Operator Logika & Pembanding

AND

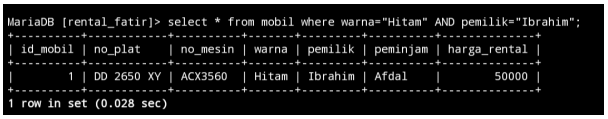
Struktur

```
SELECT FROM nama_table WHERE nama_kolom1="nilai_kolom" AND
nama_kolom2="nilai_kolom"
```

Contoh

```
select * from mobil where warna="Hitam" AND pemilik="Ibrahim";
```

Hasil



```
MariaDB [rental_fatir]> select * from mobil where warna="Hitam" AND pemilik="Ibrahim";
+----------+------------+----------+-------+---------+----------+-------------+
| id_mobil | no_plat    | no_mesin | warna | pemilik | peminjam | harga_rental |
+----------+------------+----------+-------+---------+----------+-------------+
|        1 | DD 2650 XY | ACX3560  | Hitam | Ibrahim | Afdal    |        50000 |
+----------+------------+----------+-------+---------+----------+-------------+
1 row in set (0.028 sec)
```

Analisis

- `SELECT *`: Bagian kueri ini menentukan bahwa Anda ingin mengambil semua kolom dari tabel.
- `FROM mobil`: Ini menunjukkan bahwa Anda menanyakan tabel "mobil".
- `WHERE warna="Hitam" AND pemilik="Ibrahim"` : Ini adalah kondisi untuk memfilter baris. Ini menetapkan bahwa Anda hanya menginginkan baris yang kolom "warna" sama dengan "Hitam" dan kolom "pemilik" sama dengan "Ibrahim".

Kesimpulan

`"SELECT * FROM mobil WHERE warna='Hitam' AND pemilik='Ibrahim'"` adalah bahwa perintah tersebut akan mengambil semua data dari tabel "mobil" yang memiliki nilai kolom `"warna"` sama dengan `"Hitam"` dan nilai kolom `"pemilik"` sama dengan `"Ibrahim"`, perintah ini akan mengembalikan semua baris dari tabel `"mobil"` yang memenuhi kedua kriteria tersebut.

OR

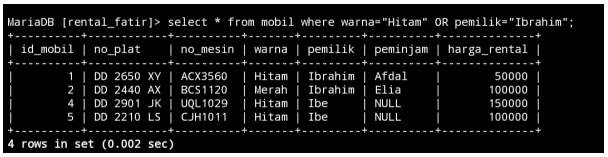
Struktur

```
SELECT FROM nama_table WHERE nama_kolom1="nilai_kolom" OR
nama_kolom2="nilai_kolom"
```

Contoh

```
 select * from mobil where warna="Hitam" OR pemilik="Ibrahim";
```

Hasil



```
MariaDB [rental_fatir]> select * from mobil where warna="Hitam" OR pemilik="Ibrahim";
+----------+------------+----------+-------+---------+----------+--------------+
| id_mobil | no_plat    | no_mesin | warna | pemilik | peminjam | harga_rental |
+----------+------------+----------+-------+---------+----------+--------------+
|        1 | DD 2650 XY | ACX3560  | Hitam | Ibrahim | Afdal    |        50000 |
|        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 |
+----------+------------+----------+-------+---------+----------+--------------+
4 rows in set (0.002 sec)
```

Analisis

- Kolom `"warna"` memiliki nilai `"Hitam"`.
- Kolom `"pemilik"` memiliki nilai `"Ibrahim"`.
  Dalam hal ini, perintah `SELECT *` digunakan untuk mengambil semua kolom (semua atribut) dari tabel "mobil". `FROM mobil` menunjukkan bahwa tabel yang dimaksud adalah “mobil”. Kondisi `WHERE` digunakan untuk memfilter baris-baris dalam tabel "mobil". Operator `OR` menunjukkan bahwa setidaknya salah satu kondisi harus dipenuhi agar baris tersebut diambil. Jadi, baris akan diambil jika warna mobil adalah "Hitam" atau jika pemilik mobil adalah "Ibrahim".

Kesimpulan

Kesimpulan `"SELECT * FROM mobil WHERE warna='Hitam' OR pemilik='Ibrahim';` adalah bahwa Anda sedang mencari semua data dari tabel "mobil" di mana nilai kolom "warna" sama dengan "Hitam" atau nilai kolom "pemilik" sama dengan "Ibrahim".

BETWEEN

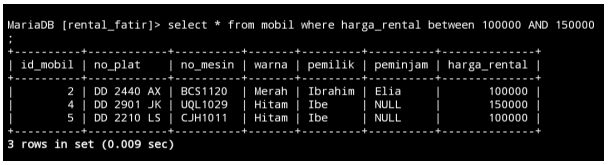
Struktur

```
SELECT * FROM nama_table WHERE nama_kolom BETWEEN nilai_kolom AND nilai_kolom
;
```

Contoh

```
select * from mobil WHERE harga_rental BETWEEN 100000 AND 150000;
```

Hasil



```
MariaDB [rental_fatir]> select * from mobil where harga_rental between 100000 AND 150000
;
+----------+------------+----------+-------+---------+----------+--------------+
| id_mobil | 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 (0.009 sec)
```

Analisis

- `SELECT` : Ini merupakan bagian dari perintah SELECT yang digunakan untuk menentukan kolom mana yang ingin Anda ambil dari tabel. Dalam hal ini, tanda "" digunakan untuk mengambil semua kolom yang ada di tabel "mobil".
- `FROM mobil` : Ini menentukan bahwa data akan diambil dari tabel bernama "mobil".
- `WHERE harga_rental BETWEEN 100000 AND 150000` : Ini adalah klausul WHERE yang digunakan untuk memfilter baris yang akan diambil berdasarkan kondisi tertentu. Dalam hal ini, kondisi yang digunakan adalah `"harga_rental BETWEEN 100000 AND 150000"`.
- Operator `BETWEEN` digunakan untuk memeriksa apakah nilai kolom "harga_rental" berada di antara dua angka yang diberikan, yaitu 100.000 dan 150.000.

Kesimpulan

Kesimpulan `"SELECT * FROM mobil WHERE harga_rental BETWEEN 100000 AND 150000;"` adalah bahwa perintah tersebut digunakan untuk mengambil semua data dari tabel "mobil" yang memenuhi kondisi harga_rental berada di antara 100.000 dan 150.000.

NOT BETWEEN

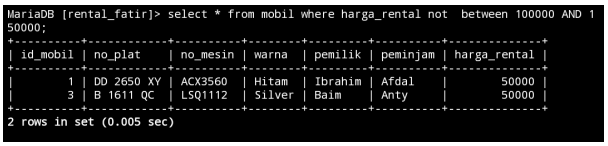
Struktur

```
SELECT * FROM nama_table WHERE nama_kolom NOT BETWEEN nilai_kolom AND
nilai_kolom ;
```

Contoh

```
select * from mobil WHERE harga_rental NOT BETWEEN 100000 AND 150000;
```

Hasil



Analisis

- `SELECT` : Bagian ini menentukan bahwa Anda ingin mengambil semua kolom dari tabel yang ditentukan.
- `FROM mobil`: Ini menunjukkan bahwa Anda menanyakan tabel bernama mobil.
- `WHERE harga_rental NOT BETWEEN 100000 AND 150000`: Ini adalah kondisi yang memfilter baris. Ini hanya memilih baris yang nilai kolomnya `harga_rental` tidak berada dalam kisaran 100.000 dan 150.000.

Kesimpulan

Kesimpulan `"SELECT * FROM mobil WHERE harga_rental NOT BETWEEN 100000 AND 150000;"` adalah bahwa perintah tersebut digunakan untuk mengambil semua data dari tabel "mobil" di mana harga_rental tidak berada di antara 100.000 dan 150.000.

<=

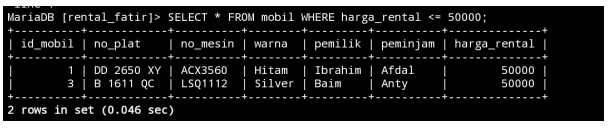
Struktur

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

Contoh

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

Hasil



Analisis

- `SELECT *`: Bagian kueri ini digunakan untuk menentukan kolom yang ingin Anda ambil. Tanda bintang `(*)` adalah karakter wildcard yang mewakili semua kolom dalam tabel "mobil".

- `FROM mobil`: Menentukan tabel tempat Anda ingin mengambil data, dalam hal ini, tabel "mobil".
- `WHERE harga_rental ≤ 50000`: Ini adalah kondisi yang memfilter baris berdasarkan kriteria tertentu. Dalam hal ini, ia hanya memilih baris yang nilai di kolom "harga_rental" kurang dari atau sama dengan 50000.

Kesimpulan

Kesimpulan `"SELECT * FROM mobil WHERE harga_rental ≤ 50000;"` adalah bahwa Anda mencari semua data dari tabel "mobil" di mana nilai kolom "harga_rental" kurang dari atau sama dengan 50000.

>=

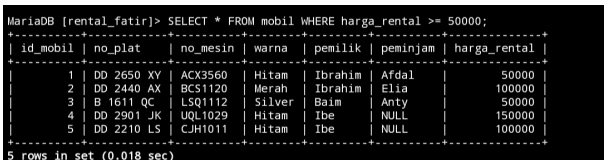
Struktur

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

Contoh

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

Hasil



```
MariaDB [rental_fatir]> SELECT * FROM mobil WHERE harga_rental >= 50000;
+----------+------------+----------+--------+---------+----------+-------------+
| id_mobil | no_plat    | no_mesin | warna  | pemilik | peminjam | harga_rental |
+----------+------------+----------+--------+---------+----------+-------------+
|        1 | DD 2650 XY | ACX3560  | 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 (0.018 sec)
```

Analisis

- `SELECT` digunakan untuk memilih kolom atau data yang ingin ditampilkan dalam hasil query.
- Tanda bintang `('*')` setelah kata kunci SELECT menunjukkan bahwa semua kolom dalam tabel "mobil" akan ditampilkan dalam hasil query.
- `FROM` digunakan untuk menentukan tabel yang akan digunakan dalam query. Dalam kasus ini, tabel yang digunakan adalah "mobil".
- `WHERE` digunakan untuk melakukan filter atau seleksi pada baris-baris data yang memenuhi kondisi tertentu.
- `Kondisi harga_rental ≥ 50000` menunjukkan bahwa hanya baris-baris data yang memiliki nilai harga_rental yang lebih besar atau sama dengan 50000 yang akan

ditampilkan.

Kesimpulan

Perintah SQL `SELECT * FROM mobil WHERE harga_rental ≥ 50000;` digunakan untuk mengambil semua data (semua kolom) dari tabel "mobil" di mana nilai pada kolom "harga_rental" lebih besar atau sama dengan 50000.

< > atau !=

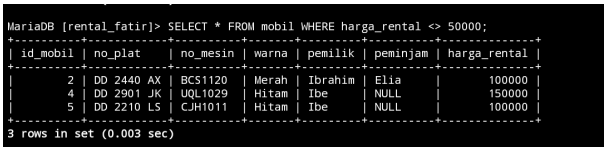
Struktur

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

Contoh

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

Hasil



```
MariaDB [rental_fatir]> SELECT * FROM mobil WHERE harga_rental <> 50000;
+----------+------------+----------+-------+---------+----------+--------------+
| id_mobil | 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 (0.003 sec)
```

Analisis

- `SELECT` : Ini menentukan bahwa Anda ingin memilih semua kolom dari tabel.
- `FROM mobil` : Ini menentukan nama tabel "mobil" dari mana Anda ingin mengambil datanya.
- `WHERE harga_rental <> 50000` : Ini adalah kondisi yang memfilter baris. Ini hanya memilih baris dimana nilai di kolom "harga_rental" tidak sama dengan 50000.

Kesimpulan

Kesimpulan dari `"SELECT * FROM mobil WHERE harga_rental <> 50000;"` adalah bahwa pernyataan tersebut akan mengembalikan semua baris dari tabel "mobil" di mana nilai kolom "harga_rental" tidak sama dengan 50000.

Tantangan

Struktur

```
SELECT [nama_kolom] FROM [nama_tabel]
WHERE [nama_kolom] = [nilai];
```

Contoh

```
SELECT nama FROM Akun WHERE nama = 'tir';
```

Hasil

```
MariaDB [rental_fatir]> select * from akun;
+----+---------+----------+----------+
| id | nama    | username | password |
+----+---------+----------+----------+
|  1 | tir     | admin    | 12345    |
|  2 | tetsuya | user     | 67890    |
|  3 | ftr     | new_user | 54321    |
+----+---------+----------+----------+
3 rows in set (0.001 sec)

MariaDB [rental_fatir]> SELECT nama FROM Akun WHERE nama = 'tir';
+------+
| nama |
+------+
| tir  |
+------+
1 row in set (0.001 sec)
```

Analisis

- `SELECT pemilik`: Bagian query ini menunjukkan bahwa Anda ingin mengambil nilai dari kolom bernama "pemilik" di tabel yang ditentukan.
- `FROM mobil`: Ini menentukan tabel tempat ingin mengambil data, dalam hal ini, tabel bernama "mobil."
- `WHERE no_plat="DD 2440 AX";`: Menunjukkan bahwa hanya ingin mengambil baris yang nilai di kolom "no_plat" sama dengan "DD 2440 AX."

Kesimpulan

Kesimpulan dari `SELECT` yang berikan yaitu mencari informasi pemilik mobil dengan nomor plat "DD 2440 AX" dari tabel mobil.

IN

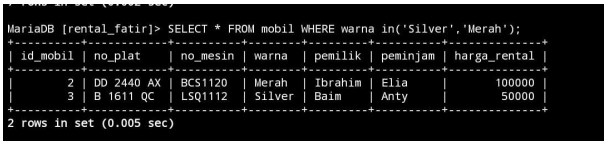
Struktur

```
select * from nama_tabel where kolom in('nilai1','nilai2');
```

Contoh

```
SELECT * FROM mobil WHERE warna in('Silver','Merah');
```

Hasil



```
MariaDB [rental_fatir]> SELECT * FROM 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 (0.005 sec)
```

Analisis

1. `SELECT * FROM mobil` : Pernyataan ini memilih semua kolom dari tabel "mobil".
2. `WHERE warna IN ('Silver', 'Merah')` : WHERE digunakan untuk menerapkan kondisi pada hasil query. kondisi yang diterapkan adalah "warna IN ('Silver', 'Merah')", yang berarti hanya baris dengan nilai kolom "warna" yang sama dengan 'Silver' atau 'Merah' yang akan diambil.

Kesimpulan

Query ini akan mengambil semua baris dari tabel "mobil" di mana nilai kolom "warna" adalah 'Silver' atau 'Merah'. query ini akan mengembalikan semua informasi tentang mobil-mobil yang memiliki warna 'Silver' atau 'Merah'. Hasilnya akan berupa kumpulan baris yang mewakili mobil-mobil dengan warna yang sesuai dengan kriteria tersebut.

IN + AND

Struktur

```
select * from nama_tabel
-> where nama_kolom in ('nilai1','nilai2')
-> AND nama_kolom = nilai3
```

Contoh

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

Hasil

```
MariaDB [rental_fatir]> select * from 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 | ACX3560  | Hitam  | Ibrahim | Afdal    |        50000 |
|        3 | B 1611 QC  | LSQ1112  | Silver | Baim    | Anty     |        50000 |
+----------+------------+----------+--------+---------+----------+--------------+
2 rows in set (0.006 sec)
```

Analisis

1. `SELECT * FROM mobil` : memilih semua kolom `(*)` dari tabel "mobil".
2. `WHERE warna in('Hitam','Silver') AND harga_rental = 50000` : Pernyataan WHERE digunakan untuk menerapkan kondisi pada hasil query. terdapat dua kondisi yang diterapkan :

- Kondisi pertama adalah `"warna IN ('Hitam', 'Silver')"`, yang berarti hanya baris dengan nilai kolom "warna" yang sama dengan 'Hitam' atau 'Silver' yang akan diambil.
- Kondisi kedua adalah `"harga_rental = 50000"` , yang berarti hanya baris dengan nilai kolom "harga_rental" yang sama dengan 50000 yang akan diambil.

Kesimpulan

Kesimpulan dari SQL tersebut adalah mencari semua entri (baris) dalam tabel `"mobil"` di mana nilai kolom `"warna" adalah 'Hitam' atau 'Silver', dan nilai kolom "harga_rental"Silver' 50000.` query tersebut akan mengembalikan semua data mobil yang memiliki warna `'Hitam' atau 'Silver'` dan `memiliki harga rental sebesar 50000.`

IN + OR

Struktur

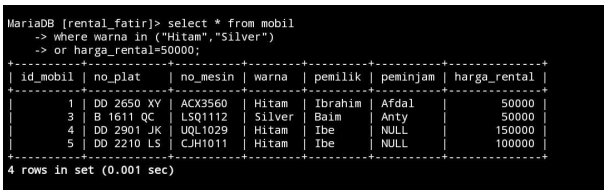
```
select * from nama_tabel
-> where nama_kolom in('nilai1','nilai2')
-> OR nama_kolom = nilai3
```

Contoh

```
SELECT * FROM mobil
-> WHERE warna in('Hitam','Silver')
```

```
  -> OR harga_rental = 50000;
```

Hasil



```
MariaDB [rental_fatir]> select * from 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 | ACX3560  | 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 |
+----------+------------+----------+--------+---------+----------+--------------+
4 rows in set (0.001 sec)
```

Analisis

- `SELECT *` : Ini memilih semua kolom dari tabel.
- `FROM mobil` : Ini menentukan nama tabel sebagai "mobil" dari mana data akan diambil.
- `WHERE warna IN ('Hitam', 'Silver')` : Kondisi ini memfilter baris berdasarkan nilai pada kolom "warna". Ini memilih baris di mana kolom "warna" memiliki nilai 'Hitam' atau 'Silver'.
- `OR harga_rental = 50000` : Kondisi ini memfilter lebih lanjut baris-baris tersebut dengan memilih baris-baris yang kolom "harga_rental"-nya bernilai 50000.

Kesimpulan

Kesimpulan SQL adalah untuk mengambil semua data dari tabel "mobil" dimana warna mobil adalah 'Hitam' atau 'Silver', atau harga rental mobil adalah 50000. menggabungkan kondisi OR antara warna mobil dan harga rental, hasilnya akan mencakup semua mobil yang memiliki warna 'Hitam' atau 'Silver', dan juga mobil dengan harga rental sebesar 50000, termasuk mobil yang memenuhi salah satu atau kedua kondisi tersebut.

IN + AND + OPERATOR

Struktur

```
select * from nama_tabel
-> where nama_kolom in('nilai1','nilai2')
-> AND nama_kolom > nilai3
```

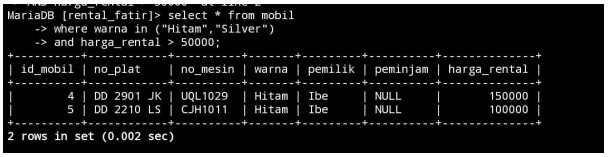
```
select * FROM nama_tabel
-> where nama_kolom in('nilai1','nilai2')
-> AND nama_kolom < nilai3
```

Contoh

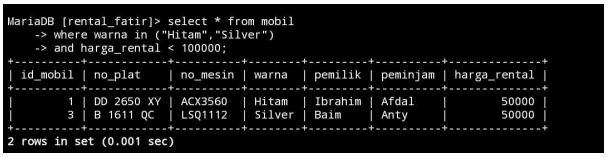
```
 SELECT * FROM mobil
-> WHERE warna in('Hitam','Silver')
-> AND harga_rental > 50000;
```

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

Hasil



```
MariaDB [rental_fatir]> select * from 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 |
+-----------+------------+-----------+-------+---------+----------+--------------+
2 rows in set (0.002 sec)
```



```
MariaDB [rental_fatir]> select * from 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 | ACX3560   | Hitam  | Ibrahim | Afdal    |        50000 |
|         3 | B 1611 QC  | LSQ1112   | Silver | Baim    | Anty     |        50000 |
+-----------+------------+-----------+--------+---------+----------+--------------+
2 rows in set (0.001 sec)
```

Analisis

1. `SELECT * FROM mobil` : Ini akan mengambil semua kolom dari tabel `mobil`.
2. `WHERE warna in('Hitam','Silver')` : Ini akan memfilter data hanya untuk mobil dengan warna hitam atau perak.
3. `AND harga_rental > 50000` : Ini akan memfilter data hanya untuk mobil dengan harga sewa lebih dari 50.000.

Kesimpulan

Query ini akan mengambil data dari tabel `mobil` yang memiliki warna hitam atau perak dan harga sewa lebih dari 50.000. Hasil dari query ini akan menampilkan semua kolom dari tabel `mobil` yang memenuhi kriteria di atas.

LIKE

Mencari awalan

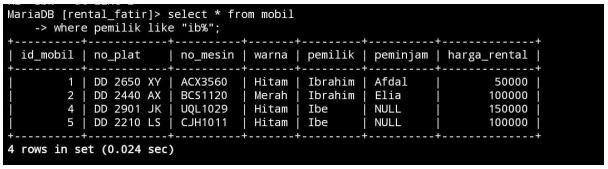
Struktur

```
SELECT * FROM mobil
-> WHERE pemilik like 'Ib%';
```

Contoh

```
SELECT * FROM mobil
-> WHERE pemilik LIKE 'Ib%';
```

Hasil



```
MariaDB [rental_fatir]> select * from mobil
    -> where pemilik like "ib%";
+----------+------------+----------+-------+---------+----------+--------------+
| id_mobil | no_plat    | no_mesin | warna | pemilik | peminjam | harga_rental |
+----------+------------+----------+-------+---------+----------+--------------+
|        1 | DD 2650 XY | ACX3560  | Hitam | Ibrahim | Afdal    |        50000 |
|        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 |
+----------+------------+----------+-------+---------+----------+--------------+
4 rows in set (0.024 sec)
```

Analisis

1. `SELECT * FROM mobil` : Ini akan mengambil semua kolom dari tabel `mobil`.
2. `WHERE pemilik LIKE 'Ib%'` : Ini akan memfilter data hanya untuk mobil yang dimiliki oleh pemilik yang nama awalnya adalah 'Ib'. Tanda persen `'%'` digunakan sebagai wildcard untuk mencari nama pemilik yang dimulai dengan 'Ib' dan mengikuti dengan karakter apapun.

Kesimpulan

Data dari tabel `mobil` yang dimiliki oleh pemilik dengan nama awal 'Ib'. Hasil dari query ini akan menampilkan semua kolom dari tabel `mobil`

Mencari akhiran

Struktur

```
SELECT * FROM mobil
-> WHERE pemilik LIKE '%m';
```

Contoh

```
SELECT * FROM mobil
-> WHERE pemilik LIKE '%m';
```

Hasil

```
MariaDB [rental_fatir]> select * from mobil
    -> where pemilik like "%m";
+----------+------------+----------+--------+---------+----------+--------------+
| id_mobil | no_plat    | no_mesin | warna  | pemilik | peminjam | harga_rental |
+----------+------------+----------+--------+---------+----------+--------------+
|        1 | DD 2650 XY | ACX3560  | 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 (0.002 sec)
```

Analisis

1. `SELECT * FROM mobil` : Ini akan mengambil semua kolom dari tabel `mobil`.
2. `WHERE pemilik LIKE '%m'` : Ini akan memfilter data hanya untuk mobil yang dimiliki oleh pemilik yang nama terakhirnya adalah 'm' atau berisi huruf 'm'. Tanda persen `'%'` digunakan sebagai wildcard untuk mencari nama pemilik yang mengakhiri dengan 'm' atau berisi huruf 'm' di mana saja.

Kesimpulan

Query ini akan mengambil data dari tabel `mobil` yang dimiliki oleh pemilik dengan nama terakhir 'm' atau berisi huruf 'm'. Hasil dari query ini akan menampilkan semua kolom dari tabel `mobil` yang memenuhi

Mencari awalan & akhiran

Struktur

```
SELECT * FROM mobil
→ WHERE pemilik LIKE 'b%m';
```

Contoh

```
SELECT * FROM mobil
→ WHERE pemilik LIKE 'b%m';
```

Hasil

```
MariaDB [rental_fatir]> select * from 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 (0.002 sec)
```

Analisis

1. `SELECT * FROM mobil` : Ini akan mengambil semua kolom dari tabel `mobil`.
2. `WHERE pemilik LIKE 'b%m'` : Ini akan memfilter data hanya untuk mobil yang dimiliki oleh pemilik yang nama awalnya dimulai dengan huruf 'b' dan nama terakhirnya berakhir

dengan huruf 'm'. Tanda persen `'%'` digunakan sebagai wildcard untuk mencari nama pemilik yang dimulai dengan huruf 'b' dan mengikuti dengan karakter apapun, lalu diakhiri dengan huruf 'm'.

Kesimpulan

tabel `mobil` yang dimiliki oleh pemilik dengan nama awal dimulai dengan huruf 'b' dan nama terakhir berakhir dengan huruf 'm'. Hasil dari query ini akan menampilkan semua kolom dari tabel `mobil`

Mencari total karakter

Struktur

```
SELECT * FROM mobil
→ WHERE pemilik LIKE 'I__';
```

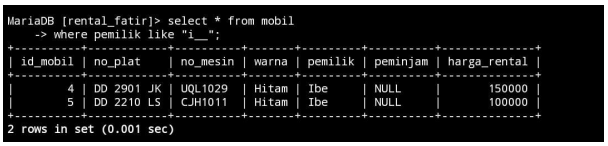
```
 SELECT * FROM mobil
→ WHERE pemilik LIKE '___';
```

Contoh

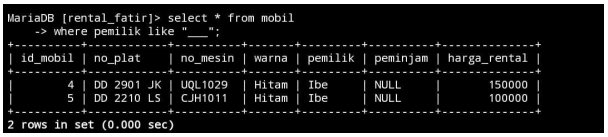
```
SELECT * FROM mobil
→ WHERE pemilik LIKE 'I__';
```

```
 SELECT * FROM mobil
→ WHERE pemilik LIKE '___';
```

Hasil



```
MariaDB [rental_fatir]> select * from 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 |
+----------+------------+----------+------+--------+----------+-------------+
2 rows in set (0.001 sec)
```



```
MariaDB [rental_fatir]> select * from mobil
    -> where pemilik like "___";
+----------+------------+----------+------+--------+----------+-------------+
| 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 |
+----------+------------+----------+------+--------+----------+-------------+
2 rows in set (0.000 sec)
```

Analisis

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

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 `'I__'` digunakan untuk memfilter baris-baris di mana nilai kolom "pemilik" terdiri dari 3 karakter, di mana karakter pertama adalah "I" (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.

Kombinasi

Struktur

```
SELECT * FROM mobil
→ WHERE pemilik LIKE '__r%';
```

```
SELECT * FROM mobil
→ WHERE pemilik LIKE '_b%';
```

Contoh

```
SELECT * FROM mobil
→ WHERE pemilik LIKE '__r%';
```

```
SELECT * FROM mobil
→ WHERE pemilik LIKE '_b%';
```

Hasil

```
MariaDB [rental_fatir]> select * from mobil
    -> where pemilik like "__r%";
+----------+------------+----------+-------+---------+----------+--------------+
| id_mobil | no_plat    | no_mesin | warna | pemilik | peminjam | harga_rental |
+----------+------------+----------+-------+---------+----------+--------------+
|        1 | DD 2650 XY | ACX3560  | Hitam | Ibrahim | Afdal    |        50000 |
|        2 | DD 2440 AX | BCS1120  | Merah | Ibrahim | Elia     |       100000 |
+----------+------------+----------+-------+---------+----------+--------------+
2 rows in set (0.008 sec)

MariaDB [rental_fatir]> select * from mobil
    -> where pemilik like "_b%";
+----------+------------+----------+-------+---------+----------+--------------+
| id_mobil | no_plat    | no_mesin | warna | pemilik | peminjam | harga_rental |
+----------+------------+----------+-------+---------+----------+--------------+
|        1 | DD 2650 XY | ACX3560  | Hitam | Ibrahim | Afdal    |        50000 |
|        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 |
+----------+------------+----------+-------+---------+----------+--------------+
4 rows in set (0.000 sec)
```

Analisis

- Mencari semua data dari tabel `mobil` (`SELECT *`).
- Di mana nilai pada kolom `pemilik` (`WHERE pemilik LIKE '__r%'`):
- **LIKE**: Operator untuk mencari pola dalam string.
- `__r%:` Pola pencarian yang digunakan.
- `__` : Mencocoki dua karakter apapun (wildcard).
- `r` : Mencari huruf "r" secara literal.
- `%` : Mencocoki nol karakter atau lebih karakter apapun.
- Mencari semua data dari tabel `mobil` (`SELECT *`).
- Di mana nilai pada kolom `pemilik` (`WHERE pemilik LIKE '_b%'`):
- **LIKE**: Operator untuk mencari pola dalam string.
- **'_b%'**: Pola pencarian yang digunakan.
- `_` : Mencocoki satu karakter apapun.
- `b` : Mencari huruf "b" secara literal.
- `%` : Mencocoki nol karakter atau lebih karakter apapun.

Kesimpulan

- `LIKE '__r%'` : Mencari data pada tabel `pemilik` dengan nama minimal 3 karakter di awalnya lalu karakter keempat adalah huruf `i`
- `LIKE '_b%' :` Mencari pemilik yang namanya memiliki huruf "b" di **posisi kedua**. Ini akan mengembalikan pemilik dengan nama minimal 3 karakter.

NOT LIKE

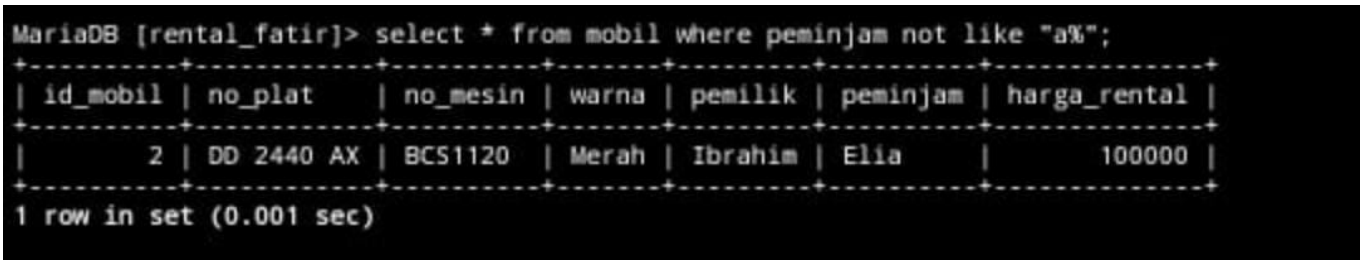
Struktur

```
SELECT * FROM mobil WHERE peminjam NOT LIKE 'A%';
```

Contoh

```
SELECT * FROM mobil WHERE peminjam NOT LIKE 'A%';
```

Hasil



```
MariaDB [rental_fatir]> select * from 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)
```

Analisis

1. `SELECT * FROM mobil` : Ini akan mengambil semua kolom dari tabel `mobil`.
2. `WHERE peminjam NOT LIKE 'A%'` : Ini akan memfilter data hanya untuk mobil yang saat ini tidak dipinjam oleh peminjam yang nama awalnya dimulai dengan huruf 'A'. Tanda persen `'%'` digunakan sebagai wildcard untuk mencari nama peminjam yang dimulai dengan huruf 'A' dan mengikuti dengan karakter apapun. Tanda negasi 'NOT' digunakan untuk mengecualikan peminjam dengan nama awal yang dimulai dengan huruf 'A'.

Kesimpulan

Kesimpulan `"SELECT * FROM mobil WHERE peminjam NOT LIKE 'A%'"` adalah bahwa query tersebut akan mengembalikan semua data dari tabel "mobil" dimana nilai kolom "peminjam" tidak dimulai dengan huruf 'A'.

Null & Not Null

Mencari data kosong

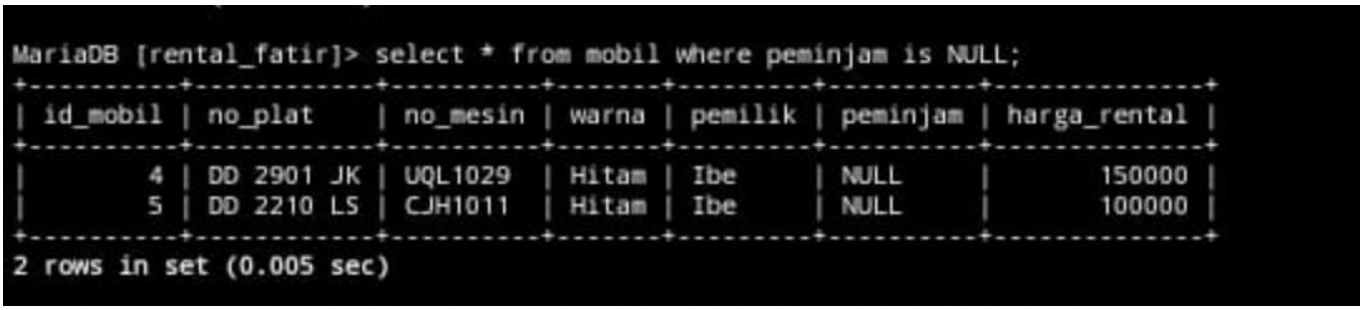
Struktur

```
select * from nama_table where nama_kolom is NULL;
```

Contoh

```
select * from mobil where peminjam is NULL;
```

Hasil



```
MariaDB [rental_fatir]> select * from 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 |
+-----------+------------+----------+-------+---------+----------+--------------+
2 rows in set (0.005 sec)
```

Analisis

- `SELECT :` Menginstruksikan untuk memilih semua kolom (atribut) dari tabel `mobil`.
- `FROM mobil:` Mendefinisikan tabel tempat data akan diambil, yaitu tabel `mobil`.
- `WHERE peminjam IS NULL:` Ini adalah klausa `WHERE` yang memfilter data berdasarkan kondisi tertentu.
- `peminjam:` Merujuk pada kolom pada tabel `mobil` yang kemungkinan berisi nama peminjam kendaraan.
- `IS NULL:` Operator perbandingan yang mengecek apakah nilai pada kolom `peminjam` adalah `NULL`.

Kesimpulan

- Perintah ini hanya mengembalikan mobil dengan kolom `peminjam` bernilai **NULL** (tidak ada data peminjam).
- Perintah ini membantu dalam mengidentifikasi mobil yang tersedia untuk dipinjam.
- Hasilnya bergantung pada data yang ada di kolom `peminjam` pada tabel `mobil`.

Mencari data yang tidak kosong

Struktur

```
select * from nama_table where nama_kolom is NOT NULL;
```

Contoh

```sql
select * from mobil where peminjam is NOT NULL;
```

Hasil

```
MariaDB [rental_fatir]> select * from mobil where peminjam is NOT NULL;
+-----------+------------+-----------+--------+---------+----------+--------------+
| id_mobil  | no_plat    | no_mesin  | warna  | pemilik | peminjam | harga_rental |
+-----------+------------+-----------+--------+---------+----------+--------------+
|         1 | DD 2650 XY | ACX3560   | 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)
```

Analisis

- `SELECT *`: Ini adalah klausa SELECT yang digunakan untuk memilih semua kolom dari tabel "mobil". Dengan menggunakan tanda bintang (*), kita memilih semua kolom yang ada dalam tabel.
- `FROM mobil`: Ini adalah klausa FROM yang menentukan tabel yang akan digunakan dalam kueri. Dalam hal ini, tabel yang digunakan adalah "mobil".
- `WHERE peminjam IS NOT NULL`: Ini adalah klausa WHERE yang digunakan untuk menerapkan kondisi pada kueri. Kondisi yang diterapkan di sini adalah "peminjam IS NOT NULL", yang berarti hanya baris-baris di mana kolom "peminjam" tidak kosong atau memiliki nilai yang tidak NULL akan dipilih.

Kesimpulan

Digunakan untuk mengambil semua baris dari tabel `"mobil"` di mana kolom `"peminjam"` memiliki nilai yang tidak NULL. Hasilnya akan berisi semua kolom dari baris-baris ini.

Order By

Mengurutkan data dari data terkecil

Struktur

```sql
select * from nama_table ORDER BY nama_kolom ASC;
```

Contoh

```sql
select * from mobil ORDER BY pemilik ASC;
```

Hasil

```
MariaDB [rental_fatir]> select * from mobil ORDER BY pemilik ASC;
+----------+------------+----------+--------+---------+----------+--------------+
| id_mobil | no_plat    | no_mesin | warna  | pemilik | peminjam | harga_rental |
+----------+------------+----------+--------+---------+----------+--------------+
|        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 | ACX3560  | Hitam  | Ibrahim | Afdal    |        50000 |
|        2 | DD 2440 AX | BCS1120  | Merah  | Ibrahim | Elia     |       100000 |
+----------+------------+----------+--------+---------+----------+--------------+
5 rows in set (0.003 sec)
```

Analisis

- `SELECT *`: Ini adalah klausa SELECT yang digunakan untuk memilih semua kolom dari tabel "mobil". Dengan menggunakan tanda bintang (*), kita memilih semua kolom yang ada dalam tabel.
- `FROM mobil`: Ini adalah klausa FROM yang menentukan tabel yang akan digunakan dalam kueri. Dalam hal ini, tabel yang digunakan adalah "mobil".
- `ORDER BY pemilik ASC`: Ini adalah klausa ORDER BY yang digunakan untuk mengurutkan hasil berdasarkan kolom "pemilik" secara menaik (ascending). Dengan menggunakan ASC, hasil akan diurutkan dari nilai terkecil hingga terbesar berdasarkan kolom "pemilik".

Kesimpulan

- `SELECT * FROM mobil` mengambil semua data dari tabel `mobil`.
- `ORDER BY pemilik ASC` mengurutkan data berdasarkan kolom `pemilik` secara ascending (terkecil ke besar).

Mengurutkan data dari data terbesar

Struktur

```
select * from nama_table ORDER BY nama_kolom desc;
```

Contoh

```
select * from mobil ORDER BY peminjam  desc;
```

Hasil

```
MariaDB [rental_fatir]> select * from mobil ORDER BY peminjam  desc;
+----------+------------+----------+--------+---------+----------+--------------+
| 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 |
|        1 | DD 2650 XY | ACX3560  | Hitam  | Ibrahim | Afdal    |        50000 |
|        4 | DD 2901 JK | UQL1029  | Hitam  | Ibe     | NULL     |       150000 |
|        5 | DD 2210 LS | CJH1011  | Hitam  | Ibe     | NULL     |       100000 |
+----------+------------+----------+--------+---------+----------+--------------+
5 rows in set (0.002 sec)
```

Analisis

- `SELECT` : Perintah ini digunakan untuk mengambil semua kolom atau field dari tabel mobil.
- `FROM mobil:` Ini menunjukkan bahwa tabel yang akan digunakan adalah `mobil`.
- `ORDER BY peminjam DESC:` Ini adalah bagian penting dari perintah yang akuntansi database untuk mengurutkan hasil berdasarkan kolom "peminjam" secara menurun (urutan menurun). Artinya, hasilnya akan diurutkan dari nilai yang paling tinggi ke nilai yang paling rendah pada kolom "peminjam".

Kesimpulan

- Pernyataan ini digunakan untuk mengambil seluruh data dari tabel "mobil".
- Data akan diurutkan berdasarkan kolom "peminjam" secara menurun (descending order). Artinya, data akan disusun dari nilai peminjam terbesar ke terkecil.
- Kesimpulannya, hasil query akan mengembalikan semua baris dari tabel "mobil" dengan urutan berdasarkan kolom "peminjam" dari nilai terbesar ke terkecil.

Membatasi data yang tampil

Struktur Query

```
SELECT * FROM [nama_tabel]    WHERE [nama_kolom] = [nilai] ORDER BY
[nama_kolom] ASC LIMIT [nilai];
```

Contoh

```
SELECT * FROM mobil
WHERE warna = 'Hitam' ORDER BY
```

Hasil

```
MariaDB [rental_fatir]> SELECT * FROM mobil WHERE warna = 'Hitam' ORDER BY harga_rental ASC LIMIT 2;
+----------+------------+----------+-------+---------+----------+------------+--------------+
| id_mobil | no_plat    | no_mesin | warna | pemilik | peminjam | deadline   | harga_rental |
+----------+------------+----------+-------+---------+----------+------------+--------------+
|        1 | DD 2650 XY | ACX3560  | Hitam | Ibrahim | Afdal    | 2024-04-24 |        50000 |
|        5 | DD 2210 LS | CJH1011  | Hitam | Ibe     | NULL     | NULL       |       100000 |
+----------+------------+----------+-------+---------+----------+------------+--------------+
2 rows in set (0.001 sec)
```

Analisis

- `SELECT * FROM :` Merupakan query awal untuk menampilkan sebuah tabel yang didalamnya berisikan kolom, di mana kolom tersebut menampung sebuah data.
- `mobil` nama dari tabel yang ingin kita tampilkan data-data nya.
- `WHERE warna = 'Hitam'` Memilih hanya baris-baris di mana nilai kolom warna adalah `'Hitam'.`
- `ORDER BY harga_rental ASC` Mengurutkan baris-baris tersebut berdasarkan kolom harga_rental secara menaik (dari harga rental terendah ke tertinggi).
- `LIMIT 2` Membatasi hasil query untuk hanya menampilkan 2 baris pertama yang memenuhi kondisi warna hitam dan diurutkan berdasarkan harga rental.

Kesimpulan

Perintah SQL tersebut mengambil data `mobil` berwarna `hitam` dengan `harga_rental` terendah, hanya menampilkan 2 hasil teratas.

DISTINCT

DISTINCT

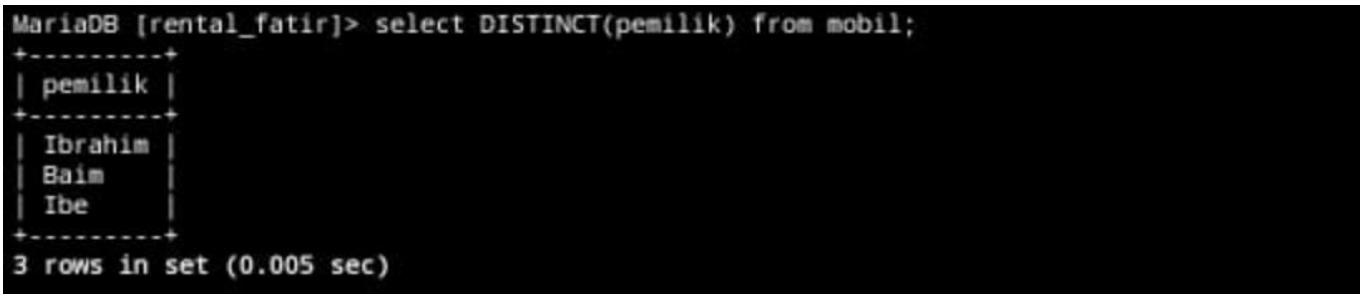
Struktur

```
select DISTINCT(nama_kolom) from nama_table;
```

Contoh

```
select DISTINCT(pemilik) from mobil;
```

Hasil



Analisis

- `SELECT DISTINCT (harga_rental)` : Ini adalah klausul SELECT yang digunakan untuk memilih kolom "harga_rental". Kata kunci "DISTINCT" digunakan untuk mengambil nilai unik dari kolom tersebut.
- `FROM mobil` : Ini adalah klausul FROM yang menentukan tabel yang digunakan dalam query, yaitu "mobil".
- `ORDER BY harga_rental DESC` : Ini adalah klausul ORDER BY yang digunakan untuk mengurutkan hasil berdasarkan kolom "harga_rental" secara menurun (descending order). Dengan kata lain, hasilnya akan ditampilkan dari harga_rental yang tertinggi ke terendah.

Kesimpulan

Kesimpulannya, hasil query akan mengembalikan nilai unik dari kolom "harga_rental" dari tabel "mobil", diurutkan dari nilai terbesar ke terkecil. Ini akan memberikan daftar harga rental yang berbeda yang tersedia untuk mobil, tanpa duplikasi, dalam urutan terurut.

DISTINCT+ORDER BY

Struktur

```
select DISTINCT(nama_kolom) from nama_table ORDER BY nama_kolom desc;
```

Contoh

```
select DISTINCT(harga_rental) from mobil ORDER BY harga_rental desc;
```

Hasil

```
MariaDB [rental_fatir]> select DISTINCT(harga_rental) from mobil ORDER BY harga_rental d
esc;
+--------------+
| harga_rental |
+--------------+
|       150000 |
|       100000 |
|        50000 |
+--------------+
3 rows in set (0.001 sec)
```

Analisis

- `SELECT DISTINCT(harga_rental)`: Bagian ini meminta data yang unik dari kolom `harga_rental` dalam tabel `mobil`. Artinya, jika ada nilai yang duplikat, hanya satu nilai yang akan diambil untuk setiap set nilai yang sama. Ini digunakan untuk memastikan bahwa hasil query ini hanya menampilkan nilai-nilai unik.
- `FROM mobil`: Bagian ini menentukan tabel dari mana data diambil. Dalam hal ini, data diambil dari tabel bernama `mobil`.
- `ORDER BY harga_rental DESC`: Bagian ini mengatur urutan hasil query berdasarkan kolom harga_rental dalam urutan menurun (dari yang terbesar ke yang terkecil). Dengan kata lain, hasilnya akan diurutkan sehingga nilai terbesar akan berada di bagian atas.

1. Mengidentifikasi Harga Rental Tertinggi:

- Dengan mengurutkan dalam urutan menurun, kita dapat dengan cepat mengidentifikasi harga rental tertinggi di dalam dataset.

2. Mengidentifikasi Rentang Harga Rental:

- Dengan melihat daftar harga yang unik, kita bisa mengetahui variasi harga rental dan rentang dari tertinggi ke terendah.

3. Analisis Harga Rental Unik:

- Karena menggunakan DISTINCT, query ini bisa membantu menemukan harga rental yang unik tanpa pengulangan, memberikan gambaran tentang berapa banyak variasi harga yang tersedia dalam tabel mobil.

Kesimpulan

query ini berguna untuk analisis harga rental yang unik dan untuk memahami rentang harga dalam tabel `mobil`, yang dapat digunakan untuk berbagai tujuan, seperti perbandingan harga, analisis pasar, atau pengambilan keputusan bisnis.

CONCAT,CONCAT_WA,AS

Menggabungkan kolom tanpa pemisah

Struktur

```
SELECT CONCAT(nama_kolom1,nama_kolom2) FROM nama_table;
```

Contoh

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

Hasil

Analisis

`SELECT DISTINCT(harga_rental)`
untuk mengambil nilai unik dari kolom `harga_rental` dalam `tabel mobil.` `DISTINCT` memastikan bahwa hanya satu dari setiap nilai yang sama ditampilkan dalam hasil. Ini berguna untuk menghindari duplikasi.

Kesimpulan

mengambil data dari `tabel mobil` dan menggabungkan nilai-nilai dari dua kolom (yaitu `pemilik` dan `Warna`), lalu mengembalikan hasilnya sebagai satu string untuk setiap baris dalam tabel `Mobil`.

Menggabungkan kolom dengan pemisah

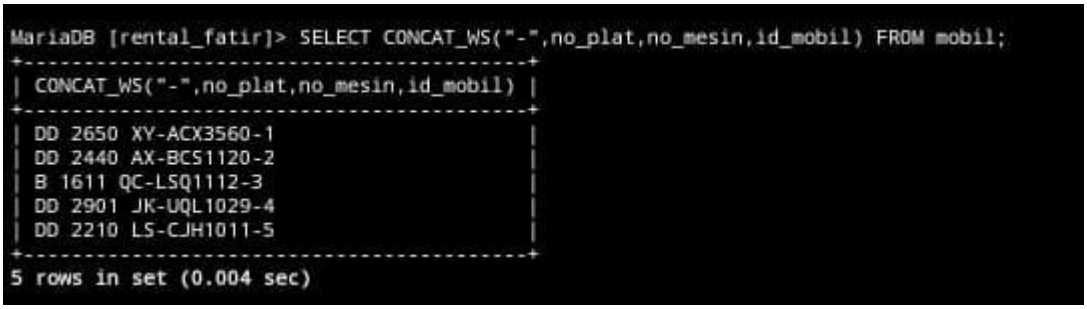
Struktur

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

Contoh

```
SELECT CONCAT_WS("-",nama_kolom1,nama_kolom2,nama_kolom3) FROM mobil;
```

Hasil



```
MariaDB [rental_fatir]> 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 1611 QC-LSQ1112-3                    |
| DD 2901 JK-UQL1029-4                   |
| DD 2210 LS-CJH1011-5                   |
+----------------------------------------+
5 rows in set (0.004 sec)
```

Analisis

- `DISTINCT` digunakan untuk memastikan bahwa hanya nilai unik yang diambil. Artinya, jika ada nilai `Harga_Rental` yang sama di beberapa baris, hanya satu yang akan dimasukkan dalam hasil.
- Ini berguna untuk menghilangkan duplikasi dan mendapatkan daftar harga sewa unik.
- `ORDER BY harga_rental desc` menunjukkan bahwa hasil akan diurutkan berdasarkan kolom `harga_rental` dalam urutan menurun (descending).
- Dengan demikian, hasilnya akan menampilkan nilai tertinggi terlebih dahulu, kemudian turun ke yang lebih rendah.
- Hasil akhir adalah daftar nilai unik dari `harga_rental` dalam tabel `mobil`, yang diurutkan dari nilai tertinggi ke terendah.
- Jika `Harga_Rental` memiliki berbagai nilai seperti 500, 400, 300, 500, dan 200, hasilnya akan menjadi 500, 400, 300, dan 200, dengan nilai 500 hanya muncul sekali karena penggunaan `DISTINCT`.

Kesimpulan

memberikan cara untuk mengambil daftar harga rental unik dari tabel `mobil`, dan mengurutkannya dalam urutan menurun, menampilkan nilai tertinggi di depan.

Memberi nama kolom alias

Struktur

```
SELECT CONCAT_WS("+",nama_kolom1,nama_kolom2) AS COLLAB FROM nama_table;
```

Contoh

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

Hasil

Analisis

- `CONCAT_WS` adalah fungsi yang menggabungkan beberapa string dan memasukkan delimiter (pemisah) di antara mereka.
- Pada pernyataan ini, simbol `+` digunakan sebagai delimiter untuk menggabungkan nilai dari kolom `Pemilik` dan `peminjam`.
- Setelah penggabungan, hasilnya diberi alias dengan nama `COLLAB`.
- Jadi, hasil dari pernyataan ini adalah daftar gabungan nilai dari kolom `pemilik` dan peminjam dengan simbol `+` di antaranya, lalu hasil ini diberi nama `COLLAB`.
- Hasil akan menunjukkan kombinasi unik dari `pemilik` dan `peminjam` dengan simbol `+` sebagai separator.
- Misalnya, jika `pemilik` adalah "John" dan `peminjam` adalah "Doe", maka hasilnya adalah "John+Doe".
- Jika ada baris dalam tabel `mobil` dengan kombinasi yang berbeda, seperti `pemilik` adalah "Jane" dan `peminjam` adalah "Smith", maka hasilnya akan menjadi "Jane+Smith".
- Pernyataan ini dapat digunakan untuk membuat representasi unik dari hubungan antara pemilik dan peminjam dalam tabel `mobil`.
- Ini berguna untuk mengidentifikasi kolaborasi atau interaksi antara dua entitas dalam tabel tersebut.
- Penggunaan `CONCAT_WS` memudahkan untuk memasukkan pemisah tanpa menambahkan spasi ekstra atau perlu memeriksa apakah ada nilai kosong.

Kesimpulan

memungkinkan Anda untuk menggabungkan dua kolom (`pemilik` dan `peminjam`) dari tabel `mobil` , dengan simbol `+` sebagai separator, dan memberikan nama alias " `COLLAB` " untuk hasil

gabungan tersebut. Hasil akhir adalah daftar kombinasi unik dengan format yang mudah dibaca dan digunakan.

VIEW

Membuat tabel virtual

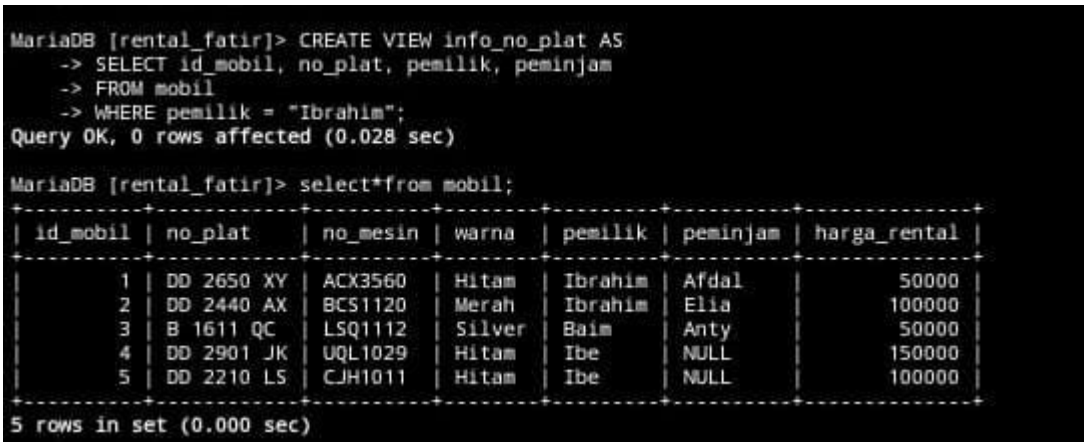
Struktur

```
CREATE VIEW nama_table AS
SELECT nama_kolom1, nama_kolom2, nama_kolom3, nama_kolom4
FROM nama_table
WHERE nama_kolom5= "Ibrahim";
```

Contoh

```
CREATE VIEW info_no_plat AS
SELECT id_mobil, no_plat, pemilik, peminjam
FROM mobil
WHERE pemilik = "Ibrahim";
```

Hasil



```
MariaDB [rental_fatir]> CREATE VIEW info_no_plat AS
    -> SELECT id_mobil, no_plat, pemilik, peminjam
    -> FROM mobil
    -> WHERE pemilik = "Ibrahim";
Query OK, 0 rows affected (0.028 sec)

MariaDB [rental_fatir]> select*from mobil;
+----------+------------+----------+--------+---------+----------+--------------+
| id_mobil | no_plat    | no_mesin | warna  | pemilik | peminjam | harga_rental |
+----------+------------+----------+--------+---------+----------+--------------+
|        1 | DD 2650 XY | ACX3560  | Hitam  | Ibrahim | Afdal    |        50000 |
|        2 | DD 2440 AX | BC51120  | Merah  | Ibrahim | Elia     |       100000 |
|        3 | B 1611 QC  | LSQ1112  | Silver | Baim    | Anty     |        50000 |
|        4 | DD 2901 JK | UQL1029  | Hitam  | Ibe     | NULL     |       150000 |
|        5 | DD 2210 LS | CJH1011  | Hitam  | Ibe     | NULL     |       100000 |
+----------+------------+----------+--------+---------+----------+--------------+
5 rows in set (0.000 sec)
```

Analisis

- View yang dibuat diberi nama `info_no_plat.
- Query memilih beberapa kolom dari tabel `mobil: id_mobil, no_plat, pemilik, dan peminjam.`
- Jadi, view ini akan berisi informasi tentang `ID mobil, nomor plat, pemilik, dan peminjam` dari tabel `mobil`.

- Kondisi `WHERE` dalam query memastikan bahwa hanya baris dengan pemilik bernama "Ibrahim" yang akan masuk ke view.
- Dengan kata lain, view ini hanya akan menampilkan data tentang mobil yang dimiliki oleh "Ibrahim".
- View `info_no_plat` mungkin dibuat untuk mendapatkan informasi tentang mobil-mobil yang dimiliki oleh "Ibrahim", termasuk ID mobil, nomor plat, pemilik, dan peminjam.
- Penggunaan view memudahkan untuk mengakses dan mengelola informasi spesifik tanpa harus menulis query ulang setiap kali ingin melihat data mobil milik "Ibrahim".
- View memungkinkan Anda untuk menyederhanakan akses ke data tertentu. Misalnya, ketika Anda perlu memeriksa mobil yang dimiliki oleh "Ibrahim", Anda hanya perlu mengakses view ini.
- Ini juga dapat digunakan untuk membuat antarmuka data yang lebih mudah dikelola atau memberikan akses yang terbatas kepada pengguna tertentu.

Kesimpulan

ini membuat view bernama `info_no_plat` yang menyaring data dari tabel `mobil` untuk hanya menampilkan mobil yang dimiliki oleh "Ibrahim", beserta beberapa kolom seperti `id_mobil, no_plat, pemilik, dan peminjam`. View ini memberikan cara yang mudah untuk melihat dan mengelola data tersebut.

Menampilkan tabel virtual

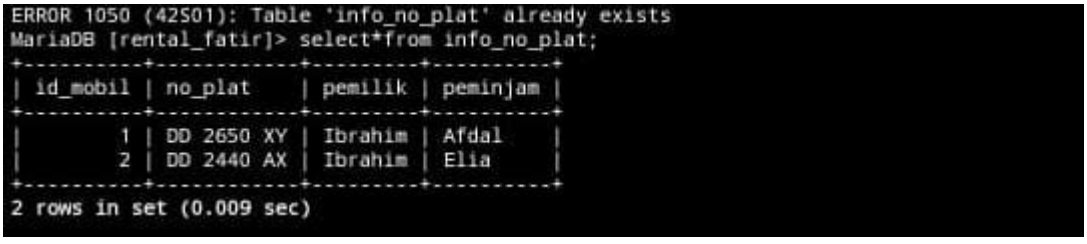
Struktur

```
SELECT * FROM nama_table;
```

Contoh

```
SELECT * FROM info_no_plat;
```

Hasil



Analisis

- `SELECT * FROM` : Merupakan query awal untuk menampilkan sebuah tabel virtual yang didalamnya berisikan kolom, di mana kolom tersebut menampung sebuah data.
- `info_no_plat` nama dari tabel virtual.

kesimpulan

Jika ingin menampilkan tabel virtual cukup dengan perintah SELECT * FROM.

Menghapus tabel virtual

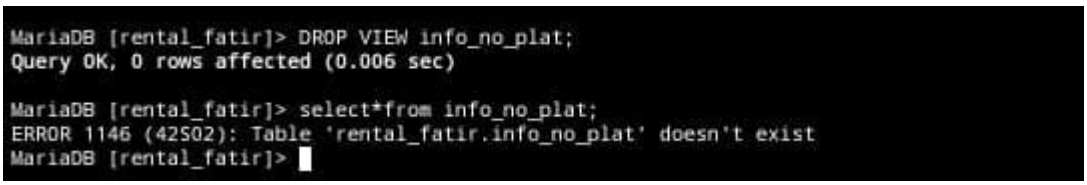
Struktur

```
DROP VIEW nama_table;
```

Contoh

```
DROP VIEW info_no_plat;
```

Hasil



Analisis

- `DROP VIEW` perintah query tersebut akan menghapus tabel virtual.
- `info_no_plat` nama tabel virtual yang akan di hapus.

kesimpulan

Jika ingin menghapus tabel virtual cukup dengan perintah DROP VIEW.

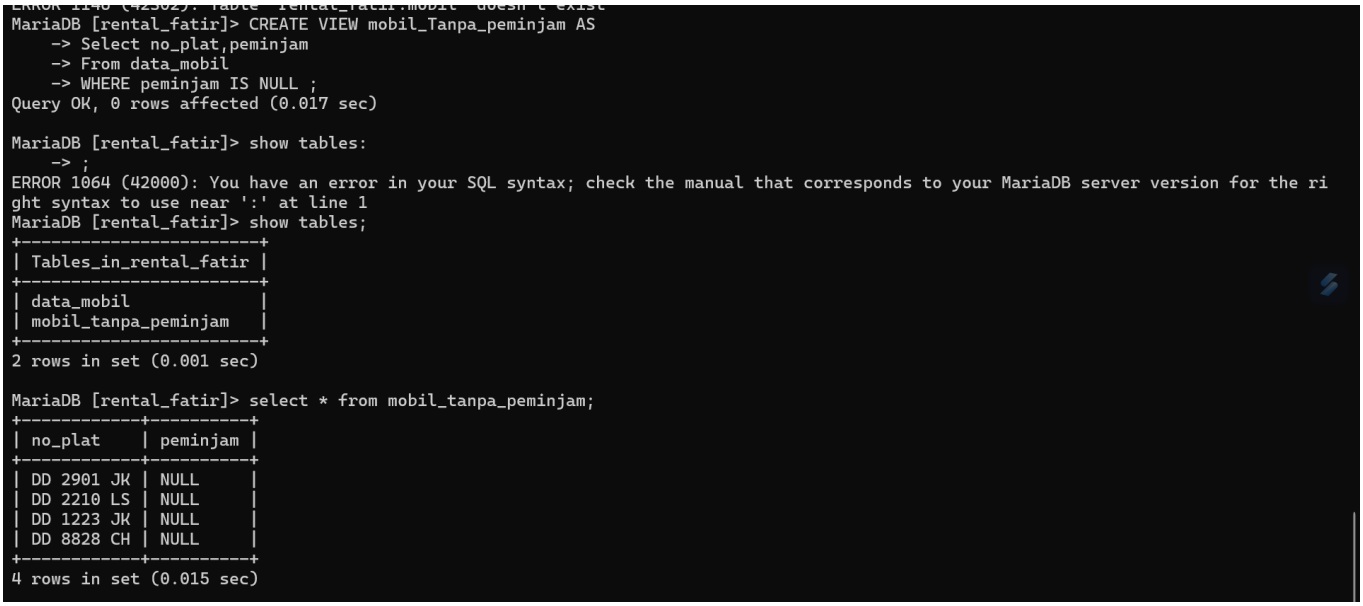
Tantangan

Mobil Tanpa Peminjam

Kode

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

Hasil



```
MariaDB [rental_fatir]> CREATE VIEW mobil_Tanpa_peminjam AS
    -> Select no_plat,peminjam
    -> From data_mobil
    -> WHERE peminjam IS NULL ;
Query OK, 0 rows affected (0.017 sec)

MariaDB [rental_fatir]> show tables:
    -> ;
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your MariaDB server version for the ri
ght syntax to use near ':' at line 1
MariaDB [rental_fatir]> show tables;
+------------------------+
| Tables_in_rental_fatir |
+------------------------+
| data_mobil             |
| mobil_tanpa_peminjam   |
+------------------------+
2 rows in set (0.001 sec)

MariaDB [rental_fatir]> select * from mobil_tanpa_peminjam;
+------------+----------+
| no_plat    | peminjam |
+------------+----------+
| DD 2901 JK | NULL     |
| DD 2210 LS | NULL     |
| DD 1223 JK | NULL     |
| DD 8828 CH | NULL     |
+------------+----------+
4 rows in set (0.015 sec)
```

Analisis

CREATE VIEW mobil_Tanpa_peminjam AS `: adalah perintah untuk membuat sebuah` view `baru dalam basis data dengan nama` mobil_Tanpa_peminjam`.

- `SELECT no_plat, peminjam`: 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 menyaring 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.

Kesimpulan

`CREATE VIEW mobil_Tanpa_peminjam AS Select no_plat, peminjam From mobil WHERE peminjam IS NULL;` digunakan untuk membuat sebuah `view` baru bernama `mobil_Tanpa_peminjam`. `Viewnya` berisi dua kolom, yaitu `no_plat` dan `peminjam`, 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`.

Update Mobil

Kode

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

Hasil

```
MariaDB [rental_fatir]> UPDATE data_mobil
    -> SET peminjam = NULL
    -> where peminjam =  'Elia' ;
Query OK, 1 row affected (0.013 sec)
Rows matched: 1  Changed: 1  Warnings: 0

MariaDB [rental_fatir]> select * from mobil_tanpa_peminjam;
+------------+----------+
| no_plat    | peminjam |
+------------+----------+
| DD 2440 AX | NULL     |
| DD 2901 JK | NULL     |
| DD 2210 LS | NULL     |
| DD 1223 JK | NULL     |
| DD 8828 CH | NULL     |
+------------+----------+
5 rows in set (0.001 sec)
```

Analisis

- `UPDATE mobil`: adalah perintah untuk memperbarui data dalam tabel yang disebut `mobil`.
- `SET peminjam= NULL`: menetapkan nilai kolom peminjam menjadi `NULL`.
- `WHERE peminjam= 'Elia'`: adalah klausa WHERE yang membatasi update hanya pada baris-baris dimana nilai kolom peminjam adalah `'Elia'`. Maksudnya perubahan hanya akan berlaku untuk baris-baris yang memiliki peminjam dengan nama `'Elia'`.
- `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.

Kesimpulan

`UPDATE mobil SET peminjam = NULL WHERE peminjam = 'Elia';` nilai pada kolom `peminjam` pada tabel `mobil` yang memiliki nilai `'Elia'` akan diubah menjadi `NULL`.
Kesimpulannya, perintah digunakan untuk menghapus atau mengubah nilai `peminjam` menjadi `NULL` untuk semua entri di tabel `mobil` dimana `peminjam` memiliki nilai `'Elia'`.

`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 `Elia`.

Kesimpulan

Berikan Kesimpulan mengapa tabel Virtual 1,2 ini dibuat?
View dapat digunakan untuk menyaring data yang sesuai dengan kriteria tertentu, seperti menampilkan entri yang memiliki nilai NULL pada kolom tertentu atau mengubah salah satu data peminjam menjadi NULL. memberikan pandangan yang jelas tentang mobil yang tersedia untuk disewakan atau yang belum dipinjam.
Dengan membuat view dapat membatasi akses ke data sensitif atau kolom tertentu dari tabel yang mungkin tidak perlu diakses oleh semua pengguna.
Dengan membuat view untuk kueri yang sering digunakan, Anda dapat menghindari pengulangan kode SQL yang sama di beberapa tempat dalam aplikasi atau prosedur penyimpanan.

Agreasi

Sum

Struktur

```
select SUM(nama_kolom) AS total
From nama_table
Where kondisi_opsional;
```

Contoh

```
select SUM(harga_rental) from mobil;
```

Hasil

```
MariaDB [rental_fatir]> select SUM(harga_rental) from mobil;
+-------------------+
| SUM(harga_rental) |
+-------------------+
|            450000 |
+-------------------+
1 row in set (0.004 sec)
```

Analisis

- `SELECT` : Digunakan untuk mengambil data dari database.
- `SUM():` Fungsi agregat dalam SQL yang digunakan untuk menghitung jumlah nilai dalam kolom.
- `harga_rental` : Ini adalah nama kolom yang ingin Anda hitung jumlahnya.
- `FROM mobil:` Menentukan tabel dari mana Anda ingin mengambil datanya, dalam hal ini, tabel bernama `mobil` .

Kesimpulan

Kesimpulan dari pernyataan `SQL SELECT ``SUM(harga_rental)`` FROM mobil;` adalah bahwa Anda ingin menghitung jumlah total `harga_rental` dari semua mobil yang ada dalam tabel mobil.

Count

Struktur

```
select Count(*) AS jumlah
From nama_table
Where kondisi_opsional;
```

Contoh

```
select Count(Peminjam) from mobil;
```

```
select Count(pemilik) from mobil;
```

Hasil

```
MariaDB [rental_fatir]> select count(pemilik) from mobil;
MariaDB [rental_fatir]> select count(pemilik) from mobil;
+----------------+
| count(pemilik) |
+----------------+
|              5 |
+----------------+
1 row in set (0.002 sec)
```

Analisis

pemilik

- `select :` Kata kunci ini digunakan untuk mengambil data dari database.
- `COUNT() :` Ini adalah fungsi agregat yang menghitung jumlah baris yang dikembalikan oleh kueri. Dalam hal ini, ia akan menghitung jumlah nilai bukan nol pada kolom `pemilik`.
- `pemilik` : Ini nama kolomnya. Fungsi ini `COUNT()` akan menghitung jumlah nilai bukan nol di kolom ini.
- `FROM mobil :` Ini menentukan tabel dari mana data akan diambil. Dalam hal ini, itu adalah tabel `mobil`. peminjam
- `SELECT` : Kata kunci ini digunakan untuk mengambil data dari database.
- `COUNT(peminjam)` : Fungsi ini menghitung jumlah nilai bukan nol pada kolom yang ditentukan, dalam hal ini, `peminjam`.
- `FROM mobil` : Ini menentukan tabel untuk mengambil data, dalam hal ini, tabel bernama `mobil`.

Kesimpulan

pemilik Kesimpulan dari `SELECT COUNT(pemilik) FROM mobil;` adalah bahwa Anda ingin menghitung jumlah entri unik dalam kolom `pemilik` dari tabel `mobil`.

peminjam Kesimpulan dari `SELECT COUNT(peminjam) FROM mobil;` adalah bahwa Anda ingin menghitung jumlah entri unik dalam kolom `peminjam` dari tabel `mobil`. Hasilnya akan memberikan jumlah `peminjam` `mobil` yang terdaftar dalam tabel.

Min

Struktur

```
select Min(nama_kolom) AS nilai_minimum
From nama_table
Where kondisi_opsional;
```

Contoh

```sql
select Min(harga_rental) AS minimal from mobil;
```

Hasil

```
MariaDB [rental_fatir]> select min(harga_rental)  As minimal from mobil;
+---------+
| minimal |
+---------+
|   50000 |
+---------+
1 row in set (0.001 sec)
```

Analisis

- `SELECT` digunakan untuk mengambil/menampilkan data.
- `MIN` digunakan untuk menampilkan nilai numerik terendah dalam suatu kolom.
- `harga_rental` nama kolom yang ingin kita tampilkan nilai terendahnya.
- `AS minima` sebagai nama alias kolom
- FROM mobil query tersebut akan mengambil data dari tabel mobil.

Kesimpulan

`MIN` digunakan untuk menampilkan nilai terendah dari suatu kolom.lan

Max

Struktur

```sql
select Max(nama_kolom) AS nilai_maksimal
From nama_table
Where kondisi_opsional;
```

Contoh

```sql
select Max(harga_rental) AS maximal from mobil;
```

Hasil

```
MariaDB [rental_fatir]> select max(harga_rental)  As maximal from mobil;
+---------+
| maximal |
+---------+
|  150000 |
+---------+
1 row in set (0.000 sec)
```

Analisis

- `SELECT` digunakan untuk mengambil/menampilkan data.
- `MAX` digunakan untuk menampilkan nilai numerik tertinggi dalam suatu kolom
- `harga_rental` nama kolom yang ingin kita tampilkan nilai tertinggi nya.
- `AS` maximun sebagai nama alias kolom
- `FROM` mobil query tersebut akan mengambil data dari tabel mobil.

Kesimpulan

`MAX` digunakan untuk menampilkan nilai numerik tertinggi dari suatu kolom.

Avg

Struktur

```
select avg(nama_kolom) AS nilai_rata-rata
From nama_table
Where kondisi_opsional;
```

Contoh

```
select AVG(harga_rental) AS RATA_RATA from mobil;
```

Hasil

```
MariaDB [rental_fatir]> select avg(harga_rental)  As RATA_RATA from mobil;
+------------+
| RATA_RATA  |
+------------+
| 90000.0000 |
+------------+
1 row in set (0.002 sec)
```

Analisis

- `SELECT` digunakan untuk mengambil/menampilkan data.
- `AVG` digunakan untuk menghitung rata-rata dari nilai numerik yang ada pada kolom
- `harga_rental` nama kolom yang ingin kita tampilkan nilai tertinggi nya.
- `AS rerata` sebagai nama alias kolom
- `FROM mobil` query tersebut akan mengambil data dari tabel mobil.

Kesimpulan

`AVG` digunakan untuk menghitung rata-rata dari nilai numerik pada suatu kolom.

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;
```

Query

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

Hasil

```
MariaDB [rental_fatir]> select warna,COUNT(id_mobil) AS Jumlah_Data_Mobil FROM data_mobil GROUP BY warna;
+--------+-------------------+
| warna  | Jumlah_Data_Mobil |
+--------+-------------------+
| Hitam  |                 4 |
| Merah  |                 1 |
| Silver |                 1 |
+--------+-------------------+
3 rows in set (0.001 sec)
```

Analisis

- `SELECT Klausa :`
  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 Klausa:`
  data_mobil: Menentukan tabel data_mobil sebagai sumber data.
- `GROUP BY Klausa:`
  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 data_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;
```

Query

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

Hasil

```
MariaDB [rental_fatir]> 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 (0.001 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 data_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 data_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;
```

Query

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

Hasil

```
MariaDB [rental_fatir]> SELECT pemilik,
    -> COUNT(id_mobil) AS jumlah_mobil
    -> FROM data_mobil GROUP BY pemilik;
+---------+--------------+
| pemilik | jumlah_mobil |
+---------+--------------+
| Baim    |            1 |
| Ibe     |            3 |
| Ibrahim |            2 |
| Valen   |            1 |
+---------+--------------+
4 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 data_mobil` menentukan tabel `data_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;
```

Query

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

Hasil

```
MariaDB [rental_fatir]> select pemilik,SUM(harga_rental) AS jumlah_pendapatan from data_mobil GROUP BY pemilik;
+---------+-------------------+
| pemilik | jumlah_pendapatan |
+---------+-------------------+
| Baim    |             50000 |
| Ibe     |            350000 |
| Ibrahim |            150000 |
| Valen   |            250000 |
+---------+-------------------+
4 rows in set (0.008 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 data_mobil** menentukan tabel `data_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

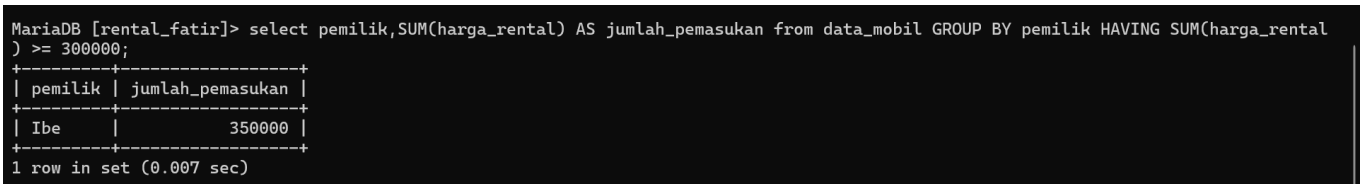
Struktur Query

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

Query

```
select pemilik,SUM(harga_rental) AS jumlah_pemasukan from data_mobil GROUP BY
pemilik HAVING SUM(harga_rental) ≥ 300000;
```

Hasil



```
MariaDB [rental_fatir]> select pemilik,SUM(harga_rental) AS jumlah_pemasukan from data_mobil GROUP BY pemilik HAVING SUM(harga_rental
) >= 300000;
+---------+-----------------+
| pemilik | jumlah_pemasukan |
+---------+-----------------+
| Ibe     |          350000 |
+---------+-----------------+
1 row in set (0.007 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_pemasukan** memberikan alias pada hasil perhitungan `SUM(harga_rental)` sehingga hasilnya akan diberi nama `jumlah_pemasukan`.
- **FROM data_mobil** menentukan tabel `data_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;
```

Query

```
select pemilik,AVG(harga_rental) AS rata_pemasukam from data_mobil GROUP BY
pemilik;
```

Hasil

```
MariaDB [rental_fatir]> select pemilik,AVG(harga_rental) AS rata_pemasukam from data_mobil GROUP BY pemilik;
+---------+----------------+
| pemilik | rata_pemasukam |
+---------+----------------+
| Baim    |     50000.0000 |
| Ibe     |    116666.6667 |
| Ibrahim |     75000.0000 |
| Valen   |    250000.0000 |
+---------+----------------+
4 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 `data_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 `data_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.

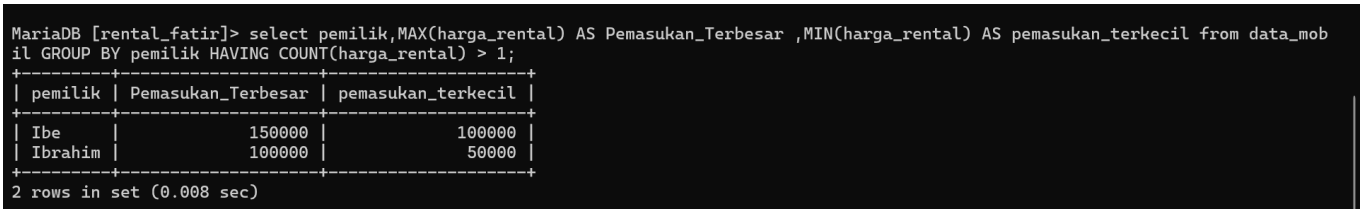
Struktur Query

```
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;
```

Query

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

Hasil



```
MariaDB [rental_fatir]> select pemilik,MAX(harga_rental) AS Pemasukan_Terbesar ,MIN(harga_rental) AS pemasukan_terkecil from data_mob
il GROUP BY pemilik HAVING COUNT(harga_rental) > 1;
+---------+--------------------+--------------------+
| pemilik | Pemasukan_Terbesar | pemasukan_terkecil |
+---------+--------------------+--------------------+
| Ibe     |             150000 |             100000 |
| Ibrahim |             100000 |              50000 |
+---------+--------------------+--------------------+
2 rows in set (0.008 sec)
```

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 data_mobil** menentukan tabel `data_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 tiga 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.