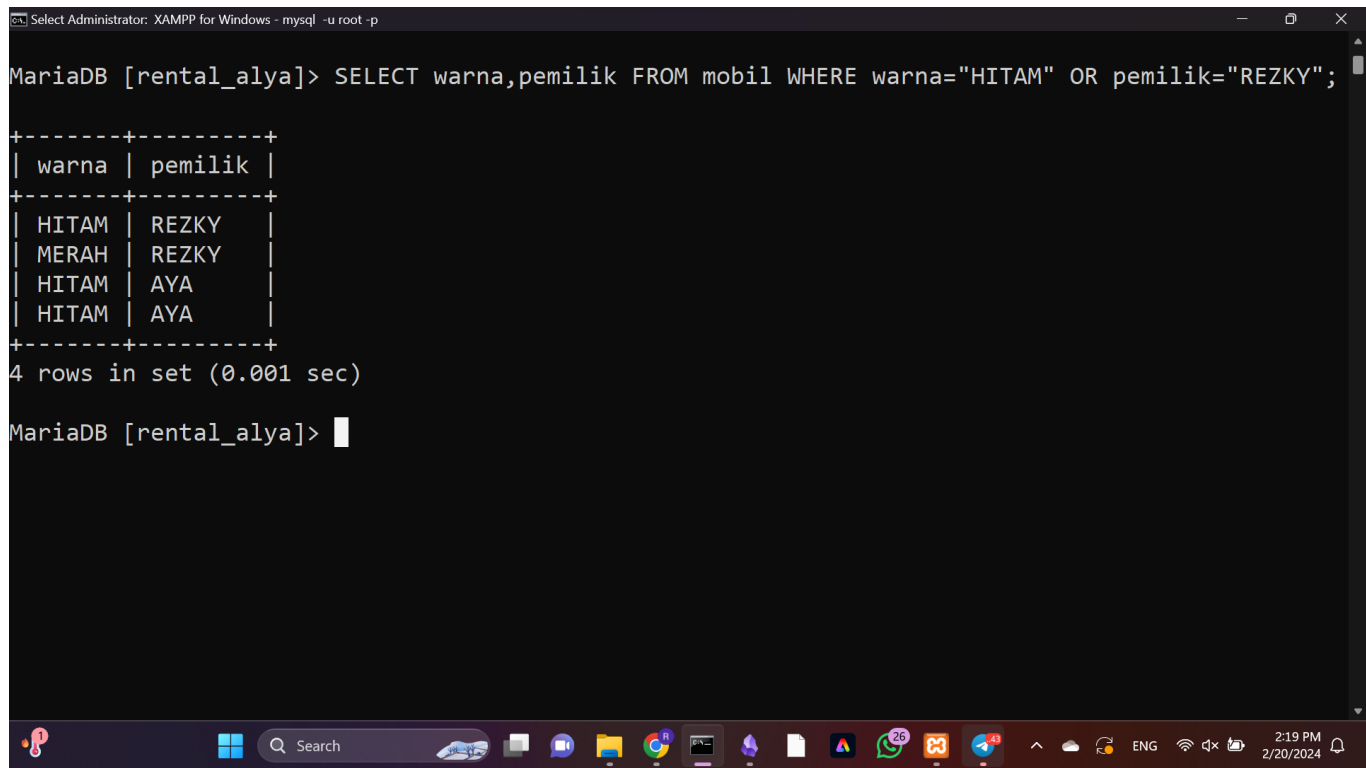
Struktur Query

```
SELECT kolom1,kolom2 FROM [nama_tabel] WHERE kolom1="nilai1" OR
kolom2="nilai2";
```

Contoh Query

```
SELECT warna,pemilik FROM mobil WHERE warna="HITAM" OR pemilik="REZKY";
```

Hasil



The screenshot shows a MySQL command prompt window titled "Select Administrator: XAMPP for Windows - mysql -u root -p". The prompt is "MariaDB [rental_alya]>". The user has entered the query: "SELECT warna,pemilik FROM mobil WHERE warna="HITAM" OR pemilik="REZKY";". The results are displayed in a table format:

warna	pemilik
HITAM	REZKY
MERAH	REZKY
HITAM	AYA
HITAM	AYA

Below the table, it says "4 rows in set (0.001 sec)". The prompt is now "MariaDB [rental_alya]> " followed by a cursor.

Kesimpulan

Jika kalian ingin menampilkan data tabel dari kolom yang nilainya telah di seleksi dengan cara memberikan syarat yang salah satunya harus di penuhi kalian bisa menggunakan query dengan struktur `select warna,pemilik from mobil where warna='HITAM' or pemilik='REZKY';`

BETWEEN

Penjelasan

1. `select` query yang digunakan untuk menampilkan masukan dari `insert`
2. `*` berarti semua kolom akan di tampilkan
3. `from` untuk memberikan tanda bahwa table mana yang akan di tampilkan
4. `mobil` nama table yang akan di tampilkan
5. `where` untuk memberikan sebuah kondisi
6. `harga_rental` nama kolom yang digunakan untuk mengkondisikan sebuah table

7. `between` Ini adalah operator yang digunakan untuk memilih rentang nilai
8. `50000 and 100000` Ini adalah nilai rentang yang digunakan dalam kriteria pemilihan data

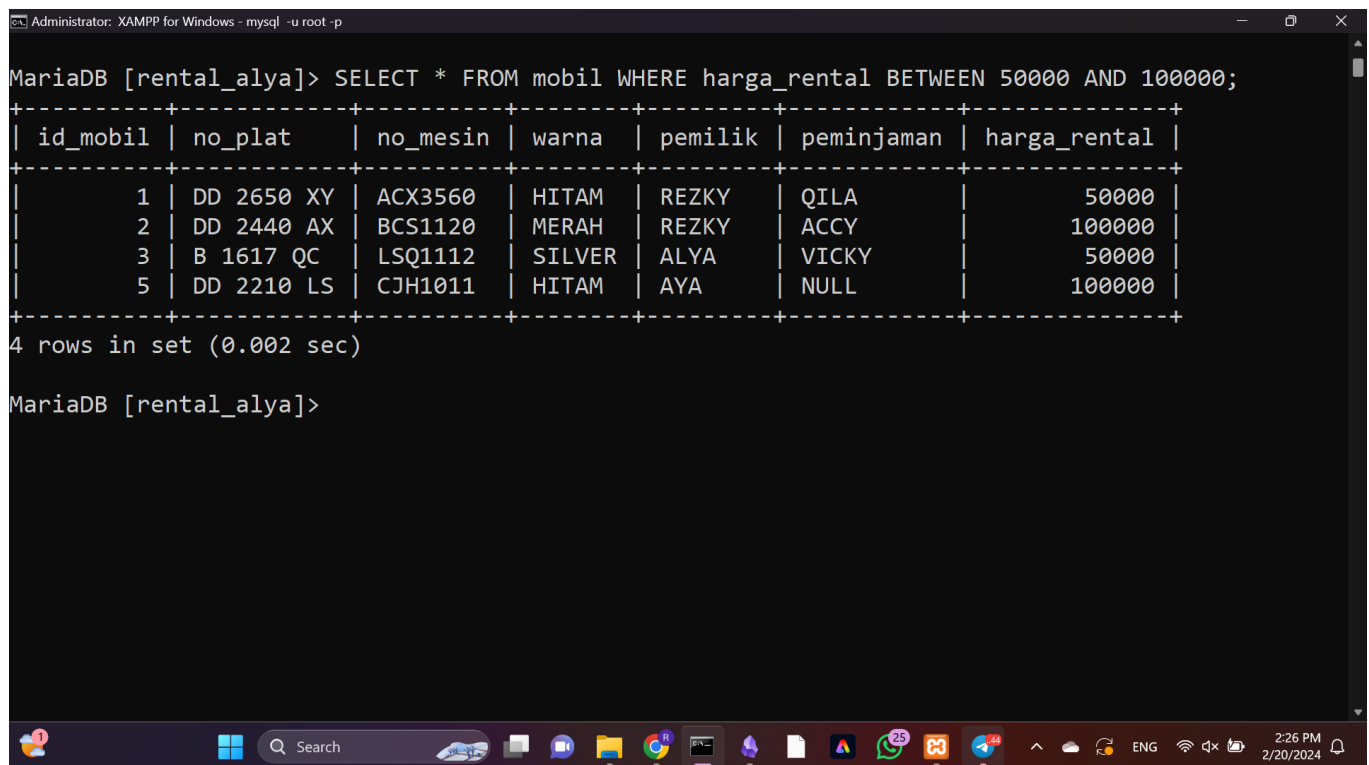
Struktur Query

```
SELECT * FROM [nama_tabel] WHERE kolom1 BETWEEN nilai1 AND nilai2;
```

Contoh Query

```
SELECT * FROM mobil WHERE harga_rental BETWEEN 50000 AND 100000;
```

Hasil



The screenshot shows a MySQL command prompt window titled "Administrator: XAMPP for Windows - mysql -u root -p". The user is in the "rental_alya" database. The query executed is `SELECT * FROM mobil WHERE harga_rental BETWEEN 50000 AND 100000;`. The results are displayed in a table with 7 columns: `id_mobil`, `no_plat`, `no_mesin`, `warna`, `pemilik`, `peminjaman`, and `harga_rental`. There are 4 rows of data. Below the table, it says "4 rows in set (0.002 sec)".

id_mobil	no_plat	no_mesin	warna	pemilik	peminjaman	harga_rental
1	DD 2650 XY	ACX3560	HITAM	REZKY	QILA	50000
2	DD 2440 AX	BCS1120	MERAH	REZKY	ACCY	100000
3	B 1617 QC	LSQ1112	SILVER	ALYA	VICKY	50000
5	DD 2210 LS	CJH1011	HITAM	AYA	NULL	100000

Kesimpulan

Jika ingin menampilkan hasil dari menyeleksi table dengan cara memberikan sebuah rentang nilai kalian bisa menggunakan sebuah query dengan struktur `select * from nama_table where nama_kolom between nilai1 and nilai2;`

NOT BETWEEN

Penjelasan

1. `select` query yang digunakan untuk menampilkan masukan dari `insert`
2. `*` berarti semua kolom akan di tampilkan
3. `from` untuk memberikan tanda bahwa table mana yang akan di tampilkan
4. `mobil` nama table yang akan di tampilkan
5. `where` untuk memberikan sebuah kondisi
6. `harga_rental` nama kolom yang digunakan untuk mengkondisikan sebuah table
7. `not between` Ini adalah operator yang digunakan untuk memilih nilai di luar rentang tertentu.
8. `100000 and 150000` Ini adalah nilai rentang yang digunakan dalam kriteria pemilihan data

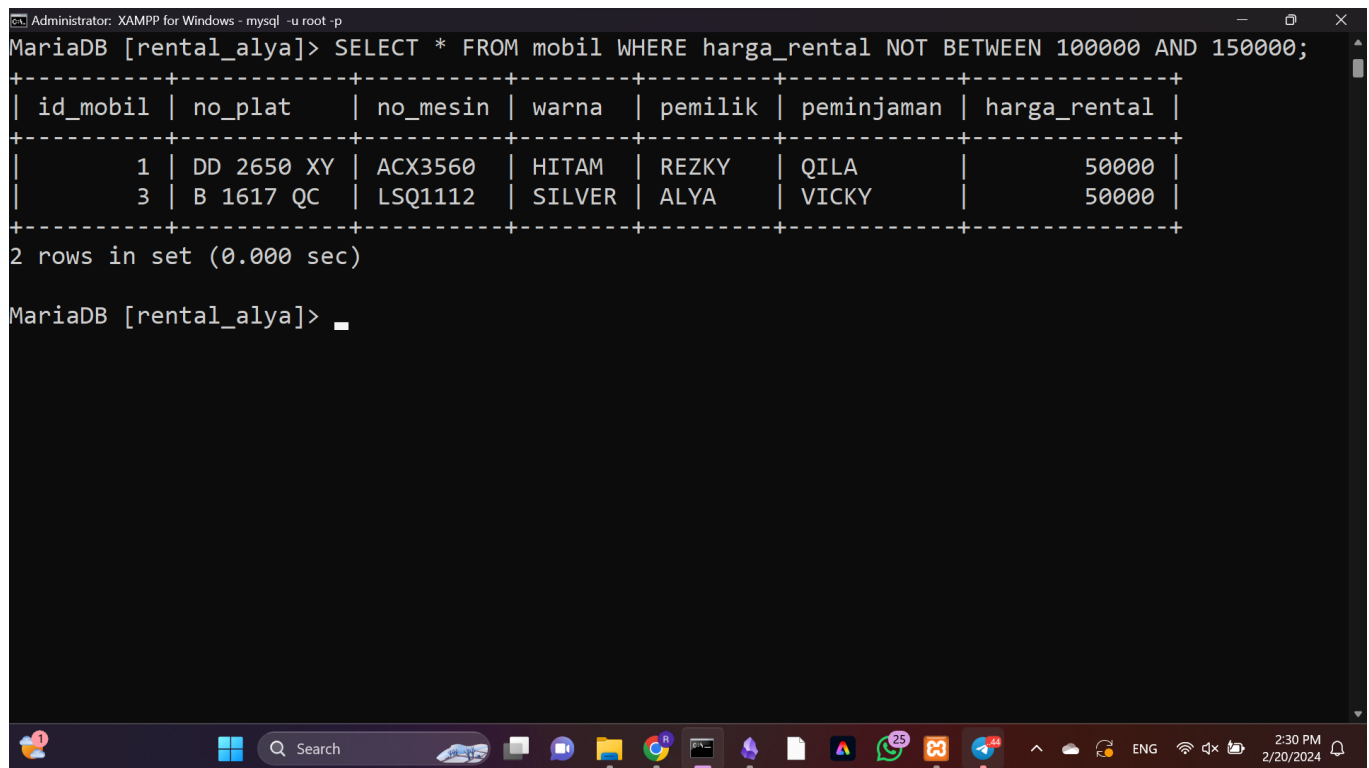
Struktur Query

```
SELECT * FROM [nama_tabel] WHERE kolom1 NOT BETWEEN nilai1 AND nilai2;
```

Contoh Query

```
SELECT * FROM mobil WHERE harga_rental NOT BETWEEN 100000 AND 150000;
```

Hasil



The screenshot shows a Windows command prompt window titled "Administrator: XAMPP for Windows - mysql -u root -p". The prompt is at the MariaDB [rental_alya] database. The user has entered the query: `SELECT * FROM mobil WHERE harga_rental NOT BETWEEN 100000 AND 150000;`. The output shows two rows of data from the 'mobil' table. The first row has id_mobil 1, no_plat DD 2650 XY, no_mesin ACX3560, warna HITAM, pemilik REZKY, peminjaman QILA, and harga_rental 50000. The second row has id_mobil 3, no_plat B 1617 QC, no_mesin LSQ1112, warna SILVER, pemilik ALYA, peminjaman VICKY, and harga_rental 50000. The prompt then returns to the MariaDB [rental_alya] database.

```
Administrator: XAMPP for Windows - mysql -u root -p
MariaDB [rental_alya]> SELECT * FROM mobil WHERE harga_rental NOT BETWEEN 100000 AND 150000;
+-----+-----+-----+-----+-----+-----+-----+
| id_mobil | no_plat | no_mesin | warna | pemilik | peminjaman | harga_rental |
+-----+-----+-----+-----+-----+-----+-----+
| 1 | DD 2650 XY | ACX3560 | HITAM | REZKY | QILA | 50000 |
| 3 | B 1617 QC | LSQ1112 | SILVER | ALYA | VICKY | 50000 |
+-----+-----+-----+-----+-----+-----+-----+
2 rows in set (0.000 sec)

MariaDB [rental_alya]> _
```

Kesimpulan

Jika ingin menampilkan hasil dari menyeleksi table dengan cara memberikan sebuah rentang nilai yang beda nya sebelumnya itu jika nilai tersebut masih berada di dalam rentang nilai yang diberikan maka akan di tampilkan sedangkan kali ini di luar dari rentang nilai yang akan di tampilkan untuk itu kalian bisa menggunakan sebuah query dengan struktur `select * from nama_table where nama_kolom not between nilai1 and nilai2;`

<=

Penjelasan

1. `select` query yang digunakan untuk menampilkan hasil dari `insert`
2. `*` arti nya semua kolom akan ditampilkan
3. `from` query yang digunakan untuk memberikan penanda bahwa table mana yang akan di tampilkan
4. `mobil` nama table yang akan ditampilkan
5. `where` query yang digunakan untuk memberikan sebuah kondisi
6. `harga_rental<=100000` sebuah kondisi yang telah di berikan dan `harga_rental` itu nama kolom, `<=` merupakan operator, dan `100000` merupakan sebuah nilai.

Struktur Query

```
SELECT * FROM [nama_tabel] WHERE kolom1 <= nilai1;
```

Contoh Query

```
SELECT * FROM mobil WHERE harga_rental <= 100000;
```

Hasil

```
Administrator: XAMPP for Windows - mysql -u root -p
MariaDB [rental_alya]> SELECT * FROM mobil WHERE harga_rental <= 100000;
+-----+-----+-----+-----+-----+-----+-----+
| id_mobil | no_plat | no_mesin | warna | pemilik | peminjaman | harga_rental |
+-----+-----+-----+-----+-----+-----+-----+
| 1 | DD 2650 XY | ACX3560 | HITAM | REZKY | QILA | 50000 |
| 2 | DD 2440 AX | BCS1120 | MERAH | REZKY | ACCY | 100000 |
| 3 | B 1617 QC | LSQ1112 | SILVER | ALYA | VICKY | 50000 |
| 5 | DD 2210 LS | CJH1011 | HITAM | AYA | NULL | 100000 |
+-----+-----+-----+-----+-----+-----+-----+
4 rows in set (0.000 sec)

MariaDB [rental_alya]>
```

Kesimpulan

Jika ingin menampilkan table dengan menggunakan hasil seleksi yang dimana jika dia lebih kecil dari nilai yang ditentukan maka dia akan tampil, yaitu dengan cara menggunakan query dengan struktur `select * from nama_table where nama_kolom<=nilai;`

>=

Penjelasan

1. `select` query yang digunakan untuk menampilkan hasil dari `insert`
2. `*` arti nya semua kolom akan ditampilkan
3. `from` query yang digunakan untuk memberikan penanda bahwa table mana yang akan di tampilkan
4. `mobil` nama table yang akan ditampilkan
5. `where` query yang digunakan untuk memberikan sebuah kondisi
6. `harga_rental>=100000` sebuah kondisi yang telah di berikan dan `harga_rental` itu nama kolom, `>=` merupakan operator, dan `100000` merupakan sebuah nilai

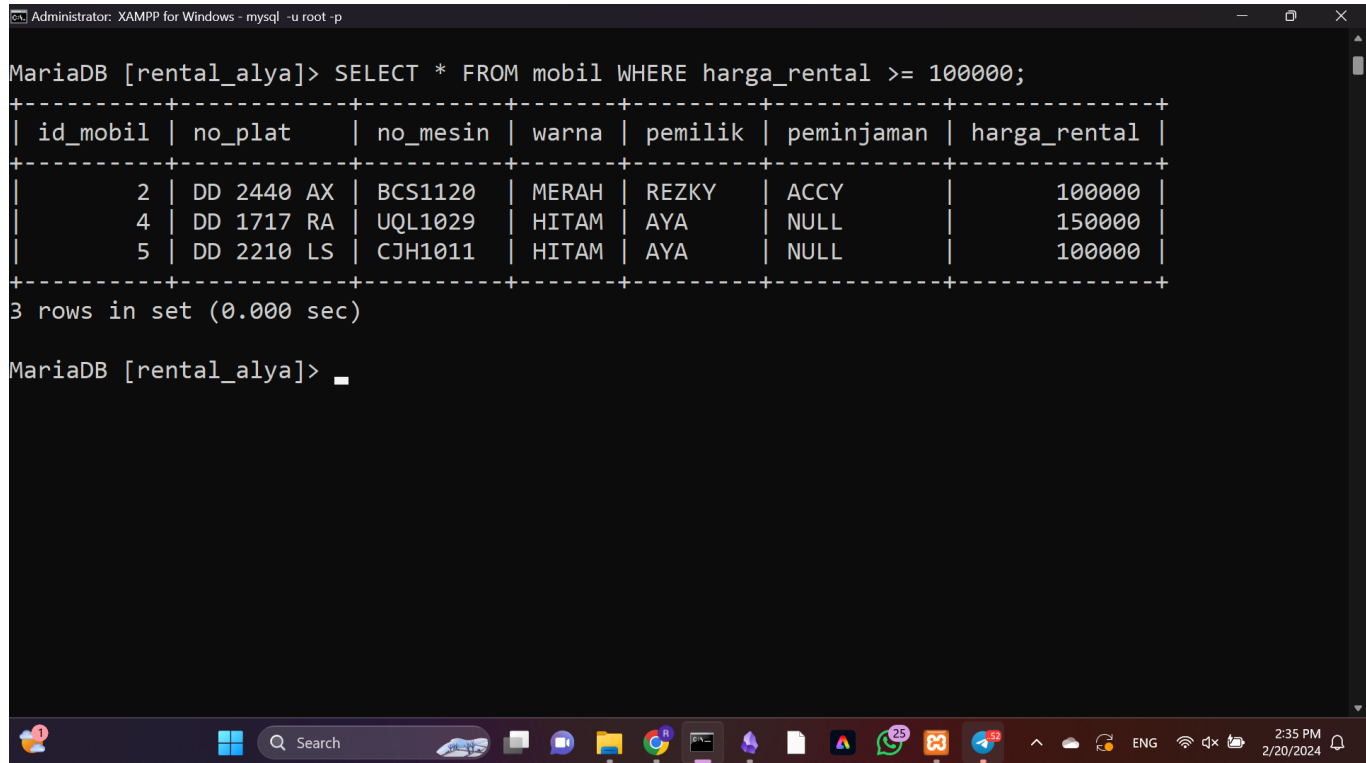
Struktur Query

```
SELECT * FROM [nama_tabel] WHERE kolom1 >= nilai1;
```

Contoh Query

```
SELECT * FROM mobil WHERE harga_rental >= 100000;
```

Hasil



The screenshot shows a MySQL command prompt window titled "Administrator: XAMPP for Windows - mysql -u root -p". The prompt is "MariaDB [rental_alya]>". The user has entered the query "SELECT * FROM mobil WHERE harga_rental >= 100000;". The results are displayed in a table format with columns: id_mobil, no_plat, no_mesin, warna, pemilik, peminjaman, and harga_rental. There are 3 rows in the set, and it took 0.000 seconds to execute. The prompt is now "MariaDB [rental_alya]> " followed by a cursor.

id_mobil	no_plat	no_mesin	warna	pemilik	peminjaman	harga_rental
2	DD 2440 AX	BCS1120	MERAH	REZKY	ACCY	100000
4	DD 1717 RA	UQL1029	HITAM	AYA	NULL	150000
5	DD 2210 LS	CJH1011	HITAM	AYA	NULL	100000

3 rows in set (0.000 sec)

MariaDB [rental_alya]>

Kesimpulan

jika ingin menampilkan table dengan menggunakan hasil seleksi yang dimana jika dia lebih besar dari nilai yang di tentukan maka dia akan tampil, yaitu dengan cara menggunakan query dengan struktur `select * from nama_table where nama_kolom<=nilai;`

<> atau !=

Penjelasan

PENJELASAN 1

1. `select` query yang digunakan untuk menampilkan hasil dari `insert`
2. `*` artinya semua kolom akan ditampilkan
3. `from` query yang digunkan untuk memberikan penanda bahwa table mana yang akan di tampilkan
4. `mobil` nama table yang akan ditampilkan

5. `where` query yang digunakan untuk memberikan sebuah kondisi
6. `harga_rental<>100000` sebuah kondisi yang telah di berikan dan `harga_rental` itu nama kolom, `<>` merupakan operator, dan `100000` merupakan sebuah nilai

PENJELASAN 2

1. `select` query yang digunakan untuk menampilkan hasil dari `insert`
2. `*` artinya semua kolom akan ditampilkan
3. `from` query yang digunkan untuk memberikan penanda bahwa table mana yang akan di tampilkan
4. `mobil` nama table yang akan ditampilkan
5. `where` query yang digunakan untuk memberikan sebuah kondisi
6. `harga_rental!=50000` sebuah kondisi yang telah di berikan dan `harga_rental` itu nama kolom, `!=` merupakan operator, dan `50000` merupakan sebuah nilai

Struktur Query

```
SELECT * FROM [nama_tabel] WHERE kolom1 <> nilai1;
```

```
SELECT * FROM [nama_tabel] WHERE kolom1 != nilai1;
```

Contoh Query

```
SELECT * FROM mobil WHERE harga_rental <> 100000;
```

```
SELECT * FROM mobil WHERE harga_rental != 50000;
```

Hasil

```
Administrator: XAMPP for Windows - mysql -u root -p

MariaDB [rental_alya]> SELECT * FROM mobil WHERE harga_rental <> 100000;
+-----+-----+-----+-----+-----+-----+-----+
| id_mobil | no_plat | no_mesin | warna | pemilik | peminjaman | harga_rental |
+-----+-----+-----+-----+-----+-----+-----+
| 1 | DD 2650 XY | ACX3560 | HITAM | REZKY | QILA | 50000 |
| 3 | B 1617 QC | LSQ1112 | SILVER | ALYA | VICKY | 50000 |
| 4 | DD 1717 RA | UQL1029 | HITAM | AYA | NULL | 150000 |
+-----+-----+-----+-----+-----+-----+-----+
3 rows in set (0.001 sec)

MariaDB [rental_alya]>
```

```
Administrator: XAMPP for Windows - mysql -u root -p

MariaDB [rental_alya]> SELECT * FROM mobil WHERE harga_rental != 50000;
+-----+-----+-----+-----+-----+-----+-----+
| id_mobil | no_plat | no_mesin | warna | pemilik | peminjaman | harga_rental |
+-----+-----+-----+-----+-----+-----+-----+
| 2 | DD 2440 AX | BCS1120 | MERAH | REZKY | ACCY | 100000 |
| 4 | DD 1717 RA | UQL1029 | HITAM | AYA | NULL | 150000 |
| 5 | DD 2210 LS | CJH1011 | HITAM | AYA | NULL | 100000 |
+-----+-----+-----+-----+-----+-----+-----+
3 rows in set (0.000 sec)

MariaDB [rental_alya]> _
```

Kesimpulan

dari kedua contoh operator kita bisa menyimpulkan bahwa operator `!=` dengan `<>` memiliki arti yang sama yang dimana jika ingin menampilkan table dengan menggunakan sebuah nilai maka nilai yang ingin di tampilkan tidak boleh sama dengan nilai yang telah ditentukan.

Tantangan login

Penjelasan

`SELECT nama FROM akun` : merupakan perintah SQL untuk mencari kolom nama yang berada di tabel akun.

`WHERE nama="githa"` : query SQL untuk mencari keberadaan data yang bernama "githa" yang berada di dalam kolom `nama`.

Query

```
select nama from akun
-> where nama = "githa";
```

Hasil

```
MariaDB [rental_alya]> select nama from akun
-> where nama = "githa";
+-----+
| nama  |
+-----+
| Githa |
+-----+
1 row in set (0.000 sec)
```

Kesimpulan

jika ingin menampilkan dari hasil seleksi yang dimana hanya ada satu nilai dari satu kolom atau hanya 1 kolom yang ingin di tampilkan, yaitu dengan cara menggunakan query dengan struktur

```
select nama_kolom1 from nama_table where nama_kolom2=nilai;
```

IN

Penjelasan

1. `SELECT *` artinya kita akan mengambil semua kolom dari tabel "mobil".
2. `FROM mobil` artinya kita akan mengambil data dari tabel "mobil".
3. `WHERE warna IN ('Silver', 'Merah')` artinya kita hanya akan mengambil baris-baris yang di mana nilai pada kolom "warna" memiliki nilai "Silver" dan "Merah".

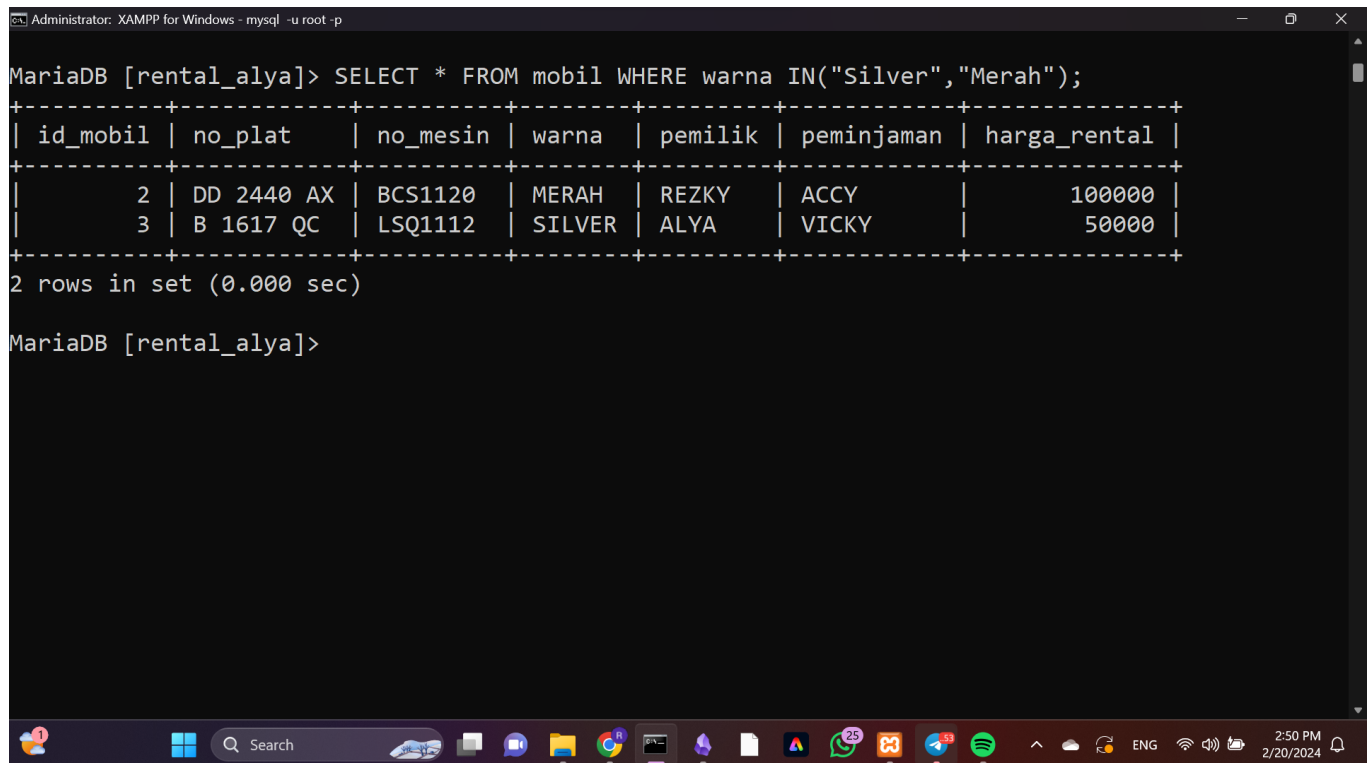
Struktur Query

```
SELECT * FROM [nama_tabel] WHERE [nama_kolom] IN("nilai1","nilai2");
```

Contoh Query

```
SELECT * FROM mobil WHERE warna IN("Silver","Merah");
```

Hasil



The screenshot shows a MySQL command prompt window titled "Administrator: XAMPP for Windows - mysql -u root -p". The user is in the "MariaDB [rental_alya]" database. The query executed is `SELECT * FROM mobil WHERE warna IN("Silver","Merah");`. The results are displayed in a table format with 7 columns: `id_mobil`, `no_plat`, `no_mesin`, `warna`, `pemilik`, `peminjaman`, and `harga_rental`. Two rows are returned, corresponding to the 'Merah' and 'Silver' cars. Below the table, it says "2 rows in set (0.000 sec)".

id_mobil	no_plat	no_mesin	warna	pemilik	peminjaman	harga_rental
2	DD 2440 AX	BCS1120	MERAH	REZKY	ACCY	100000
3	B 1617 QC	LSQ1112	SILVER	ALYA	VICKY	50000

Kesimpulan

Perintah `SELECT *` digunakan untuk mengambil semua kolom dari tabel "mobil". Selanjutnya, `FROM mobil` menunjukkan bahwa data diambil dari tabel "mobil". Klausa `WHERE warna IN ('Silver', 'Merah')` digunakan untuk memfilter baris-baris di mana nilai kolom "warna" adalah "Silver" atau "Merah". Jadi, query ini akan mengembalikan baris-baris di mana nilai kolom "warna" berisi nilai "Silver" atau "Merah".

IN + AND

Penjelasan

1. `SELECT *` artinya kita akan mengambil semua kolom dari tabel "mobil".

2. `FROM mobil` artinya kita akan mengambil data dari tabel "mobil".
3. `WHERE warna IN ('Silver', 'Merah')` artinya kita hanya akan mengambil baris-baris di mana nilai kolom "warna" adalah "Silver" atau "Merah".
4. `AND harga_rental = 50000` artinya kita hanya akan mengambil baris-baris di mana nilai kolom "harga_rental" adalah 50000.

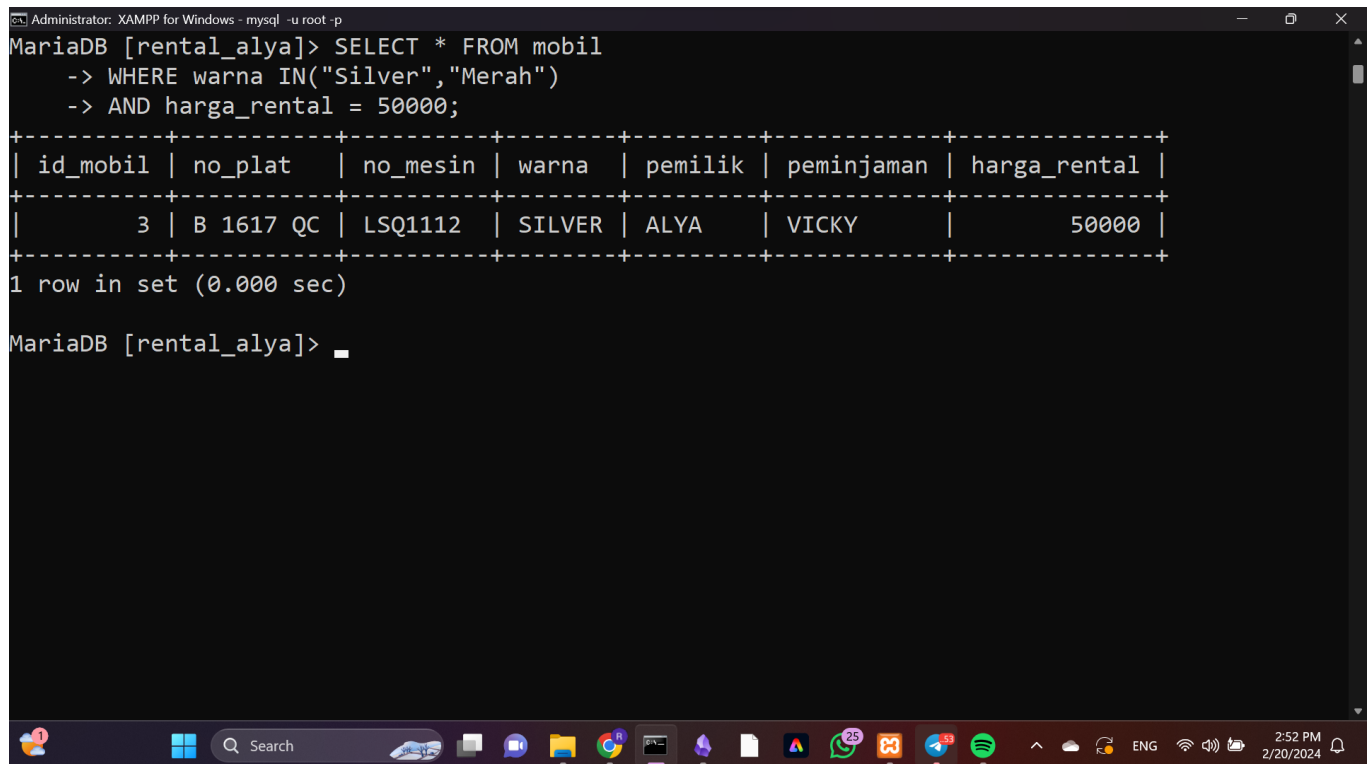
Struktur Query

```
SELECT * FROM [nama_tabel]
-> WHERE [nama_kolom1] IN("nilai1","nilai2")
-> AND [nama_kolom2] = nilai;
```

Contoh Query

```
SELECT * FROM mobil
-> WHERE warna IN("Silver","Merah")
-> AND harga_rental = 50000;
```

Hasil



The screenshot shows a MySQL command prompt window titled "Administrator: XAMPP for Windows - mysql -u root -p". The user is logged in as root. The prompt is "MariaDB [rental_alya]>". The user enters the query: `SELECT * FROM mobil`, followed by `-> WHERE warna IN("Silver","Merah")`, and finally `-> AND harga_rental = 50000;`. The result is displayed in a table format with columns: `id_mobil`, `no_plat`, `no_mesin`, `warna`, `pemilik`, `peminjaman`, and `harga_rental`. The result shows one row: `3`, `B 1617 QC`, `LSQ1112`, `SILVER`, `ALYA`, `VICKY`, and `50000`. The prompt then returns to `MariaDB [rental_alya]>`.

```
Administrator: XAMPP for Windows - mysql -u root -p
MariaDB [rental_alya]> SELECT * FROM mobil
-> WHERE warna IN("Silver","Merah")
-> AND harga_rental = 50000;
+-----+-----+-----+-----+-----+-----+-----+
| id_mobil | no_plat | no_mesin | warna | pemilik | peminjaman | harga_rental |
+-----+-----+-----+-----+-----+-----+-----+
| 3 | B 1617 QC | LSQ1112 | SILVER | ALYA | VICKY | 50000 |
+-----+-----+-----+-----+-----+-----+-----+
1 row in set (0.000 sec)

MariaDB [rental_alya]>
```

Kesimpulan

Perintah `SELECT *` digunakan untuk mengambil semua kolom dari tabel "mobil". Selanjutnya, `FROM mobil` menunjukkan bahwa data diambil dari tabel "mobil". Klausa `WHERE warna IN ('Silver', 'Merah')` digunakan untuk memfilter baris-baris di mana nilai kolom "warna" adalah "Hitam" atau "Merah". Selain itu, `AND harga_rental = 50000` digunakan untuk memfilter baris-baris di mana nilai kolom "harga_rental" adalah 50000. Jadi, query ini akan mengembalikan baris-baris di mana nilai kolom "warna" adalah "Silver" atau "Merah", dan nilai kolom "harga_rental" adalah 50000.

IN + OR

Penjelasan

- `SELECT *` : Memilih semua kolom dari tabel.
- `FROM mobil` : Menunjukkan bahwa data diambil dari tabel `mobil`.
- `WHERE warna IN ('Hitam', 'Silver') OR harga_rental = 150000` : Menggunakan klausa `WHERE` untuk memfilter baris berdasarkan kondisi bahwa nilai kolom `warna` adalah 'Hitam' atau 'Silver', atau nilai kolom `harga_rental` adalah 150000.

Struktur Query

```
SELECT * FROM [nama_tabel]
-> WHERE [nama_kolom1] IN ('nilai1','nilai2')
-> OR [nama_kolom2] = nilai1;
```

Contoh Query

```
SELECT * FROM mobil
-> WHERE warna IN ('Hitam','Silver')
-> OR harga_rental = 150000;
```

Hasil

```
Administrator: XAMPP for Windows - mysql -u root -p
MariaDB [rental_alya]> SELECT * FROM mobil
  -> WHERE warna IN ('Hitam','Silver')
  -> OR harga_rental = 150000;
+-----+-----+-----+-----+-----+-----+-----+
| id_mobil | no_plat | no_mesin | warna | pemilik | peminjaman | harga_rental |
+-----+-----+-----+-----+-----+-----+-----+
| 1 | DD 2650 XY | ACX3560 | HITAM | REZKY | QILA | 50000 |
| 3 | B 1617 QC | LSQ1112 | SILVER | ALYA | VICKY | 50000 |
| 4 | DD 1717 RA | UQL1029 | HITAM | AYA | NULL | 150000 |
| 5 | DD 2210 LS | CJH1011 | HITAM | AYA | NULL | 100000 |
+-----+-----+-----+-----+-----+-----+-----+
4 rows in set (0.000 sec)

MariaDB [rental_alya]> _
```

Kesimpulan

Perintah `SELECT *` digunakan untuk memilih semua kolom dari tabel. Selanjutnya, `FROM mobil` menunjukkan bahwa data diambil dari tabel `mobil`. Klausa `WHERE warna IN ('Hitam', 'Silver') OR harga_rental = 150000` digunakan untuk memfilter baris berdasarkan kondisi bahwa nilai kolom `warna` adalah 'Hitam' atau 'Silver', atau nilai kolom `harga_rental` adalah 150000. Dengan demikian, query ini akan mengembalikan baris-baris di mana nilai kolom `warna` adalah 'Hitam' atau 'Silver', atau nilai kolom `harga_rental` adalah 150000.

IN + AND + OPERATOR

Penjelasan

PENJELASAN 1

1. `SELECT *` artinya kita akan mengambil semua kolom dari tabel "mobil".
2. `FROM mobil` artinya kita akan mengambil data dari tabel "mobil".
3. `WHERE warna IN ('Hitam', 'Silver')` artinya kita hanya akan mengambil baris-baris di mana nilai kolom "warna" adalah "Hitam" atau "Silver".
4. `OR harga_rental > 50000` artinya kita juga akan mengambil baris-baris di mana nilai kolom "harga_rental" lebih besar dari 50000.

PENJELASAN 2

1. `SELECT *` artinya kita akan mengambil semua kolom dari tabel "mobil".

2. `FROM mobil` artinya kita akan mengambil data dari tabel "mobil".
3. `WHERE warna IN ('Hitam', 'Silver')` artinya kita hanya akan mengambil baris-baris di mana nilai kolom "warna" adalah "Hitam" atau "Silver".
4. `OR harga_rental < 150000` artinya kita juga akan mengambil baris-baris di mana nilai kolom "harga_rental" kurang dari 150000.

Struktur Query

```
SELECT * FROM [nama_tabel]
-> WHERE [nama_kolom1] IN ('nilai1','nilai2')
-> AND [nama_kolom2] > nilai1;

SELECT * FROM [nama_tabel]
-> WHERE [nama_kolom1] IN ('nilai1','nilai2')
-> AND [nama_kolom2] < nilai1![[tntngan lbih kcil.png]];
```

Contoh Query

```
SELECT * FROM mobil
-> WHERE warna IN ('Hitam','Silver')
-> AND harga_rental > 50000;

SELECT * FROM mobil
-> WHERE warna IN ('Hitam','Silver')
-> AND harga_rental < 150000;
```

Hasil


```
Administrator: XAMPP for Windows - mysql -u root -p

MariaDB [rental_alya]> SELECT * FROM mobil
-> WHERE warna IN ('Hitam','Silver')
-> AND harga_rental > 50000;

+-----+-----+-----+-----+-----+-----+-----+
| id_mobil | no_plat | no_mesin | warna | pemilik | peminjaman | harga_rental |
+-----+-----+-----+-----+-----+-----+-----+
| 4 | DD 1717 RA | UQL1029 | HITAM | AYA | NULL | 150000 |
| 5 | DD 2210 LS | CJH1011 | HITAM | AYA | NULL | 100000 |
+-----+-----+-----+-----+-----+-----+-----+

2 rows in set (0.000 sec)

MariaDB [rental_alya]>
```

```
Administrator: XAMPP for Windows - mysql -u root -p

MariaDB [rental_alya]> SELECT * FROM mobil
-> WHERE warna IN ('Hitam','Silver')
-> AND harga_rental < 150000;

+-----+-----+-----+-----+-----+-----+-----+
| id_mobil | no_plat | no_mesin | warna | pemilik | peminjaman | harga_rental |
+-----+-----+-----+-----+-----+-----+-----+
| 1 | DD 2650 XY | ACX3560 | HITAM | REZKY | QILA | 50000 |
| 3 | B 1617 QC | LSQ1112 | SILVER | ALYA | VICKY | 50000 |
| 5 | DD 2210 LS | CJH1011 | HITAM | AYA | NULL | 100000 |
+-----+-----+-----+-----+-----+-----+-----+

3 rows in set (0.000 sec)

MariaDB [rental_alya]>
```

Kesimpulan

KESIMPULAN 1

Perintah `SELECT *` digunakan untuk mengambil semua kolom dari tabel "mobil". Selanjutnya, `FROM mobil` menunjukkan bahwa data diambil dari tabel "mobil". Klausa `WHERE warna IN ('Hitam', 'Silver')` digunakan untuk memfilter baris-baris di mana nilai kolom "warna" adalah "Hitam" atau "Silver". Selain itu, `OR harga_rental > 50000` digunakan untuk juga

memfilter baris-baris di mana nilai kolom "harga_rental" lebih besar dari 50000. Jadi, pernyataan ini akan mengambil baris-baris di mana nilai kolom "warna" adalah "Hitam" atau "Silver", atau di mana nilai kolom "harga_rental" lebih besar dari 50000.

KESIMPULAN 2

Perintah `SELECT *` digunakan untuk mengambil semua kolom dari tabel "mobil". Selanjutnya, `FROM mobil` menunjukkan bahwa data diambil dari tabel "mobil". Klausa `WHERE warna IN ('Hitam', 'Silver')` digunakan untuk memfilter baris-baris di mana nilai kolom "warna" adalah "Hitam" atau "Silver". Selain itu, `OR harga_rental < 150000` digunakan untuk juga memfilter baris-baris di mana nilai kolom "harga_rental" kurang dari 150000. Jadi, pernyataan ini akan mengambil baris-baris di mana nilai kolom "warna" adalah "Hitam" atau "Silver", atau di mana nilai kolom "harga_rental" kurang dari 150000.

LIKE

Mencari Awalan

Penjelasan

1. `SELECT *` artinya kita akan mengambil semua kolom dari tabel "mobil".
2. `FROM mobil` artinya kita akan mengambil data dari tabel "mobil".
3. `WHERE pemilik LIKE 'ay%'` artinya kita hanya akan mengambil baris-baris di mana nilai kolom "pemilik" dimulai dengan kata "ay" (dilanjutkan dengan karakter apa pun, karena simbol `%` dalam pola pencocokan).

Struktur Query

```
select * from [nama_tabel]
-> where [nama_kolom] like 'nama_awal%';
```

Contoh Query

```
select * from mobil
-> where pemilik like 'ay%';
```

Hasil

```
MariaDB [rental_alya]> select * from mobil  
-> where pemilik like 'ay%';
```

```
+-----+-----+-----+-----+-----+-----+-----+  
| id_mobil | no_plat   | no_mesin | warna | pemilik | peminjaman | harga_rental |  
+-----+-----+-----+-----+-----+-----+-----+  
|         4 | DD 1717 RA | UQL1029  | HITAM | AYA      | NULL        | 150000        |  
|         5 | DD 2210 LS | CJH1011  | HITAM | AYA      | NULL        | 100000        |  
+-----+-----+-----+-----+-----+-----+-----+  
2 rows in set (0.002 sec)
```

Kesimpulan

Perintah `SELECT *` digunakan untuk mengambil semua kolom dari tabel "mobil". Selanjutnya, `FROM mobil` menunjukkan bahwa data diambil dari tabel "mobil". Klausa `WHERE pemilik LIKE 'ay%'` digunakan untuk memfilter baris-baris di mana nilai kolom "pemilik" dimulai dengan kata "ay" (dilanjutkan dengan karakter apa pun, karena simbol `%` dalam pola pencocokan). Jadi, pernyataan ini akan mengambil baris-baris di mana nilai kolom "pemilik" memenuhi pola tersebut.

Mencari Akhiran

Penjelasan

1. `SELECT *` artinya kita akan mengambil semua kolom dari tabel "mobil".
2. `FROM mobil` artinya kita akan mengambil data dari tabel "mobil".
3. `WHERE pemilik LIKE '%y'` artinya kita hanya akan mengambil baris-baris di mana nilai kolom "pemilik" diakhiri dengan huruf "y" (dimulai dengan karakter apa pun, karena simbol `%` sebelum "m" dalam pola pencocokan).

Struktur Query

```
select * from [nama_tabel]  
-> where [nama_kolom] like '%nama_akhir';
```

Contoh Query

```
select * from mobil  
-> where pemilik like '%y';
```

Hasil

```
MariaDB [rental_alya]> select * from mobil
```

```
-> where pemilik like '%y';
```

```
+-----+-----+-----+-----+-----+-----+-----+
| id_mobil | no_plat   | no_mesin | warna | pemilik | peminjaman | harga_rental |
+-----+-----+-----+-----+-----+-----+-----+
|          1 | DD 2650 XY | ACX3560  | HITAM | REZKY   | QILA       | 50000        |
|          2 | DD 2440 AX | BCS1120  | MERAH | REZKY   | ACCY       | 100000       |
+-----+-----+-----+-----+-----+-----+-----+
2 rows in set (0.001 sec)
```

Kesimpulan

Perintah `SELECT *` digunakan untuk mengambil semua kolom dari tabel "mobil". Selanjutnya, `FROM mobil` menunjukkan bahwa data diambil dari tabel "mobil". Klausa `WHERE pemilik LIKE '%y'` digunakan untuk memfilter baris-baris di mana nilai kolom "pemilik" diakhiri dengan huruf "y" (dimulai dengan karakter apa pun, karena simbol `%` sebelum "y" dalam pola pencocokan). Jadi, pernyataan ini akan mengambil baris-baris di mana nilai kolom "pemilik" memenuhi pola tersebut.

Mencari Awalan & Akhiran

Penjelasan

1. `SELECT *` artinya kita akan mengambil semua kolom dari tabel "mobil".
2. `FROM mobil` artinya kita akan mengambil data dari tabel "mobil".
3. `WHERE pemilik LIKE 'a%a'` artinya kita hanya akan mengambil baris-baris di mana nilai kolom "pemilik" dimulai dengan huruf "a", diikuti oleh setidaknya satu karakter apa pun (dilambangkan oleh simbol `%`), dan diakhiri dengan huruf "a".

Struktur Query

```
select * from [nama_tabel]
-> where [nama_kolom] like 'nama_awal%nama_akhir';
```

Contoh Query

```
select * from mobil
-> where pemilik like 'a%a';
```

Hasil

```
MariaDB [rental_alya]> select * from mobil  
-> where pemilik like 'a%a';
```

id_mobil	no_plat	no_mesin	warna	pemilik	peminjaman	harga_rental
3	B 1617 QC	LSQ1112	SILVER	ALYA	VICKY	50000
4	DD 1717 RA	UQL1029	HITAM	AYA	NULL	150000
5	DD 2210 LS	CJH1011	HITAM	AYA	NULL	100000

3 rows in set (0.001 sec)

Kesimpulan

Perintah `SELECT *` digunakan untuk mengambil semua kolom dari tabel "mobil". Selanjutnya, `FROM mobil` menunjukkan bahwa data diambil dari tabel "mobil". Klausa `WHERE pemilik LIKE 'a%a'` digunakan untuk memfilter baris-baris di mana nilai kolom "pemilik" dimulai dengan huruf "a", diikuti oleh setidaknya satu karakter apa pun (dilambangkan oleh simbol `%`), dan diakhiri dengan huruf "a". Jadi, pernyataan ini akan mengambil baris-baris di mana nilai kolom "pemilik" memenuhi pola tersebut.

Berdasarkan Total Karakter

Penjelasan

PENJELASAN 1

1. `SELECT *` artinya kita akan mengambil semua kolom dari tabel "mobil".
2. `FROM mobil` artinya kita akan mengambil data dari tabel "mobil".
3. `WHERE pemilik LIKE 'A__'` artinya kita hanya akan mengambil baris-baris di mana nilai kolom "pemilik" terdiri dari 3 karakter, di mana karakter pertama adalah "A" (dilambangkan oleh underscore `_`) dan dua karakter berikutnya adalah karakter apa pun.

PENJELASAN 2

1. `SELECT *` artinya kita akan mengambil semua kolom dari tabel "mobil".
2. `FROM mobil` artinya kita akan mengambil data dari tabel "mobil".
3. `WHERE pemilik LIKE '___'` artinya kita hanya akan menampilkan sebuah tabel dengan syarat nilai pemilik harus memiliki 3 huruf

Struktur Query

```
select * from [nama_tabel]  
-> Where [nama_kolom] LIKE 'awalan+_sesuaijumlahkarakter';
```

```
select * from [nama_tabel]
-> Where [nama_kolom] LIKE '_sesuaijumlahkarakter';
```

Contoh Query

```
select * from mobil
-> Where pemilik LIKE 'A___';

select * from mobil
-> Where pemilik LIKE '___';
```

Hasil

```
MariaDB [rental_alya]> select * from mobil
-> Where pemilik LIKE 'A___';
```

id_mobil	no_plat	no_mesin	warna	pemilik	peminjaman	harga_rental
3	B 1617 QC	LSQ1112	SILVER	ALYA	VICKY	50000

1 row in set (0.001 sec)

```
MariaDB [rental_alya]> select * from mobil
-> Where pemilik LIKE '___';
```

id_mobil	no_plat	no_mesin	warna	pemilik	peminjaman	harga_rental
4	DD 1717 RA	UQL1029	HITAM	AYA	NULL	150000
5	DD 2210 LS	CJH1011	HITAM	AYA	NULL	100000

2 rows in set (0.001 sec)

Kesimpulan

KESIMPULAN 1

Perintah `SELECT *` digunakan untuk mengambil semua kolom dari tabel "mobil". Selanjutnya, `FROM mobil` menunjukkan bahwa data diambil dari tabel "mobil". Klausa `WHERE pemilik LIKE 'A___'` digunakan untuk memfilter baris-baris di mana nilai kolom "pemilik" terdiri dari 3 karakter, di mana karakter pertama adalah "A" (dilambangkan oleh underscore `_`) dan dua karakter berikutnya adalah karakter apa pun. Jadi, pernyataan ini akan mengambil baris-baris di mana nilai kolom "pemilik" memenuhi pola tersebut.

KESIMPULAN 2

Perintah `SELECT *` digunakan untuk mengambil semua kolom dari tabel "mobil". Selanjutnya,

`FROM mobil` menunjukkan bahwa data diambil dari tabel "mobil". Klausa `WHERE pemilik LIKE '___'` digunakan untuk menampilkan tabel dengan syarat nilai kolom "pemilik" harus memiliki tepat 3 huruf. Jadi, pernyataan ini akan menampilkan baris-baris di mana nilai kolom "pemilik" memiliki tepat 3 huruf.

Kombinasi

Penjelasan

PENJELASAN 1

1. `SELECT *` artinya kita akan mengambil semua kolom dari tabel "mobil".
2. `FROM mobil` artinya kita akan mengambil data dari tabel "mobil".
3. `WHERE pemilik LIKE '__y%'` artinya kita hanya akan mengambil baris-baris di mana nilai kolom "pemilik" terdiri dari setidaknya 3 karakter, di mana dua karakter pertama adalah karakter apa pun (dilambangkan oleh dua underscore `_`), karakter ketiga adalah "y", dan karakter-karakter berikutnya adalah karakter apa pun (dilambangkan oleh simbol `%`).

PENJELASAN 2

1. `SELECT *` artinya kita akan mengambil semua kolom dari tabel "mobil".
2. `FROM mobil` artinya kita akan mengambil data dari tabel "mobil".
3. `WHERE pemilik LIKE '__a%'` artinya kita hanya akan mengambil baris-baris di mana nilai kolom "pemilik" terdiri dari setidaknya 3 karakter, di mana karakter pertama adalah karakter apa pun (dilambangkan oleh satu underscore `_`), karakter ketiga adalah "a", dan karakter-karakter berikutnya adalah karakter apa pun (dilambangkan oleh simbol `%`).

Struktur Query

```
select * from mobil
    -> Where pemilik LIKE 'A%___';

select * from mobil
    -> Where pemilik LIKE '__a%';
```

Contoh Query

```
select * from mobil
    -> Where pemilik LIKE '__y%';
```

```
select * from mobil
-> Where pemilik LIKE '__a%';
```

Hasil

```
MariaDB [rental_alya]> select * from mobil
-> Where pemilik LIKE 'A%__';
```

id_mobil	no_plat	no_mesin	warna	pemilik	peminjaman	harga_rental
3	B 1617 QC	LSQ1112	SILVER	ALYA	VICKY	50000

1 row in set (0.001 sec)

```
MariaDB [rental_alya]> select * from mobil
-> Where pemilik LIKE '__a%';
```

id_mobil	no_plat	no_mesin	warna	pemilik	peminjaman	harga_rental
4	DD 1717 RA	UQL1029	HITAM	AYA	NULL	150000
5	DD 2210 LS	CJH1011	HITAM	AYA	NULL	100000

2 rows in set (0.001 sec)

```
MariaDB [rental_alya]> select * from mobil
-> Where pemilik LIKE '__y%';
```

id_mobil	no_plat	no_mesin	warna	pemilik	peminjaman	harga_rental
3	B 1617 QC	LSQ1112	SILVER	ALYA	VICKY	50000

1 row in set (0.001 sec)

Kesimpulan

KESIMPULAN 1

Perintah `SELECT *` digunakan untuk mengambil semua kolom dari tabel "mobil". Selanjutnya, `FROM mobil` menunjukkan bahwa data diambil dari tabel "mobil". Klausa `WHERE pemilik LIKE '__y%'` digunakan untuk memfilter baris-baris di mana nilai kolom "pemilik" terdiri dari setidaknya 3 karakter, di mana dua karakter pertama adalah karakter apa pun (dilambangkan oleh dua underscore `_`), karakter ketiga adalah "y", dan karakter-karakter berikutnya adalah karakter apa pun (dilambangkan oleh simbol `%`). Jadi, pernyataan ini akan mengambil baris-baris di mana nilai kolom "pemilik" memenuhi pola tersebut.

KESIMPULAN 2

Perintah `SELECT *` digunakan untuk mengambil semua kolom dari tabel "mobil". Selanjutnya, `FROM mobil` menunjukkan bahwa data diambil dari tabel "mobil". Klausa `WHERE pemilik LIKE '__a%'` digunakan untuk memfilter baris-baris di mana nilai kolom "pemilik" terdiri dari setidaknya 2 karakter, di mana karakter pertama adalah karakter apa pun (dilambangkan oleh satu underscore `_`), karakter ketiga adalah "a", dan karakter-karakter berikutnya adalah karakter apa pun (dilambangkan oleh simbol `%`). Jadi, pernyataan ini akan mengambil baris-baris di mana nilai kolom "pemilik" memenuhi pola tersebut.

NOT LIKE

Penjelasan

1. `SELECT *` : Memilih semua kolom dari tabel 'mobil'.
2. `FROM mobil` : Menentukan tabel yang digunakan untuk mengambil data, dalam hal ini tabel 'mobil'.
3. `WHERE peminjam NOT LIKE 'A%'` : Menggunakan klausa `WHERE` untuk memfilter baris-baris yang akan diambil. Kondisi `peminjam NOT LIKE 'A%'` digunakan untuk memeriksa apakah nilai kolom 'peminjam' tidak dimulai dengan huruf 'A'. Operator `NOT LIKE` digunakan untuk memeriksa apakah nilai tidak cocok dengan pola yang diberikan, dalam hal ini, pola 'A%' berarti dimulai dengan 'A'. Jadi, pernyataan ini akan mengembalikan baris-baris di mana nilai kolom 'peminjam' tidak dimulai dengan huruf 'A'.

Struktur Query

```
select * from [nama_tabel]
-> where [nama_kolom] NOT LIKE 'nama_awal%/%nama_akhir';
```

Contoh Query

```
select * from mobil
-> where pemilik NOT LIKE 'A%';
```

Hasil

```
MariaDB [rental_alya]> select * from mobil  
-> where pemilik NOT LIKE 'A%';
```

id_mobil	no_plat	no_mesin	warna	pemilik	peminjaman	harga_rental
1	DD 2650 XY	ACX3560	HITAM	REZKY	QILA	50000
2	DD 2440 AX	BCS1120	MERAH	REZKY	ACCY	100000

2 rows in set (0.001 sec)

Kesimpulan

Perintah `SELECT *` digunakan untuk memilih semua kolom dari tabel 'mobil'. Selanjutnya, `FROM mobil` menunjukkan tabel yang digunakan untuk mengambil data, yaitu tabel 'mobil'. Klausula `WHERE peminjam NOT LIKE 'A%'` digunakan untuk memfilter baris-baris yang akan diambil. Kondisi `peminjam NOT LIKE 'A%'` memeriksa apakah nilai kolom 'peminjam' tidak dimulai dengan huruf 'A'. Operator `NOT LIKE` digunakan untuk memeriksa apakah nilai tidak cocok dengan pola yang diberikan; dalam hal ini, pola 'A%' berarti dimulai dengan 'A'. Jadi, pernyataan ini akan mengembalikan baris-baris di mana nilai kolom 'peminjam' tidak dimulai dengan huruf 'A'.

NULL & NOT NULL

Mencari data kosong

Penjelasan

1. `SELECT *`: Memilih semua kolom dari tabel 'mobil'.
2. `FROM mobil`: Menentukan tabel yang digunakan untuk mengambil data, yaitu tabel 'mobil'.
3. `WHERE peminjam IS NULL`: Menggunakan klausa `WHERE` untuk memfilter baris-baris yang akan diambil. Kondisi `peminjam IS NULL` digunakan untuk memeriksa apakah nilai kolom 'peminjam' adalah `NULL`. Operator `IS NULL` digunakan untuk memeriksa apakah nilai kolom adalah `NULL`.

Struktur Query

```
select * from mobil  
-> where peminjaman is null;
```

Contoh Query

```
select * from mobil
-> where peminjaman is null;
```

Hasil

```
MariaDB [rental_alya]> select * from mobil
```

```
-> where peminjaman is null;
```

id_mobil	no_plat	no_mesin	warna	pemilik	peminjaman	harga_rental
4	DD 1717 RA	UQL1029	HITAM	AYA	NULL	150000
5	DD 2210 LS	CJH1011	HITAM	AYA	NULL	100000

2 rows in set (0.001 sec)

Kesimpulan

Perintah `SELECT *` digunakan untuk memilih semua kolom dari tabel 'mobil'. Kemudian, `FROM mobil` menunjukkan tabel yang digunakan untuk mengambil data, yaitu tabel 'mobil'.

Selanjutnya, `WHERE peminjam IS NULL` menggunakan klausa `WHERE` untuk memfilter baris-baris yang akan diambil. Kondisi `peminjam IS NULL` digunakan untuk memeriksa apakah nilai kolom 'peminjam' adalah `NULL`. Operator `IS NULL` digunakan untuk memeriksa apakah nilai kolom adalah `NULL`. Dengan demikian, query ini akan mengembalikan baris-baris di mana kolom 'peminjam' memiliki nilai `NULL`.

Mencari data yang tidak kosong

Penjelasan

1. `SELECT *`: Memilih semua kolom dari tabel.
2. `FROM mobil`: Menunjukkan bahwa data diambil dari tabel `mobil`.
3. `WHERE peminjam IS NOT NULL`: Menggunakan klausa `WHERE` untuk memfilter baris berdasarkan kondisi bahwa nilai kolom `peminjam` tidak `NULL`.

Struktur Query

```
select * from mobil
-> where peminjaman is not null;
```

Contoh Query

```
select * from mobil
-> where peminjaman is not null;
```

Hasil

```
MariaDB [rental_alya]> select * from mobil
-> where peminjaman IS NOT NULL;
```

id_mobil	no_plat	no_mesin	warna	pemilik	peminjaman	harga_rental
1	DD 2650 XY	ACX3560	HITAM	REZKY	QILA	50000
2	DD 2440 AX	BCS1120	MERAH	REZKY	ACCY	100000
3	B 1617 QC	LSQ1112	SILVER	ALYA	VICKY	50000

```
3 rows in set (0.001 sec)
```

Kesimpulan

Perintah SQL `SELECT * FROM mobil WHERE peminjaman IS NOT NULL;` digunakan untuk mengambil semua baris dari tabel "mobil" di mana kolom "peminjaman" memiliki nilai yang tidak `NULL`. Dengan kata lain, perintah ini mengambil data mobil yang sedang dipinjam atau sudah dipinjam dari tabel tersebut.

ORDER BY & LIMIT

Mengurutkan data dari data terkecil

Penjelasan

1. `SELECT *`: Memilih semua kolom dari tabel 'mobil'.
2. `FROM mobil`: Menunjukkan tabel yang digunakan untuk mengambil data, yaitu tabel 'mobil'.
3. `ORDER BY pemilik ASC`: Menggunakan klausa `ORDER BY` untuk mengurutkan hasil query berdasarkan kolom 'pemilik' secara ascending (ASC). Ini berarti data akan diurutkan dari nilai paling rendah ke nilai paling tinggi berdasarkan abjad.

Struktur Query

```
select * from [nama_tabel]
-> ORDER BY [nama_kolom] ASC;
```

Contoh Query

```
select * from mobil
-> ORDER BY pemilik ASC;
```

Hasil

```
MariaDB [rental_alya]> select * from mobil
-> ORDER BY pemilik ASC;
```

id_mobil	no_plat	no_mesin	warna	pemilik	peminjaman	harga_rental
3	B 1617 QC	LSQ1112	SILVER	ALYA	VICKY	50000
4	DD 1717 RA	UQL1029	HITAM	AYA	NULL	150000
5	DD 2210 LS	CJH1011	HITAM	AYA	NULL	100000
1	DD 2650 XY	ACX3560	HITAM	REZKY	QILA	50000
2	DD 2440 AX	BCS1120	MERAH	REZKY	ACCY	100000

5 rows in set (0.001 sec)

Kesimpulan

Perintah `SELECT *` digunakan untuk memilih semua kolom dari tabel 'mobil'. Kemudian, `FROM mobil` menunjukkan tabel yang digunakan untuk mengambil data, yaitu tabel 'mobil'.

Selanjutnya, `ORDER BY pemilik ASC` menggunakan klausa `ORDER BY` untuk mengurutkan hasil query berdasarkan kolom 'pemilik' secara ascending (ASC), yang berarti data akan diurutkan dari nilai paling rendah ke nilai paling tinggi berdasarkan abjad.

Mengurutkan data dari yang terbesar

Penjelasan

1. `SELECT *`: Memilih semua kolom dari tabel 'mobil'.
2. `FROM mobil`: Menunjukkan tabel yang digunakan untuk mengambil data, yaitu tabel 'mobil'.
3. `ORDER BY pemilik DESC`: Menggunakan klausa `ORDER BY` untuk mengurutkan hasil query berdasarkan kolom 'pemilik' secara descending (DESC). Ini berarti data akan diurutkan dari nilai paling tinggi ke nilai paling rendah berdasarkan abjad.

Struktur Query

```
select * from [nama_tabel]
-> ORDER BY [nama_kolom] DESC;
```

Contoh Query

```
select * from mobil
-> ORDER BY pemilik DESC;
```

Hasil

```
MariaDB [rental_alya]> select * from mobil
-> ORDER BY pemilik DESC;
```

id_mobil	no_plat	no_mesin	warna	pemilik	peminjaman	harga_rental
1	DD 2650 XY	ACX3560	HITAM	REZKY	QILA	50000
2	DD 2440 AX	BCS1120	MERAH	REZKY	ACCY	100000
4	DD 1717 RA	UQL1029	HITAM	AYA	NULL	150000
5	DD 2210 LS	CJH1011	HITAM	AYA	NULL	100000
3	B 1617 QC	LSQ1112	SILVER	ALYA	VICKY	50000

5 rows in set (0.001 sec)

Kesimpulan

Perintah `SELECT *` digunakan untuk memilih semua kolom dari tabel 'mobil'. Kemudian, `FROM mobil` menunjukkan tabel yang digunakan untuk mengambil data, yaitu tabel 'mobil'.

Selanjutnya, `ORDER BY peminjam DESC` menggunakan klausa `ORDER BY` untuk mengurutkan hasil query berdasarkan kolom 'peminjam' secara descending (DESC), yang berarti data akan diurutkan dari nilai paling tinggi ke nilai paling rendah berdasarkan abjad.

Membatasi data yang tampil

Penjelasan

Query `SELECT * FROM daftar_mobil WHERE warna = "hitam" ORDER BY harga_rental ASC LIMIT 2;` menampilkan semua kolom dari tabel "daftar_mobil" di mana kolom "warna" bernilai "hitam", lalu akan diurutkan berdasarkan kolom "harga_rental" secara ascending (dari yang terendah ke tertinggi), dan akan dibatasi hanya menampilkan 2 baris data.

Struktur Query

```
SELECT * FROM (nama_tabel) WHERE (nama_kolom = "nilai" ORDER BY (nama_kolom)
ASC LIMIT 2;
```

Contoh Query

```
SELECT * FROM daftar_mobil WHERE warna = "hitam" ORDER BY harga_rental ASC  
LIMIT 2;
```

Hasil

```
MariaDB [rental_alya]> SELECT * FROM daftar_mobil WHERE warna = "hitam" ORDER BY harga_rental  
ASC LIMIT 2;  
+-----+-----+-----+-----+-----+-----+-----+  
| id_mobil | no_plat | no_mesin | warna | pemilik | peminjaman | harga_rental |  
+-----+-----+-----+-----+-----+-----+-----+  
| 1 | DD 2650 XY | ACX3560 | HITAM | REZKY | QILA | 50000 |  
| 5 | DD 2210 LS | CJH1011 | HITAM | AYA | NULL | 100000 |  
+-----+-----+-----+-----+-----+-----+-----+  
2 rows in set (0.001 sec)
```

Kesimpulan

query `SELECT * FROM daftar_mobil WHERE warna = "hitam" ORDER BY harga_rental ASC LIMIT 2;` akan menampilkan informasi 2 mobil berwarna hitam dengan harga rental terendah dari tabel "daftar_mobil".

DISTINCT

Penjelasan

untuk menyeleksi data yang duplikat

PENJELASAN 1

1. `SELECT DISTINCT(pemilik)` : Memilih nilai unik dari kolom 'pemilik'. Penggunaan `DISTINCT` menghilangkan duplikat dan hanya mengembalikan nilai unik.
2. `FROM mobil` : Menunjukkan bahwa data diambil dari tabel 'mobil'.

PENJELASAN 2

1. `SELECT DISTINCT(harga_rental)` : Memilih nilai unik dari kolom 'harga_rental'. `DISTINCT` digunakan untuk menghilangkan nilai yang sama dan hanya mengembalikan nilai yang unik.
2. `FROM mobil` : Menunjukkan bahwa data diambil dari tabel 'mobil'.
3. `ORDER BY harga_rental DESC` : Mengurutkan hasil berdasarkan kolom 'harga_rental' secara descending. Ini berarti data akan diurutkan dari nilai tertinggi ke terendah berdasarkan harga rental.

Struktur Query

```
SELECT DISTINCT[(nama_kolom)] FROM [nama_tabel];
```

```
SELECT DISTINCT[(nama_kolom)] FROM [nama_tabel] ORDER BY ]nama_kolom] DESC;
```

Contoh Query

```
SELECT DISTINCT(pemilik) FROM mobil;
```

```
SELECT DISTINCT(harga_rental) FROM mobil ORDER BY harga_rental DESC;
```

Hasil

```
MariaDB [rental_alya]> SELECT DISTINCT(pemilik) FROM mobil;
+-----+
| pemilik |
+-----+
| REZKY   |
| ALYA    |
| AYA     |
+-----+
3 rows in set (0.001 sec)
```

```
MariaDB [rental_alya]> SELECT DISTINCT(harga_rental) FROM mobil ORDER BY harga_rental DESC;
+-----+
| harga_rental |
+-----+
| 150000       |
| 100000       |
| 50000        |
+-----+
3 rows in set (0.001 sec)
```

Kesimpulan

KESIMPULAN 1

Perintah `SELECT DISTINCT(pemilik)` digunakan untuk memilih nilai unik dari kolom 'pemilik'. Dengan menggunakan `DISTINCT`, duplikat dihilangkan sehingga hanya nilai unik yang akan dikembalikan. Selanjutnya, `FROM mobil` menunjukkan bahwa data diambil dari tabel 'mobil'.

KESIMPULAN 2

Perintah `SELECT DISTINCT(harga_rental)` digunakan untuk memilih nilai unik dari kolom 'harga_rental'. Penggunaan `DISTINCT` menghilangkan nilai yang sama dan hanya

mengembalikan nilai yang unik. Selanjutnya, `FROM mobil` menunjukkan bahwa data diambil dari tabel 'mobil'. Kemudian, `ORDER BY harga_rental DESC` mengurutkan hasil berdasarkan kolom 'harga_rental' secara descending, artinya data akan diurutkan dari nilai tertinggi ke terendah berdasarkan harga rental.

CONCAT, CONCAT_WS, AS

Menggabungkan kolom tanpa pemisah

Penjelasan

1. `SELECT CONCAT(pemilik, warna)` : Menggunakan fungsi `CONCAT()` untuk menggabungkan nilai dari kolom 'pemilik' dan 'warna' menjadi satu nilai. Hasilnya akan berupa nilai yang merupakan penggabungan dari nilai 'pemilik' dan 'warna' tanpa ada pemisah di antaranya.
2. `FROM mobil` : Menunjukkan bahwa data diambil dari tabel 'mobil'.

Struktur Query

```
SELECT CONCAT(nama_kolom) FROM nama_tabel;
```

Contoh Query

```
SELECT CONCAT(pemilik, warna) FROM mobil;
```

Hasil

```
MariaDB [rental_alya]> SELECT CONCAT(pemilik, warna) FROM mobil;
+-----+
| CONCAT(pemilik, warna) |
+-----+
| REZKYHITAM             |
| REZKYMERAH            |
| ALYASILVER              |
| AYA HITAM               |
| AYA HITAM               |
+-----+
5 rows in set (0.001 sec)
```

```
MariaDB [rental_alya]> SELECT CONCAT(pemilik, " ", warna) FROM mobil;
+-----+
| CONCAT(pemilik, " ", warna) |
+-----+
| REZKY HITAM                 |
| REZKY MERAH                |
| ALYA SILVER                  |
| AYA HITAM                    |
| AYA HITAM                    |
+-----+
```

Kesimpulan

query `SELECT CONCAT(pemilik, warna) FROM mobil;` akan menghasilkan satu kolom yang berisi informasi kombinasi antara pemilik dan warna mobil dari tabel "mobil".

Menggabungkan kolom dengan pemisah

Penjelasan

1. `SELECT CONCAT_WS("-", no_plat, no_mesin, id_mobil);` Menggunakan fungsi `CONCAT_WS()` untuk menggabungkan nilai dari kolom 'no_plat', 'no_mesin', dan 'id_mobil' dengan pemisah "-" di antara setiap nilai.
2. `FROM mobil;` Menunjukkan bahwa data diambil dari tabel 'mobil'.

Struktur Query

```
SELECT CONCAT_WS("-", nama2_kolom) FROM nama_tabel;
```

Contoh Query

```
SELECT CONCAT_WS("-",no_plat,no_mesin,id_mobil) FROM mobil;
```

Hasil

```
MariaDB [rental_alya]> SELECT CONCAT_WS("-",no_plat,no_mesin,id_mobil) FROM mobil;
+-----+
| CONCAT_WS("-",no_plat,no_mesin,id_mobil) |
+-----+
| DD 2650 XY-ACX3560-1                      |
| DD 2440 AX-BCS1120-2                      |
| B 1617 QC-LSQ1112-3                      |
| DD 1717 RA-UQL1029-4                      |
| DD 2210 LS-CJH1011-5                      |
+-----+
5 rows in set (0.000 sec)
```

Kesimpulan

Perintah `SELECT CONCAT_WS("-", no_plat, no_mesin, id_mobil)` menggabungkan nilai dari kolom 'no_plat', 'no_mesin', dan 'id_mobil' dengan menggunakan fungsi `CONCAT_WS()`, di mana setiap nilai dipisahkan oleh tanda "-". Selanjutnya, query tersebut menunjukkan bahwa data diambil dari tabel 'mobil'.

Memberikan nama kolom alias

Penjelasan

1. `SELECT CONCAT_WS("+", pemilik, peminjam) AS COLLAB`: Menggunakan fungsi `CONCAT_WS()` untuk menggabungkan nilai dari kolom 'pemilik' dan 'peminjam' dengan pemisah "+" di antara setiap nilai. Hasilnya diberi alias 'COLLAB'.
2. `FROM mobil`: Menunjukkan bahwa data diambil dari tabel 'mobil'.

Struktur Query

```
SELECT CONCAT_WS("+",nama2_kolom) AS nama_kolom_baru FROM nama_tabel;
```

Contoh Query

```
SELECT CONCAT_WS("+",pemilik,peminjaman) AS COLLAB FROM mobil;
```

Hasil

```
MariaDB [rental_alya]> SELECT CONCAT_WS("+",pemilik,peminjaman) AS COLLAB FROM mobil;
+-----+
| COLLAB |
+-----+
| REZKY+QILA |
| REZKY+ACCY |
| ALYA+VICKY |
| AYA       |
| AYA       |
+-----+
5 rows in set (0.001 sec)
```

Kesimpulan

query `SELECT CONCAT_WS("+",pemilik,peminjaman) AS COLLAB FROM mobil;` akan menghasilkan satu kolom baru bernama "COLLAB" yang berisi informasi kombinasi antara pemilik dan peminjaman mobil, dengan pemisah berupa tanda "+" (plus).

VIEW

Membuat tabel virtual

Penjelasan

1. `CREATE VIEW info_no_plat AS` : Perintah ini digunakan untuk membuat view baru dengan nama `info_no_plat`.
2. `SELECT id_mobil, no_plat, pemilik, peminjam FROM mobil WHERE pemilik = "REZKY";` : Ini adalah query yang akan menjadi isi dari view `info_no_plat`. Query ini mengambil kolom `id_mobil`, `no_plat`, `pemilik`, dan `peminjam` dari tabel `mobil` hanya untuk baris-baris di mana `pemilik` adalah "REZKY".

Struktur Query

```
CREATE VIEW nama_kolom_baru AS
-> SELECT nama2_kolom
-> FROM nama_tabel;
-> WHERE nama_kolom = isi_kolom;
```

Contoh Query

```
CREATE VIEW info_no_plat AS
-> SELECT id_mobil, no_plat, pemilik, peminjaman
-> FROM mobil
-> WHERE pemilik = "REZKY";
```

Hasil

```
MariaDB [rental_alya]> CREATE VIEW info_no_plat AS
-> SELECT id_mobil, no_plat, pemilik, peminjaman
-> FROM mobil
-> WHERE pemilik = "REZKY";
Query OK, 0 rows affected (0.048 sec)
```

```
MariaDB [rental_alya]> show tables;
```

```
+-----+
| Tables_in_rental_alya |
+-----+
| akun                  |
| info_no_plat          |
| mobil                 |
+-----+
3 rows in set (0.000 sec)
```

Kesimpulan

Perintah `CREATE VIEW info_no_plat AS` digunakan untuk membuat view baru dengan nama `info_no_plat`, sedangkan query `SELECT id_mobil, no_plat, pemilik, peminjam FROM mobil WHERE pemilik = "REZKY";` akan menjadi isi dari view tersebut. Query tersebut mengambil kolom `id_mobil`, `no_plat`, `pemilik`, dan `peminjam` dari tabel `mobil` hanya untuk baris-baris di mana nilai kolom `pemilik` adalah "REZKY".

Menampilkan tabel virtual

Penjelasan

- `SELECT *` berarti memilih semua kolom yang ada di dalam tabel.
- `FROM info_no_plat` menentukan tabel yang akan diambil datanya, yaitu tabel `info_no_plat`.

Struktur Query

```
SELECT * FROM nama_tabel_baru;
```

Contoh Query

```
SELECT * FROM info_no_plat;
```

Hasil

```
MariaDB [rental_alya]> select * from info_no_plat;
+-----+-----+-----+-----+
| id_mobil | no_plat      | pemilik | peminjaman |
+-----+-----+-----+-----+
|          1 | DD 2650 XY  | REZKY   | QILA        |
|          2 | DD 2440 AX  | REZKY   | ACCY        |
+-----+-----+-----+-----+
2 rows in set (0.001 sec)
```

Kesimpulan

Query `SELECT * FROM info_no_plat;` akan mengembalikan semua data yang ada di dalam tabel `info_no_plat`, dengan menampilkan semua kolom yang tersedia di dalam tabel tersebut. Tabel `info_no_plat` mungkin berisi informasi terkait nomor plat kendaraan, seperti jenis kendaraan, pemilik, tanggal registrasi, dan informasi lainnya.

Menghapus tabel virtual

Penjelasan

- `DROP VIEW`: Ini adalah perintah untuk menghapus view.
- `info_no_plat`: Nama view yang akan dihapus.

Struktur Query

```
DROP VIEW nama_tabel_baru;
```

Contoh Query

```
DROP VIEW info_no_plat;
```

Hasil

BEFORE

```
MariaDB [rental_alya]> show tables;
```

```
+-----+  
| Tables_in_rental_alya |  
+-----+  
| akun                   |  
| car                    |  
| daftar_mobil           |  
| info_no_plat           |  
| mobil_tanpa_peminjam   |  
| tantangan               |  
+-----+
```

AFTER

```
MariaDB [rental_alya]> DROP VIEW info_no_plat;  
Query OK, 0 rows affected (0.001 sec)
```

```
MariaDB [rental_alya]> show tables;
```

```
+-----+  
| Tables_in_rental_alya |  
+-----+  
| akun                  |  
| car                   |  
| daftar_mobil          |  
| mobil_tanpa_peminjam  |  
| tantangan             |  
+-----+
```

Kesimpulan

Perintah ini digunakan untuk menghapus view dengan nama `info_no_plat` dari database. Ketika perintah ini dijalankan, view tersebut akan dihapus dan tidak akan lagi tersedia untuk digunakan. Ini memungkinkan Anda untuk membersihkan definisi view yang tidak lagi diperlukan dari database Anda.

Tantangan View

Nomor 1

Penjelasan

`CREATE VIEW mobil_tanpa_peminjam AS` : adalah perintah untuk membuat sebuah view baru atau seperti tabel baru dalam basis data dengan nama `mobil_tanpa_peminjam`.

- `SELECT no_plat, peminjaman` : adalah perintah untuk memilih dua kolom, yaitu `no_plat` dan `peminjam`, dari tabel `mobil`.
- `FROM mobil` : Menunjukkan bahwa data diambil dari tabel bernama `mobil`.
- `WHERE peminjam IS NULL` : adalah klausa `WHERE` yang mencari baris-baris dari tabel `mobil` dimana nilai kolom `peminjam` adalah `NULL`.
- `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

```
CREATE VIEW
-> mobil_tanpa_peminjam AS
-> SELECT no_plat,peminjaman
-> FROM mobil
-> WHERE peminjaman IS NULL;
```

Hasil

```
MariaDB [rental_alya]> CREATE VIEW
```

```
-> mobil_tanpa_peminjam AS
-> SELECT no_plat,peminjaman
-> FROM mobil
-> WHERE peminjaman IS NULL;
```

```
Query OK, 0 rows affected (0.047 sec)
```

```
MariaDB [rental_alya]> SELECT * FROM mobil_tanpa_peminjam;
```

```
+-----+-----+
| no_plat | peminjaman |
+-----+-----+
| B 1617 QC | NULL      |
| DD 1717 RA | NULL      |
| DD 2210 LS | NULL      |
| DD 8347 DT | NULL      |
+-----+-----+
```

```
4 rows in set (0.002 sec)
```

Kesimpulan

CREATE VIEW mobil_tanpa_peminjam AS Select no_plat, peminjaman FROM mobil WHERE peminjaman IS NULL; digunakan untuk membuat sebuah view baru bernama mobil_Tanpa_peminjam. Viewnya berisi dua kolom, yaitu no_plat dan peminjaman, yang diambil dari tabel mobil hanya baris-baris yang memiliki nilai NULL pada kolom peminjam yang dimasukkan ke dalam view.

SELECT * FROM mobil_tanpa_peminjam; digunakan untuk menampilkan semua data dari view mobil_Tanpa_peminjam, yang telah dibuat sebelumnya dengan kriteria yang bernilai NULL.

Nomor 2

Penjelasan

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

- SET peminjaman = NULL : menetapkan nilai kolom peminjam menjadi NULL.
- WHERE peminjam= 'ALYA' : adalah klausa WHERE yang membatasi update hanya pada baris-baris dimana nilai kolom peminjam adalah 'ALYA'. Maksudnya perubahan hanya akan berlaku untuk baris-baris yang memiliki peminjam dengan nama 'ALYA'.
- 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 = 'ALYA';
```

Hasil

```

MariaDB [rental_alya]> UPDATE mobil
  -> SET peminjaman = NULL
  -> WHERE peminjaman = 'ALYA';
Query OK, 0 rows affected (0.001 sec)
Rows matched: 0  Changed: 0  Warnings: 0

MariaDB [rental_alya]> SELECT * FROM mobil_tanpa_peminjam;
+-----+-----+
| no_plat | peminjaman |
+-----+-----+
| B 1617 QC | NULL |
| DD 1717 RA | NULL |
| DD 2210 LS | NULL |
| DD 8347 DT | NULL |
+-----+-----+
4 rows in set (0.001 sec)

```

Kesimpulan

UPDATE mobil SET peminjaman = NULL WHERE peminjaman = 'ALYA'; nilai pada kolom peminjaman pada tabel mobil yang memiliki nilai 'ALYA' akan diubah menjadi NULL. Kesimpulannya, perintah digunakan untuk menghapus atau mengubah nilai peminjaman menjadi NULL untuk semua data di tabel mobil yang berada di kolom peminjaman memiliki nilai 'ALYA'.

SELECT * FROM mobil_tanpa_peminjam; digunakan untuk menampilkan semua data dari view mobil_tanpa_peminjam, yang telah dibuat sebelumnya dengan mengubah atau menghapus nilai peminjam menjadi NULL untuk tabel mobil dimana peminjam memiliki nilai ALYA.

Nomor 3

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

Dengan membuat view, kita 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

SUM

Penjelasan

- `SELECT SUM(harga_rental)` : Fungsi `SUM()` digunakan untuk menghitung total atau jumlah dari semua nilai dalam kolom `harga_rental`.
- `FROM mobil` : Menentukan tabel `mobil` dari mana data akan diambil.

Struktur Query

```
SELECT SUM(nama_kolom) FROM nama_tabel;
```

Contoh Query

```
SELECT SUM(harga_rental) FROM mobil;
```

Hasil

```
MariaDB [rental_alya]> SELECT SUM(harga_rental) FROM mobil;
+-----+
| SUM(harga_rental) |
+-----+
|           600000 |
+-----+
1 row in set (0.001 sec)

MariaDB [rental_alya]>
```

Kesimpulan

Query `SELECT SUM(harga_rental) FROM mobil;` akan menghitung dan menampilkan total atau jumlah keseluruhan dari nilai-nilai yang terdapat pada kolom `harga_rental` di dalam tabel `mobil`. Hasil query ini akan memberikan informasi total atau keseluruhan harga rental untuk semua mobil yang ada di dalam tabel tersebut.

Count

Penjelasan

PENJELASAN 1

- `SELECT COUNT(pemilik)` - Fungsi `COUNT()` digunakan untuk menghitung jumlah baris atau record yang ada di dalam kolom `pemilik`.
- `FROM mobil` - Menentukan tabel `mobil` dari mana data akan diambil.

PENJELASAN 2

- `SELECT COUNT(peminjaman)` - Fungsi `COUNT()` digunakan untuk menghitung jumlah baris atau record yang ada di dalam kolom `peminjaman`.
- `FROM mobil` - Menentukan tabel `mobil` dari mana data akan diambil.

Struktur Query

```
SELECT COUNT(nama_kolom) FROM nama_tabel;
```

```
SELECT COUNT(nama_kolom) FROM nama_tabel;
```

Contoh Query

```
SELECT COUNT(pemilik) FROM mobil;
```

```
SELECT COUNT(peminjaman) FROM mobil;
```

Hasil

```
MariaDB [rental_alya]> SELECT COUNT(pemilik) FROM mobil;
+-----+
| COUNT(pemilik) |
+-----+
|                6 |
+-----+
1 row in set (0.001 sec)
```

```

MariaDB [rental_alya]> SELECT COUNT(peminjaman) FROM mobil;
+-----+
| COUNT(peminjaman) |
+-----+
|                2 |
+-----+
1 row in set (0.001 sec)

MariaDB [rental_alya]> _

```

Kesimpulan

KESIMPULAN 1

Query `SELECT COUNT(pemilik) FROM mobil;` akan menghitung dan menampilkan jumlah total pemilik mobil yang ada di dalam tabel `mobil`. Hasil query ini akan memberikan informasi tentang berapa banyak pemilik mobil yang tercatat di dalam tabel tersebut.

KESIMPULAN 2

Query `SELECT COUNT(peminjaman) FROM mobil;` akan menghitung dan menampilkan jumlah total peminjaman mobil yang ada di dalam tabel `mobil`. Hasil query ini akan memberikan informasi tentang berapa banyak peminjaman mobil yang tercatat di dalam tabel tersebut.

MIN

Penjelasan

- `SELECT MIN(harga_rental)` : Fungsi `MIN()` digunakan untuk mencari nilai minimum (terkecil) dari kolom `harga_rental`.
- `AS MINIMAL` : Hasil dari fungsi `MIN()` akan ditampilkan dengan nama alias "MINIMAL".
- `FROM mobil` : Menentukan tabel `mobil` dari mana data akan diambil.

Struktur Query

```
SELECT MIN(nama_kolom) AS nilai_minimum FROM nama_tabel;
```

Contoh Query

```
SELECT MIN(harga_rental) AS MINIMAL FROM mobil;
```

Hasil

```
MariaDB [rental_alya]> SELECT MIN(harga_rental) AS MINIMAL FROM mobil;
+-----+
| MINIMAL |
+-----+
| 50000 |
+-----+
1 row in set (0.111 sec)

MariaDB [rental_alya]>
```

Kesimpulan

Query `SELECT MIN(harga_rental) AS MINIMAL FROM mobil;` akan mencari dan menampilkan harga rental mobil yang paling murah atau terkecil dari semua data yang ada di dalam tabel `mobil`. Hasil query ini akan memberikan informasi tentang harga rental mobil terendah yang tercatat di dalam tabel tersebut.

MAX

Penjelasan

- `SELECT MAX(harga_rental)` : Fungsi `MAX()` digunakan untuk mencari nilai maksimum (terbesar) dari kolom `harga_rental`.
- `AS MAXIMAL` : Hasil dari fungsi `MAX()` akan ditampilkan dengan nama alias "MAXIMAL".
- `FROM mobil` : Menentukan tabel `mobil` dari mana data akan diambil.

Struktur Query

```
SELECT MAX(nama_kolom) AS nilai_minimum FROM nama_tabel;
```

Contoh Query

```
SELECT MAX(harga_rental) AS MAXIMAL FROM mobil;
```

Hasil

```
MariaDB [rental_alya]> SELECT MAX(harga_rental) AS MAXIMAL FROM mobil;
+-----+
| MAXIMAL |
+-----+
| 150000 |
+-----+
1 row in set (0.000 sec)

MariaDB [rental_alya]>
```

Kesimpulan

Query `SELECT MAX(harga_rental) AS MAXIMAL FROM mobil;` akan mencari dan menampilkan harga rental mobil yang paling mahal atau terbesar dari semua data yang ada di dalam tabel `mobil`. Hasil query ini akan memberikan informasi tentang harga rental mobil tertinggi yang tercatat di dalam tabel tersebut.

AVG

Penjelasan

- `SELECT AVG(harga_rental)` : Fungsi `AVG()` digunakan untuk menghitung rata-rata (average) dari nilai-nilai pada kolom `harga_rental`.
- `AS RATA_RATA` : Hasil dari fungsi `AVG()` akan ditampilkan dengan nama alias "RATA_RATA".
- `FROM mobil` : Menentukan tabel `mobil` dari mana data akan diambil.

Struktur Query

```
SELECT AVG(nama_kolom) AS rata_rata FROM nama_tabel;
```

Contoh Query

```
SELECT AVG(harga_rental) AS RATA_RATA FROM mobil;
```

Hasil


```
MariaDB [rental_alya]> SELECT AVG(harga_rental) AS RATA_RATA FROM mobil;
+-----+
| RATA_RATA |
+-----+
| 100000.0000 |
+-----+
1 row in set (0.001 sec)
```

Kesimpulan

Query `SELECT AVG(harga_rental) AS RATA_RATA FROM mobil;` akan menghitung dan menampilkan rata-rata (average) harga rental mobil dari semua data yang ada di dalam tabel `mobil`. Hasil query ini akan memberikan informasi tentang harga rental mobil rata-rata yang tercatat di dalam tabel tersebut.

GROUP BY & HAVING

Tantangan 7 Nomor

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

Struktur Query

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

Contoh Query

```
select warna,COUNT(id_mobil) AS Jumlah_Data_Mobil FROM daftar_mobil GROUP BY
warna;
```

Hasil

```
MariaDB [rental_alya]> select warna,COUNT(id_mobil) AS Jumlah_Data_Mobil FROM daftar_mobil GROUP BY warna;
```

warna	Jumlah_Data_Mobil
BIRU	2
HIJAU	1
HITAM	5
MERAH	2
PUTIH	1
SILVER	2

```
6 rows in set (0.001 sec)
```

Analisis

- `SELECT warna` : Memilih kolom warna dari tabel data_mobil.
- `COUNT(id_mobil) AS Jumlah_Data_Mobil` : Menghitung jumlah baris (mobil) untuk setiap warna unik dan memberi alias Jumlah_Data_Mobil pada hasil hitungan tersebut.
- `FROM daftar_mobil` : Menentukan tabel daftar_mobil sebagai sumber data.
- `GROUP BY warna` : Mengelompokkan hasil query berdasarkan nilai di kolom warna. Setiap nilai unik dalam kolom warna akan menjadi satu grup.

Kesimpulan

1. Mengelompokkan Data Berdasarkan Warna: Data dalam tabel daftar_mobil dikelompokkan berdasarkan kolom warna.
2. Menghitung Jumlah Mobil untuk Setiap Warna: Menggunakan fungsi COUNT(id_mobil) untuk menghitung jumlah mobil dalam setiap grup warna.
3. Memberikan Hasil yang Jelas: Hasil dari query ini menunjukkan jumlah mobil untuk setiap warna dalam tabel data_mobil, dengan kolom Jumlah_Data_Mobil menunjukkan hitungan tersebut.

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

Struktur Query

```
select nama_data,COUNT(nama_data) AS nama_sementara from nama_tabel GROUP BY
nama_data HAVING COUNT(nama_data) >= 3;
```

Contoh Query

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

Hasil

```
MariaDB [rental_alya]> select pemilik,COUNT(id_mobil) AS jumlah_mobil from daftar_mobil GROUP BY pemilik HAVING COUNT(id_mobil) >= 3;
```

pemilik	jumlah_mobil
ALIF	3
ANDI	3
REZKY	4

3 rows in set (0.001 sec)

Analisis

1. SELECT Klausa pemilik: Kolom ini dipilih dari tabel data_mobil. Kolom pemilik berisi data tentang pemilik mobil.
2. COUNT(id_mobil) AS jumlah_mobil: Fungsi agregat COUNT digunakan untuk menghitung jumlah baris dalam setiap grup yang memiliki pemilik yang sama. Hasil hitungan ini diberi alias jumlah_mobil, sehingga dalam hasil akhir, kolom ini akan diberi nama jumlah_mobil.
3. FROM Klausa daftar_mobil: Tabel ini merupakan sumber data dari query. Tabel ini diasumsikan berisi data mobil, termasuk kolom pemilik dan id_mobil.
4. GROUP BY pemilik: Pernyataan ini mengelompokkan baris-baris data berdasarkan nilai dalam kolom pemilik. Semua baris yang memiliki nilai pemilik yang sama akan dimasukkan ke dalam grup yang sama.
5. HAVING COUNT(id_mobil) >= 3: Pernyataan ini menyaring grup-grup yang terbentuk berdasarkan hasil agregat. Hanya grup yang memiliki jumlah baris (mobil) setidaknya 3 yang akan dimasukkan dalam hasil akhir. HAVING digunakan setelah pengelompokan data, berbeda dengan WHERE yang digunakan sebelum pengelompokan.

Kesimpulan

1. Mengelompokkan Data Berdasarkan Pemilik: Data dalam tabel daftar_mobil dikelompokkan berdasarkan kolom pemilik.
2. Menghitung Jumlah Mobil untuk Setiap Pemilik: Menggunakan fungsi COUNT(id_mobil) untuk menghitung jumlah mobil dalam setiap grup pemilik. Hasil hitungan ini diberi alias jumlah_mobil.
3. Menyaring Grup dengan Klausa HAVING: Menggunakan klausa HAVING untuk menyaring dan hanya menampilkan grup yang memiliki jumlah mobil (baris) setidaknya 3.

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;
```

Contoh Query

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

Hasil

```
MariaDB [rental_alya]> SELECT pemilik,  
-> COUNT(id_mobil) AS jumlah_mobil  
-> FROM daftar_mobil GROUP BY pemilik;
```

```
+-----+-----+  
| pemilik | jumlah_mobil |  
+-----+-----+  
| ALIF    | 3            |  
| ALYA    | 1            |  
| ANDI    | 3            |  
| AYA     | 2            |  
| REZKY   | 4            |  
+-----+-----+  
5 rows in set (0.001 sec)
```

Analisis

- `SELECT` merupakan perintah SQL yang digunakan untuk memilih data dari database.
- `pemilik` adalah nama kolom yang akan diambil dari tabel `data_mobil`.
- `COUNT(id_mobil)` adalah fungsi yang digunakan untuk menghitung jumlah baris dalam kolom `id_mobil`.
- `AS jumlah_mobil` memberikan alias pada hasil perhitungan `COUNT(id_mobil)` sehingga hasilnya akan diberi nama `jumlah_mobil`.
- `FROM daftar_mobil` menentukan tabel `daftar_mobil` sebagai sumber data.
- `GROUP BY pemilik` mengelompokkan data berdasarkan kolom `pemilik` dan melakukan perhitungan `COUNT` untuk setiap kelompok.

Kesimpulan

Perintah SQL ini akan menghasilkan daftar pemilik mobil beserta jumlah mobil yang dimiliki oleh masing-masing pemilik. Hasil query akan menampilkan dua kolom: `pemilik` yang berisi nama pemilik, dan `jumlah_mobil` yang berisi jumlah mobil yang dimiliki oleh pemilik tersebut. Perintah `GROUP BY` memastikan bahwa perhitungan `COUNT(id_mobil)` dilakukan untuk setiap pemilik secara terpisah.

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;
```

Contoh Query

```
select pemilik,SUM(harga_rental) AS jumlah_pendapatan from daftar_mobil GROUP BY pemilik;
```

Hasil

```
MariaDB [rental_alya]> select pemilik,SUM(harga_rental) AS jumlah_pendapatan from daftar_mobil GROUP BY pemilik;
```

pemilik	jumlah_pendapatan
ALIF	360000
ALYA	50000
ANDI	285000
AYA	250000
REZKY	380000

```
5 rows in set (0.051 sec)
```

Analisis

- **SELECT** merupakan perintah yang digunakan untuk memilih data dari database.
- **pemilik** adalah nama kolom yang akan diambil dari tabel `data_mobil`.
- **SUM(harga_rental)** adalah fungsi yang digunakan untuk menghitung total nilai dari kolom `harga_rental`.
- **AS jumlah_pendapatan** memberikan alias pada hasil perhitungan `SUM(harga_rental)` sehingga hasilnya akan diberi nama `jumlah_pendapatan`.
- **FROM daftar_mobil** menentukan tabel `daftar_mobil` sebagai sumber data.

- **GROUP BY pemilik** mengelompokkan data berdasarkan kolom `pemilik` dan melakukan perhitungan `SUM` untuk setiap kelompok.

Kesimpulan

Perintah SQL ini akan menghasilkan daftar pemilik mobil beserta total pendapatan dari harga rental yang mereka miliki. Hasil query akan menampilkan dua kolom: `pemilik` yang berisi nama pemilik, dan `jumlah_pendapatan` yang berisi total pendapatan dari harga rental mobil untuk setiap pemilik.

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;
```

Contoh Query

```
select pemilik,SUM(harga_rental) AS jumlah_pemasukan from daftar_mobil GROUP BY pemilik HAVING SUM(harga_rental) >= 300000;
```

Hasil

```
MariaDB [rental_alya]> select pemilik,SUM(harga_rental) AS jumlah_pemasukan from daftar_mobil GROUP BY pemilik HAVING SUM(harga_rental) >= 300000;
+-----+-----+
| pemilik | jumlah_pemasukan |
+-----+-----+
| ALIF    | 360000            |
| REZKY   | 380000            |
+-----+-----+
2 rows in set (0.002 sec)
```

Analisis

- **SELECT** merupakan perintah yang digunakan untuk memilih data dari database.
- **pemilik** adalah nama kolom yang akan diambil dari tabel `daftar_mobil`.
- **SUM(harga_rental)** adalah fungsi yang digunakan untuk menghitung total nilai dari kolom `harga_rental`.

- **AS jumlah_pemasukan** memberikan alias pada hasil perhitungan `SUM(harga_rental)` sehingga hasilnya akan diberi nama `jumlah_pemasukan`.
- **FROM daftar_mobil** menentukan tabel `daftar_mobil` sebagai sumber data.
- **GROUP BY pemilik** mengelompokkan data berdasarkan kolom `pemilik` dan melakukan perhitungan `SUM` untuk setiap kelompok.
- **HAVING SUM(harga_rental) >= 300000** merupakan klausa yang digunakan untuk menyaring kelompok hasil perhitungan `SUM(harga_rental)` yang nilainya lebih besar atau sama dengan 300000.

Kesimpulan

Perintah SQL ini akan menghasilkan daftar pemilik mobil beserta total pendapatan dari harga rental yang mereka miliki, tetapi hanya untuk pemilik yang total pendapatannya sama dengan atau lebih dari 300000. Hasil query akan menampilkan dua kolom: `pemilik` yang berisi nama pemilik, dan `jumlah_pemasukan` yang berisi total pendapatan dari harga rental mobil untuk setiap pemilik yang memenuhi kriteria `HAVING` tersebut.

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;
```

Contoh Query

```
select pemilik,AVG(harga_rental) AS rata_pemasukan from daftar_mobil GROUP BY
pemilik;
```

Hasil

```
MariaDB [rental_alya]> select pemilik,AVG(harga_rental) AS rata_pemasukan from daftar_mobil GROUP BY pemilik;
+-----+-----+
| pemilik | rata_pemasukan |
+-----+-----+
| ALIF    | 120000.0000    |
| ALYA    | 50000.0000     |
| ANDI    | 95000.0000     |
| AYA     | 125000.0000    |
| REZKY   | 95000.0000     |
+-----+-----+
5 rows in set (0.001 sec)
```

Analisis

- **SELECT** merupakan perintah yang digunakan untuk memilih data dari database.
- **pemilik** adalah nama kolom yang akan diambil dari tabel `daftar_mobil`. Kolom ini menyimpan informasi tentang pemilik mobil.
- **AVG(harga_rental)** adalah fungsi yang digunakan untuk menghitung nilai rata-rata dari kolom `harga_rental`.
- **AS rata_pemasukan** memberikan alias pada hasil perhitungan `AVG(harga_rental)` sehingga hasilnya akan diberi nama `rata_pemasukan`.
- **FROM data_mobil** menentukan tabel `daftar_mobil` sebagai sumber data.
- **GROUP BY pemilik** mengelompokkan data berdasarkan kolom `pemilik` dan melakukan perhitungan `AVG` untuk setiap kelompok.

Kesimpulan

Perintah SQL ini akan menghasilkan daftar pemilik mobil beserta nilai rata-rata pendapatan dari harga rental yang mereka miliki. Hasil query akan menampilkan dua kolom: `pemilik` yang berisi nama pemilik, dan `rata_pemasukan` yang berisi rata-rata pendapatan dari harga rental mobil untuk setiap pemilik.

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

Struktur Query

```
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 Query

```
select pemilik, MAX(harga_rental) AS Pemasukan_Terbesar, MIN(harga_rental) AS  
pemasukan_terkecil from daftar_mobil GROUP BY pemilik HAVING  
COUNT(harga_rental) > 1;
```

Hasil


```

MariaDB [rental_alya]> select pemilik,MAX(harga_rental) AS Pemasukan_Terbesar,
-> MIN(harga_rental) AS pemasukan_terkecil from daftar_mobil
-> GROUP BY pemilik HAVING COUNT(harga_rental) > 1;
+-----+-----+-----+
| pemilik | Pemasukan_Terbesar | pemasukan_terkecil |
+-----+-----+-----+
| ALIF    | 130000             | 110000             |
| ANDI    | 105000             | 85000              |
| AYA     | 150000             | 100000             |
| REZKY   | 140000             | 50000              |
+-----+-----+-----+
4 rows in set (0.001 sec)

MariaDB [rental_alya]> _

```

Analisis

- **SELECT** merupakan perintah yang digunakan untuk memilih data dari database.
- **pemilik** adalah nama kolom yang akan diambil dari tabel `data_mobil`. Kolom ini menyimpan informasi tentang pemilik mobil.
- **MAX(harga_rental)** adalah fungsi yang digunakan untuk menghitung nilai maksimum dari kolom `harga_rental`.
- **AS Pemasukan_Terbesar** memberikan alias pada hasil perhitungan `MAX(harga_rental)` sehingga hasilnya akan diberi nama `Pemasukan_Terbesar`.
- **MIN(harga_rental)** adalah fungsi yang digunakan untuk menghitung nilai minimum dari kolom `harga_rental`.
- **AS pemasukan_terkecil** memberikan alias pada hasil perhitungan `MIN(harga_rental)` sehingga hasilnya akan diberi nama `pemasukan_terkecil`.
- **FROM daftar_mobil** menentukan tabel `daftar_mobil` sebagai sumber data.
- **GROUP BY pemilik** mengelompokkan data berdasarkan kolom `pemilik` dan melakukan perhitungan `MAX` dan `MIN` untuk setiap kelompok.
- **HAVING COUNT(harga_rental) > 1** merupakan klausa yang digunakan untuk menyaring kelompok yang memiliki lebih dari satu baris data di kolom `harga_rental`.

Kesimpulan

Perintah SQL ini akan menghasilkan daftar pemilik mobil beserta nilai pemasukan terbesar dan pemasukan terkecil dari harga rental yang mereka miliki. Hasil query akan menampilkan 4 kolom: `pemilik` yang berisi nama pemilik, `Pemasukan_Terbesar` yang berisi nilai tertinggi dari harga rental, dan `pemasukan_terkecil` yang berisi nilai terendah dari harga rental untuk setiap pemilik yang memiliki lebih dari satu data rental.

RELASI

Data Tabel Pegawai

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

☐ Check all

With selected:

Edit

Copy

Delete

Export

Perubahan Struktur Tabel Pegawai

BEFORE

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

AFTER

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

Perubahan Data Tabel Pegawai & Cabang

Pegawai

BEFORE

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

☐ Profiling
 [[Edit inline](#)]
 [[Edit](#)]
 [[Explain SQL](#)]
 [[Create PHP code](#)]
 [[Refresh](#)]

☐ Show all
 | Number of rows:
 Filter rows:
 Sort by key:

Extra options

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

☐ Check all
 With selected:
 Edit
 Copy
 Delete
 Export

AFTER

				IdPeg	Nama	NamaKel	JK	Alamat	Kota	Telp	NoCab
<input type="checkbox"/>				10107	Rezky	Aya	P	Jl.Suci 78	Bandung	022-555768	C101
<input type="checkbox"/>				10176	Andi	Ashadelah	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

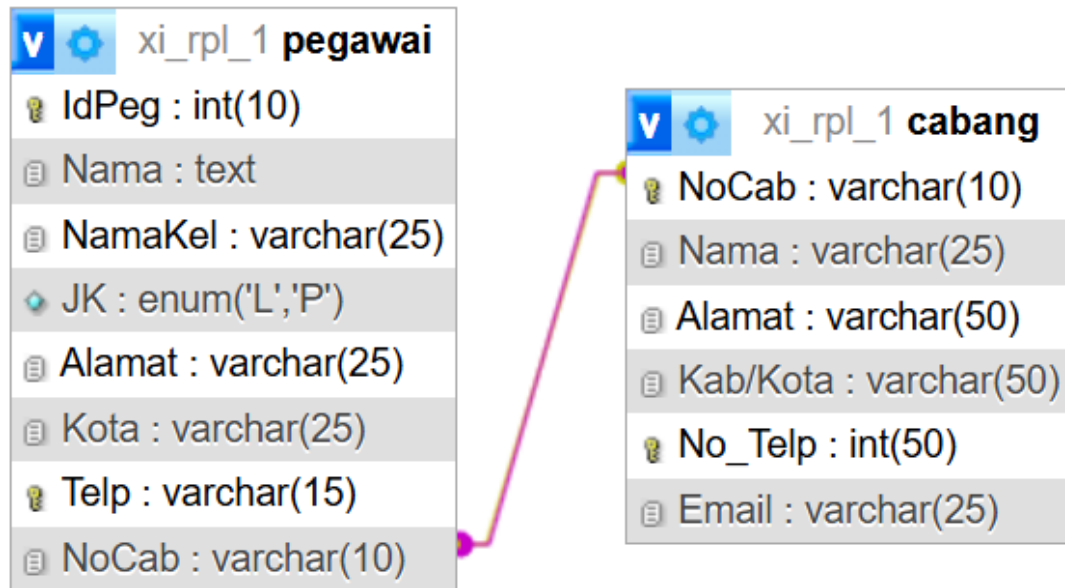
BEFORE

				NoCab	Nama	Alamat	Kab/Kota	No_Telp	Email
<input type="checkbox"/>				C101	Aqila	Jl.Korban	Soppeng	83473643	aqila@gmail.com
<input type="checkbox"/>				C102	Arsy	Jl. Rappocini	Bone	824872392	Arsy@gmail.com
<input type="checkbox"/>				C103	Adel	Jl. Maccini	Pare-Pare	847239238	adel@gmail.com
<input type="checkbox"/>				C104	Alya	Jl.Rappokalling	Makassar	852378749	alyarstm@gmail.com

AFTER

				NoCab	Nama	Alamat	Kab/Kota	No_Telp	Email
<input type="checkbox"/>				C101	Aqila	Jl.Korban	Soppeng	83473643	aqila@gmail.com
<input type="checkbox"/>				C102	Accy	Jl. Rappojawa	Sengkang	237823625	Accy@gmail.com
<input type="checkbox"/>				C103	Dela	Jl. Rappocini	Sinjai	82363562	Dela@gmail.com
<input type="checkbox"/>				C104	Alya	Jl.Rappokalling	Makassar	852378749	alyarstm@gmail.com

Hasil Relasi Pegawai & Cabang



Query & Hasil

Actions	Constraint properties	Column	Foreign key constraint (INNODB)		
			Database	Table	Column
	pegawai_ibfk_1				
Drop	ON DELETE CASCADE	NoCab	xi_rpl_1	cabang	NoCab
	ON UPDATE CASCADE	+ Add column			