

| Operator Logika & Pembanding

| AND

| Struktur

```
SELECT warna, pemilik FROM mobil WHERE warna="Hitam" AND pemilik="Ibrahim"
```

| Contoh

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

| Hasil

```
MariaDB [rental_fina]> 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 (0.012 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 warna, pemilik FROM mobil WHERE warna="Hitam" OR pemilik="Ibrahim"
```

Contoh

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

Hasil

```
MariaDB [rental_fina]> 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 (0.039 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 mobil WHERE harga_rental BETWEEN 100000 AND 200000;
```

Contoh

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

Hasil

```
MariaDB [rental_fina]> 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 (0.001 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 klausal 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 mobil WHERE harga_rental NOT BETWEEN 100000 AND 200000;
```

Contoh

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

Hasil

```
MariaDB [rental_fina]> select * from mobil WHERE harga_rental NOT BETWEEN 100000 AND 150000;
```

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

- `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

```
MariaDB [rental_fina]> 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
3	B 1611 QC	LSQ1112	Silver	Baim	Anty	50000

2 rows in set (0.001 sec)

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_fina]> 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 (0.001 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_fina]> 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 (0.017 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 Login**

| **Struktur**

```
SELECT * FROM akun;
```

| **Contoh**

```
SELECT nama FROM akun WHERE password="12345";
```

Hasil

```
MariaDB [rental_mda]> SELECT * FROM akun;
+----+-----+-----+-----+
| id  | nama  | username | password |
+----+-----+-----+-----+
| 1   | Githa | admin    | 12345    |
| 2   | Rani  | user     | 67890    |
| 3   | Qolby | new_user | 54321    |
+----+-----+-----+-----+
3 rows in set (0.001 sec)
```

```
MariaDB [rental_mda]> SELECT nama FROM akun WHERE password="12345";
+-----+
| nama  |
+-----+
| Githa |
+-----+
1 row in set (0.002 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="B 1611 QC"**: Menunjukkan bahwa hanya ingin mengambil baris yang nilai di kolom "no_plat" sama dengan "B 1611 QC."

Kesimpulan

Kesimpulan dari **SELECT** yang berikan yaitu mencari informasi pemilik mobil dengan nomor plat "B 1611 QC" 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_fina]> 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 (0.016 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_fina]> 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 (0.042 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" adalah 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_fina]> 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_fina]> 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 (0.039 sec)
```

```
MariaDB [rental_fina]> SELECT * FROM mobil  
-> WHERE warna in('Hitam','Silver')  
-> AND harga_rental < 50000;  
Empty 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 nama_tabel
-> WHERE pemilik like 'Ib%';
```

Contoh

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

Hasil

```
MariaDB [rental_fina]> 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 (0.039 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 nama_tabel
-> WHERE nama_colom LIKE '%m';
```

Contoh

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

Hasil

```
MariaDB [rental_fina]> 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 (0.000 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 nama_tabel
-> WHERE nama_colom LIKE 'b%m';
```

Contoh

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

Hasil

```
MariaDB [rental_fina]> 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 (0.001 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 nama_tabel
-> WHERE nama_colom LIKE 'I__';
```

```
SELECT * FROM nama_tabel
-> WHERE nama_colom LIKE '____';
```

Contoh

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

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

Hasil

```
MariaDB [rental_fina]> 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_fina]> 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 (0.001 sec)

Analisis

- `SELECT` : Bagian ini memerintahkan database untuk mengambil semua kolom dari `mobil` tabel.
- `FROM mobil` : Ini menentukan tabel dari mana data akan diambil. Dalam hal ini, itu adalah `mobil` meja.
- `WHERE pemilik LIKE 'I__'` : Klausa ini memfilter hasil berdasarkan nilai di `pemilik` kolom. Operator `LIKE` digunakan untuk pencocokan pola.
- `pemilik` : Ini mengacu pada kolom nama yang berisi informasi pemilik.
- `'I__'` : Ini adalah pola yang harus dicocokkan. Tanda kutip tunggal menunjukkan string literal. Di sini, `'I__'` mewakili pemilik yang namanya dimulai dengan huruf 'I' diikuti dengan dua garis bawah (dua karakter apa saja).
- `SELECT` : Instruksi ini meminta untuk memilih semua kolom (atribut) dari tabel `mobil`.
- `FROM mobil` : Ini menentukan tabel tempat data akan diambil, yaitu tabel `mobil`.
- `WHERE pemilik LIKE '___'` : Ini adalah klausa `WHERE` yang memfilter data berdasarkan kriteria tertentu.
- `pemilik` : Ini adalah nama kolom yang akan difilter. Biasanya ini mewakili nama pemilik kendaraan.
- `MENYUKAI '___'` :
- `LIKE` : Operator yang digunakan untuk mencari pola tertentu dalam string.
- `'___'` : Adalah pola pencarian yang digunakan.
- `___` - Mencocoki **tiga karakter apapun** (wildcard).

Kesimpulan

- `(LIKE 'I__')` : Mencari pemilik yang **diawali dengan huruf "I", diikuti oleh dua karakter apapun**. Ini akan mengembalikan pemilik dengan nama minimal 3 karakter.

- `(LIKE '___')` : Mencari pemilik yang memiliki nama dengan **panjang tepat 3 karakter apapun**.

Kombinasi

Struktur

```
SELECT * FROM nama_tabel  
-> WHERE nama_colom LIKE '___r%';
```

```
SELECT * FROM nama_tabel  
-> WHERE nama_colom LIKE '_b%';
```

Contoh

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

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

Hasil

```
MariaDB [rental_fina]> 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 (0.001 sec)

```
MariaDB [rental_fina]> 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 (0.001 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 pemilik yang namanya **berakhiran dengan huruf "r"**. Ini akan mengembalikan pemilik dengan nama minimal 3 karakter.
- (`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 nama_tabel WHERE nama_colom NOT LIKE 'A%';
```

Contoh

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

Hasil

```
MariaDB [rental_fina]> 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 (0.000 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_tabel WHERE nama_colom IS NULL;
```

Contoh

```
SELECT * FROM mobil WHERE peminjam IS NULL;
```

Hasil

```
MariaDB [rental_fina]> 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 (0.001 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_tabel WHERE nama_colom IS NOT NULL;
```

Contoh

```
SELECT * FROM mobil WHERE peminjam IS NOT NULL;
```

Hasil

```
MariaDB [rental_fina]> 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 terkecil

Struktur

```
SELECT * FROM nama_tabel ORDER BY nama_colom ASC;
```

Contoh

```
SELECT * FROM mobil ORDER BY pemilik ASC;
```

Hasil

```
MariaDB [rental_fina]> 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 (0.001 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".
- `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 terbesar

Struktur

```
SELECT * FROM nama_tabel ORDER BY nama_colom DESC;
```

Contoh

```
SELECT * FROM mobil ORDER BY peminjam DESC;
```

Hasil

```
MariaDB [rental_fina]> 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 (0.001 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

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

Contoh

```
SELECT * FROM mobil WHERE warna = "Hitam" ORDER BY harga ASC LIMIT 2;
```

Hasil

```
MariaDB [rental_fina]> SELECT * FROM mobil WHERE warna = "Hitam" ORDER BY harga_rental 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 (0.004 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

Struktur

```
SELECT DISTINCT (nama_colom) FROM nama tabel;
```

```
SELECT DISTINCT (nama_colom) FROM nama_colom ORDER BY nama_colom DESC;
```

Contoh

```
SELECT DISTINCT (warna) FROM nama tabel;
```

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

Hasil

```
MariaDB [rental_fina]> SELECT DISTINCT (harga_rental) FROM mobil ORDER BY harga_rental DESC;
```

harga_rental
150000
100000
50000

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

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.

CONCAT, CONCAT_WS, AS

Menggabungkan kolom tanpa pemisah

Struktur

```
SELECT CONCAT(colom1,colom2) FROM nama tabel;
```

Contoh

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

Hasil

```
MariaDB [rental_fina]> SELECT CONCAT (pemilik,warna) FROM mobil;
+-----+
| CONCAT (pemilik,warna) |
+-----+
| IbrahimHitam           |
| IbrahimMerah           |
| BaImSilver             |
| IbeHitam               |
| IbeHitam               |
| MADAPUTIH              |
+-----+
6 rows in set (0.003 sec)
```

Analisis

- `SELECT` digunakan untuk memilih kolom yang ingin ditampilkan.
- `CONCAT()` menggabungkan dua nilai string menjadi satu.
- `pemilik` dan `warna` adalah kolom dari tabel `mobil`.

Kesimpulan

perintah tersebut akan menggabungkan nilai dari kolom "pemilik" dan "warna" dalam tabel "mobil" menjadi satu string untuk setiap baris dalam tabel tersebut. Hasilnya akan berupa satu kolom baru yang bernama "CONCAT" yang berisi hasil penggabungan tersebut.

Menggabungkan kolom dengan pemisah

Struktur

```
SELECT CONCAT_WS ('-', colom1, colom2, colom3) FROM nama tabel ;
```

Contoh

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

Hasil


```
MariaDB [rental_fina]> 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 |
| DD 2378 AZ-ACFDT-6 |
+-----+
6 rows in set (0.002 sec)
```

Analisis

- `SELECT` digunakan untuk memilih kolom yang ingin ditampilkan.
- `CONCAT_WS` berfungsi untuk menggabungkan beberapa nilai string menjadi satu string, dipisahkan dengan separator yang ditentukan.
 - `WS` artinya "With Space" (dengan spasi).
 - `"-"` adalah pembatas yang digunakan dalam kueri ini (bisa diganti dengan karakter lain).
- `no_plat`, `no_mesin`, dan `id_mobil` adalah kolom dari tabel `mobil`.

Kesimpulan

kesimpulan dari `SELECT CONCAT_WS("-", no_plat, no_mesin, id_mobil) FROM mobil;` adalah bahwa Anda sedang menggabungkan kolom `no_plat`, `no_mesin`, dan `id_mobil` dalam tabel `mobil` menjadi satu string dengan menggunakan tanda hubung (-) sebagai pemisah.

Memberikan nama kolom alias

Struktur

```
SELECT CONCAT_WS ("+",namacolom1,colom2) AS COLLAB FROM nama tabel;
```

Contoh

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

Hasil

```
MariaDB [rental_fina]> SELECT CONCAT_WS ("+", pemilik, peminjam) AS COLLAB FROM mobil;
+-----+
| COLLAB |
+-----+
| Ibrahim+Afdal |
| Ibrahim+Elia |
| Baim+Anty |
| Ibe |
| Ibe |
| MADA |
+-----+
6 rows in set (0.009 sec)
```

Analisis

- `SELECT` digunakan untuk memilih kolom yang ingin ditampilkan.
- `CONCAT_WS` berfungsi untuk menggabungkan dua nilai string menjadi satu string, dipisahkan dengan separator yang ditentukan.
 - `WS` artinya "With Space" (dengan spasi).
 - `+` adalah pembatas yang digunakan dalam kueri ini (bisa diganti dengan karakter lain).
- `pemilik` dan `peminjam` adalah kolom dari tabel `mobil`.
- `AS COLLAB` memberi alias pada kolom baru sebagai "COLLAB".

Kesimpulan

Kesimpulan dari pernyataan SQL `SELECT CONCAT_WS("+", pemilik, peminjam) AS COLLAB FROM mobil;` adalah bahwa Anda sedang menggabungkan kolom pemilik dan peminjam dalam tabel mobil menjadi satu string dengan menggunakan tanda tambah (+) sebagai pemisah.

VIEW

Membuat tabel virtual

Struktur

```
CREATE VIEW nama_colom_virtual AS
-> SELECT colom1, colom2, colom3, colom4
-> FROM mobil
-> WHERE colom4 = "Ibrahim";
```

Contoh

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

Hasil

```
MariaDB [rental_fina]> show tables;
+-----+
| Tables_in_rental_fina |
+-----+
| info_no_plat          |
| mobil                 |
| pelanggan             |
+-----+
3 rows in set (0.001 sec)
```

Analisis

- `CREATE VIEW` digunakan untuk membuat view baru.
- `info_no_plat` adalah nama view yang dibuat.
- `SELECT` digunakan untuk memilih kolom yang ingin ditampilkan dalam view.
- `id_mobil, no_plat, pemilik, dan peminjam` adalah kolom yang ditampilkan dalam view.
- `FROM` menunjukkan tabel sumber data, yaitu `mobil`.
- `WHERE` digunakan untuk menyaring data berdasarkan kondisi, yaitu `pemilik = "Ibrahim"`.

Kesimpulan

Kesimpulan dari `CREATE VIEW info_no_plat AS SELECT id_mobil, no_plat, pemilik, peminjam FROM mobil WHERE pemilik = "Ibrahim";` adalah bahwa Anda sedang membuat sebuah view dengan nama "info_no_plat". View ini akan menampilkan kolom-kolom id_mobil, no_plat, pemilik, dan peminjam dari tabel mobil hanya untuk baris-baris di mana nilai kolom pemilik adalah "Ibrahim".

Menampilkan tabel virtual

Struktur

```
SELECT * FROM nama_colom_virtual;
```

Contoh

```
SELECT * FROM info_no_plat;
```

Hasil

```
MariaDB [rental_fina]> SELECT * FROM info_no_plat;
+-----+-----+-----+-----+
| id_mobil | no_plat   | pemilik | peminjam |
+-----+-----+-----+-----+
|          1 | DD 2650 XY | Ibrahim | Afdal    |
|          2 | DD 2440 AX | Ibrahim | Elia     |
+-----+-----+-----+-----+
2 rows in set (0.002 sec)
```

Analisis

- `SELECT *` digunakan untuk memilih semua kolom yang ada di view `info_no_plat`.
- `FROM info_no_plat` menunjukkan bahwa data akan diambil dari view `info_no_plat`.

Kesimpulan

perintah `SELECT *`, Anda memilih semua kolom yang ada dalam view `info_no_plat`. Hasilnya akan mengembalikan semua baris yang memenuhi kondisi `pemilik = "Ibrahim"` dari tabel `mobil`, termasuk semua kolom yang ada dalam view tersebut.

Menghapus tabel virtual

Struktur

```
DROP VIEW info_no_plat;
```

Contoh

```
DROP VIEW info_no_plat;
```

Hasil

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

MariaDB [rental_fina]> show tables;
+-----+
| Tables_in_rental_fina |
+-----+
| mobil                 |
| pelanggan             |
+-----+
2 rows in set (0.001 sec)
```

Analisis

- `DROP VIEW` : Ini adalah perintah yang digunakan untuk menghapus view dari database. Dalam hal ini, perintah ini digunakan untuk menghapus view "info_no_plat".
- `info_no_plat` : Ini adalah nama view yang akan dihapus. Nama ini harus sesuai dengan nama view yang ingin Anda hapus.

| Kesimpulan

Perintah `DROP VIEW` digunakan untuk menghapus view yang telah dibuat sebelumnya. Dalam hal ini, perintah tersebut akan menghapus view "info_no_plat" dari database. Setelah perintah dijalankan, view tersebut tidak akan lagi tersedia dan tidak dapat digunakan untuk mengambil data.

| TANTANGAN VIEW

| nomor 1

| penjelasan

- Perintah `CREATE VIEW` digunakan untuk membuat sebuah view baru.
- `mobil_tanpa_peminjaman` adalah nama dari view yang akan dibuat.
- Kata kunci `AS` menunjukkan bahwa definisi view akan mengikuti setelahnya.
- Bagian ini adalah perintah `SELECT` yang menentukan kolom mana yang akan dimasukkan ke dalam view.
- `no_plat` dan `peminjaman` adalah kolom yang diambil dari tabel `daftar_mobil`.
- Menentukan tabel sumber data, yaitu `daftar_mobil`.

| query

```
CREATE VIEW
-> mobil_tanpa_peminjaman AS
-> SELECT no_plat,peminjaman
-> FROM daftar_mobil;
```

| hasil

```
MariaDB [rental_mada]> select * from mobil_tanpa_peminjaman;
+-----+-----+
| no_plat | peminjaman |
+-----+-----+
| DD 2650 XY | Afdal |
| DD 2440 AX | Elia |
| B 1611 QC | Anty |
| DD 2901 JK | Null |
| DD 2210 LS | NULL |
| A 1406 ZIS | NULL |
+-----+-----+
6 rows in set (0.050 sec)
```

kesimpulan

Tampilan `mobil_tanpa_peminjaman` akan menampilkan informasi tentang mobil-mobil dari tabel `daftar_mobil` yang **tidak sedang dipinjamkan**.

nomor 2

penjelasan

- Perintah `UPDATE` digunakan untuk memperbarui baris-baris dalam tabel.
- `daftar_mobil` adalah nama tabel yang akan diperbarui.
- Bagian `SET` menentukan kolom mana yang akan diperbarui dan nilai baru yang akan ditetapkan.
- `peminjaman = NULL` berarti kolom `peminjaman` akan diatur menjadi `NULL`.
- Klausa `WHERE` digunakan untuk menentukan kondisi yang harus dipenuhi oleh baris yang akan diperbarui.
- `peminjaman = 'Afdal'` berarti hanya baris-baris di mana nilai kolom `peminjaman` adalah `'Afdal'` yang akan diperbarui.

query

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

hasil

```
MariaDB [rental_mada]> SELECT * FROM mobil_tanpa_peminjaman;
+-----+-----+
| no_plat | peminjaman |
+-----+-----+
| DD 2650 XY | NULL |
| DD 2440 AX | Elia |
| B 1611 QC | Anty |
| DD 2901 JK | Null |
| DD 2210 LS | NULL |
| A 1406 ZIS | NULL |
+-----+-----+
6 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

Perintah SUM digunakan untuk menghitung total dari nilai-nilai numerik dalam suatu kolom. Ini sering digunakan bersama dengan pernyataan SELECT untuk mengambil hasil agregasi dari data dalam tabel.

Struktur

```
SELECT SUM(nama_colom) AS total
FROM nama_table
WHERE kondisi_opsional;
```

Contoh

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

Hasil

```
MariaDB [rental_fina]> SELECT SUM(harga_rental) FROM mobil;
+-----+
| SUM(harga_rental) |
+-----+
|          450000   |
+-----+
1 row in set (0.041 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

Perintah COUNT digunakan untuk menghitung jumlah baris yang memenuhi suatu kondisi atau untuk menghitung jumlah baris dalam suatu tabel.

Struktur

```
SELECT COUNT(*) AS jumlah
FROM nama_tabel
WHERE kondisi_opsional;
```

Contoh


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

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

Hasil

```
MariaDB [rental_fina]> SELECT COUNT(pemilik) FROM mobil;
+-----+
| COUNT(pemilik) |
+-----+
|                5 |
+-----+
1 row in set (0.032 sec)
```

```
MariaDB [rental_fina]> SELECT COUNT(peminjam) FROM mobil;
+-----+
| COUNT(peminjam) |
+-----+
|                 3 |
+-----+
1 row in set (0.001 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

Fungsi MIN digunakan menemukan nilai minimum dalam suatu kolom. Fungsi ini berguna ketika kita ingin mengetahui nilai terkecil dalam satu set data.

Struktur

```
SELECT MIN(nama_colom) AS nilai_minimum
FROM nama_tabel
WHERE kondisi_opsional;
```

Contoh

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

Hasil

```
MariaDB [rental_fina]> SELECT MIN(harga_rental) AS MINIMAL FROM mobil;
+-----+
| MINIMAL |
+-----+
|  50000  |
+-----+
1 row in set (0.040 sec)
```

Analisis

- `SELECT`: Menunjukkan bahwa ingin mengambil data dari database.
- `MIN(harga_rental)`: Menghitung nilai minimum kolom `harga_rental`.
- `AS MINIMAL`: Menetapkan alias `MINIMAL` pada hasil fungsi `MIN(harga_rental)`.
- `FROM mobil`: Menentukan tabel `mobil` untuk mengambil data.

Kesimpulan

Kesimpulan dari `SELECT MIN(harga_rental) AS MINIMAL FROM mobil;` adalah bahwa Anda ingin mencari nilai terkecil (minimum) dari kolom "harga_rental" dalam tabel "mobil".

Max

Fungsi MAX digunakan untuk menemukan nilai maksimum dari suatu kolom. Fungsi ini berguna ketika kita ingin mengetahui nilai terbesar dalam satu set data.

Struktur

```
SELECT MAX(nama_colom) AS nilai_maksimum
FROM nama_tabel
WHERE kondisi_opsional;
```

Contoh

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

Hasil

```
MariaDB [rental_fina]> SELECT MAX(harga_rental) AS MAXIMAL FROM mobil;
+-----+
| MAXIMAL |
+-----+
| 150000 |
+-----+
1 row in set (0.038 sec)
```

Analisis

- `SELECT`: Digunakan untuk mengambil data dari database.
- `MAX(harga_rental)`: Fungsi ini menghitung nilai maksimum kolom 'harga_rental' dan mengambil nilai tertinggi yang ada di kolom itu.
- `AS MAXIMAL`: Mengubah nama hasil perhitungan `MAX(harga_rental)` menjadi 'MAXIMAL'. Itu adalah alias untuk hasilnya.

Kesimpulan

Kesimpulan dari pernyataan SQL ini adalah bahwa Anda ingin mencari nilai terbesar (maksimum) dari kolom "harga_rental" dalam tabel "mobil". Hasilnya akan disajikan dengan menggunakan alias "MAXIMAL" dalam kolom hasil.

AVG

Fungsi agregasi yang digunakan untuk menghitung nilai rata-rata

Struktur

```
SELECT AVG(nama_colom) AS rata_rata
FROM nama_tabel
WHERE kondisi_opsional;
```

Contoh

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

Hasil

```
MariaDB [rental_fina]> SELECT AVG(harga_rental) AS RATA_RATA FROM mobil;
+-----+
| RATA_RATA |
+-----+
| 90000.0000 |
+-----+
1 row in set (0.001 sec)
```

Analisis

- `SELECT AVG(harga_rental)`: Ini adalah perintah SQL yang digunakan untuk mengambil nilai rata-rata dari kolom `harga_rental`. `AVG()` adalah fungsi agregat yang digunakan untuk menghitung rata-rata dari sekumpulan nilai.
- `AS RATA_RATA`: Menggunakan kata kunci `AS` untuk memberi nama pada hasil rata-rata. Dalam hal ini, hasil rata-rata dari `harga_rental` akan dinamai `RATA_RATA`.
- `FROM mobil`: Ini adalah bagian dari perintah SQL yang menunjukkan bahwa data diambil dari tabel bernama `mobil`. Jadi, perintah ini mengambil nilai rata-rata dari kolom `harga_rental` dari tabel `mobil`.

Kesimpulan

Kesimpulan dari pernyataan SQL `SELECT AVG(harga_rental) AS RATA_RATA FROM mobil;` adalah bahwa Anda ingin menghitung nilai rata-rata (average) dari kolom "harga_rental" dalam tabel "mobil".

Relasi

Data tabel

Browse

Structure

SQL

Search

Insert

Export

Import

Privileges

Operati

Extra options

←T→

▼

idpeg

nama

namakel

jk

alamat

kota

telp

nocab

☐

Edit

Copy

Delete

10107

Emya

Salsalina

P

Jl.suci 78

Bandung

022-555768

C101

☐

Edit

Copy

Delete

10176

Diah

Wahyuni

P

Jl.Maluku 56

Bandung

022-555934

C101

☐

Edit

Copy

Delete

10246

Dian

Anggraini

P

Jl.Mawar 5

Semarang

024-555102

C103

☐

Edit

Copy

Delete

10252

Antoni

Irawan

L

Jl.A.Yani 15

Jakarta

021-555888

C102

☐

Edit

Copy

Delete

10307

Erik

Andrian

L

Jl.Manggis 155

Semarang

024-555236

C103

☐

Edit

Copy

Delete

10314

Ayu

Rahmadani

P

Jl.Malaka 342

Jakarta

021-555098

C102

☐

Edit

Copy

Delete

10324

Martin

Susanto

L

Jl.Bima 51

Jakarta

021-555785

C102

☐

Edit

Copy

Delete

10407

Rio

Gunawan

L

Jl.Melati 356

Surabaya

031-555231

C104

☐

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

Before

[Edit dikotak] [Ubah] [Buat kode PHP]									
<div> Struktur tabel Tampilan hubungan </div>									
#	Nama	Jenis	Penyortiran	Atribut	Tak Ternilai	Bawaan	Komentar	Ekstra	Tindakan
<input type="checkbox"/> 1	IdPeg	int(5)			Tidak	Tidak ada			Ubah Hapus Lainnya
<input type="checkbox"/> 2	Nama	varchar(100)	utf8mb4_general_ci		Tidak	Tidak ada			Ubah Hapus Lainnya
<input type="checkbox"/> 3	NamaKel	varchar(100)	utf8mb4_general_ci		Tidak	Tidak ada			Ubah Hapus Lainnya
<input type="checkbox"/> 4	JK	varchar(11)	utf8mb4_general_ci		Tidak	Tidak ada			Ubah Hapus Lainnya
<input type="checkbox"/> 5	Alamat	varchar(1000)	utf8mb4_general_ci		Tidak	Tidak ada			Ubah Hapus Lainnya
<input type="checkbox"/> 6	Kota	varchar(20)	utf8mb4_general_ci		Tidak	Tidak ada			Ubah Hapus Lainnya
<input type="checkbox"/> 7	Telp	int(15)			Tidak	Tidak ada			Ubah Hapus Lainnya
<input type="checkbox"/> 8	NoCap	varchar(5)	utf8mb4_general_ci		Tidak	Tidak ada			Ubah Hapus Lainnya

After

Browse Structure SQL Search Insert Export Import Privileges Operations										
Table structure		Relation view								
#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action	
<input type="checkbox"/> 1	Idpeg	int(5)			No	None			Change	Drop More
<input type="checkbox"/> 2	nama	varchar(50)	utf8mb4_general_ci		No	None			Change	Drop More
<input type="checkbox"/> 3	namaKel	varchar(50)	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(50)	utf8mb4_general_ci		No	None			Change	Drop More
<input type="checkbox"/> 6	kota	varchar(50)	utf8mb4_general_ci		No	None			Change	Drop More
<input type="checkbox"/> 7	no_telp	int(15)			No	None			Change	Drop More
<input type="checkbox"/> 8	noCab	varchar(10)	utf8mb4_general_ci		No	None			Change	Drop More

☐ Check all
 With selected:
 Browse
 Change
 Drop
 Primary
 Unique
 Index
 SQL

Perubahan Data Tabel

before pegawai

Browse Structure SQL Search Insert Export Import Privileges Operati										
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 pegawai

Server: 127.0.0.1 » Database: xi_rpl » Table: pegawai

[Browse](#)
[Structure](#)
[SQL](#)
[Search](#)
[Insert](#)
[Export](#)
[Import](#)
[Privileges](#)
[Op](#)

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

☐ Show all | Number of rows: 25 | Filter rows: Sort by key: None

Extra options

				IdPeg	Nama	NamaKel	JK	Alamat	Kota	Telp	NoCab
<input type="checkbox"/>				10107	Nur	Afni	P	Jl. limboto	makassar	022-555768	C101
<input type="checkbox"/>				10176	Diah	Wahyuni	P	Jl. maluku 56	bandung	022-555934	C101
<input type="checkbox"/>				10246	Dian	Anggraini	P	Jl. mawar 5	semarang	024-555102	C103
<input type="checkbox"/>				10252	Antoni	Irawan	L	Jl. A. yani 15	jakarta	021-555888	C102
<input type="checkbox"/>				10307	Erik	Andrian	L	Jl. manggis 155	semarang	024-555236	C103
<input type="checkbox"/>				10314	Ayu	Rahmadani	P	Jl. malaka 342	jakarta	021-555098	C102
<input type="checkbox"/>				10324	Martin	Susanto	L	Jl. bima 51	jakarta	021-555785	C102
<input type="checkbox"/>				10407	Rio	Gunawan	L	Jl. melati	surabaya	031-555231	C104
<input type="checkbox"/>				10415	Susan	Sumantri	P	Jl. pahlawan 24	surabaya	031-555120	C104

before cabang

Extra options

				noCab	nama	alamat	kabupaten	no_telp	kab_kota	email
<input type="checkbox"/>				C101	frhan	Jl.lompobattang	Makassar	819183765	Makassar	frhan@gmail.com
<input type="checkbox"/>				C102	hansar	Jl.maccini	Makassar	819183764	Makassar	hansar@gmail.com
<input type="checkbox"/>				C103	rehan	Jl.rajawali	Makassar	819183763	Makassar	rehan@gmail.com
<input type="checkbox"/>				C104	ardi	Jl.maccini	Makassar	819183762	Makassar	ardi@gmail.com

☐ Pilih Semua Dengan pilihan:

after cabang

Extra options

				No_cab	Nama	Alamat	Kabupaten	No_telp	Kab_kota	Email
<input type="checkbox"/>				C101	Fatsa Akhwani	Jl. Antang	kota makassar	85739352	Makassar	ftsaaakhwani23@gmail.com
<input type="checkbox"/>				C102	Nur Afni Ramadani	Jl. Limboto	kota makassar	86683273	Makassar	Nurafniaf@gmail.com
<input type="checkbox"/>				C103	Siti Nur Hasiza. A	Jl. Amblam	kota makassar	86857389	Makassar	sitinrhasiza@gmail.com

☐ Check all With selected:

Hasil relasi & pegawai

warna	jumlah_warna
Hitam	3
Merah	2
Silver	2

3 rows in set (0.001 sec)

Analisis

- `SELECT`: Memilih kolom `warna` dari tabel `mobil`.
- `COUNT(id_mobil) AS jumlah_warna`: Menghitung jumlah baris (atau mobil dalam konteks ini) yang memiliki nilai yang sama pada kolom `warna`. Hasil perhitungannya akan disajikan dengan nama kolom baru `jumlah_warna`.
- `FROM mobil`: Menunjukkan tabel sumber data yang digunakan adalah `mobil`.
- `GROUP BY warna`: Mengelompokkan hasil berdasarkan nilai unik yang terdapat di kolom `warna`. Ini berarti semua baris dengan nilai `warna` yang sama akan dikelompokkan menjadi satu baris dalam hasil akhir, dan `COUNT(id_mobil)` akan memberikan jumlah mobil dengan warna tersebut.

Kesimpulan

Query ini digunakan untuk menghitung jumlah mobil berdasarkan warnanya. menggunakan `GROUP BY warna`, dapat melihat berapa banyak mobil yang memiliki warna yang sama.

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

Query

```
SELECT pemilik,
-> COUNT(id_mobil) AS jumlah_mobil
-> FROM mobil
-> GROUP BY pemilik
-> HAVING COUNT(id_mobil) >= 3;
```

Hasil

pemilik	jumlah_mobil
mada	3

1 row in set (0.008 sec)

Analisis

- `SELECT`: Memilih kolom `pemilik` dari tabel `mobil`.
- `COUNT(id_mobil) AS jumlah_mobil`: Menghitung jumlah mobil yang dimiliki oleh setiap `pemilik`. Hasil perhitungan disimpan dalam kolom baru yang dinamai `jumlah_mobil`.
- `FROM mobil`: Menunjukkan bahwa data diambil dari tabel `mobil`.
- `GROUP BY pemilik`: Mengelompokkan hasil berdasarkan nilai unik pada kolom `pemilik`. Ini berarti semua baris dengan nilai `pemilik` yang sama dikelompokkan menjadi satu baris dalam hasil akhir.
- `HAVING COUNT(id_mobil) >= 3`: Kondisi `HAVING` digunakan setelah `GROUP BY` untuk memfilter hasil berdasarkan jumlah mobil (`COUNT(id_mobil)`) yang dimiliki oleh setiap `pemilik`. Dalam kasus ini, hanya grup `pemilik` yang memiliki 3 mobil atau lebih yang akan disertakan dalam hasil akhir query.

Kesimpulan

untuk menemukan dan menampilkan semua `pemilik` yang memiliki 3 mobil atau lebih dalam tabel `mobil`. Dengan menggunakan `GROUP BY pemilik` dan `HAVING COUNT(id_mobil) >= 3`, Anda dapat mengidentifikasi pemilik-pemilik yang memiliki mobil dalam jumlah yang signifikan.

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

Query

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

Hasil

pemilik	jumlah_mobil
alice	1
fni	1
gusion	1
mada	3
yuzhong	1

5 rows in set (0.039 sec)

Analisis

- `SELECT`: Memilih kolom `pemilik` dari tabel `mobil`.
- `COUNT(id_mobil) AS jumlah_mobil`: Menghitung jumlah mobil yang dimiliki oleh setiap `pemilik`. Hasil perhitungan disimpan dalam kolom baru yang dinamai `jumlah_mobil`.
- `FROM mobil`: Menunjukkan bahwa data diambil dari tabel `mobil`.
- `GROUP BY pemilik`: Mengelompokkan hasil berdasarkan nilai unik pada kolom `pemilik`. Ini berarti semua baris dengan nilai `pemilik` yang sama dikelompokkan menjadi satu baris dalam hasil akhir.

Kesimpulan

Query ini digunakan untuk melihat jumlah mobil yang dimiliki oleh setiap `pemilik` dalam tabel `mobil`. Dengan menggunakan `GROUP BY pemilik`, kita dapat melihat distribusi atau jumlah mobil yang dimiliki oleh masing-masing pemilik secara terperinci.

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

Query

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

Hasil

pemilik	total_pendapatan
alice	5550000
fni	100000
gusion	150000
mada	500000
yuzhong	100000

5 rows in set (0.040 sec)

Analisis

- `SELECT` : Memilih kolom `pemilik` dari tabel `mobil`.
- `SUM(harga_rental) AS total_pendapatan` : Menghitung jumlah total dari `harga_rental` untuk setiap `pemilik`. Hasil perhitungan disimpan dalam kolom baru yang dinamai `total_pendapatan`.
- `FROM mobil` : Menunjukkan bahwa data diambil dari tabel `mobil`.
- `GROUP BY pemilik` : Mengelompokkan hasil berdasarkan nilai unik pada kolom `pemilik`. Ini berarti semua baris dengan nilai `pemilik` yang sama dikelompokkan menjadi satu baris dalam hasil akhir.

Kesimpulan

Query SQL untuk mengetahui total pendapatan dari penyewaan 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

Query

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

Hasil

pemilik	total_pemasukan
alice	5550000
mada	500000

2 rows in set (0.039 sec)

Analisis

- `SELECT` : Memilih kolom `pemilik` dari tabel `mobil`.
- `SUM(harga_rental) AS total_pemasukan` : Menghitung jumlah total dari `harga_rental` untuk setiap `pemilik`. Hasil perhitungan disimpan dalam kolom baru yang dinamai `total_pemasukan`.
- `FROM mobil` : Menunjukkan bahwa data diambil dari tabel `mobil`.
- `GROUP BY pemilik` : Mengelompokkan hasil berdasarkan nilai unik pada kolom `pemilik`. Ini berarti semua baris dengan nilai `pemilik` yang sama dikelompokkan menjadi satu baris dalam hasil akhir.
- `HAVING SUM(harga_rental) >= 300000` : Kondisi `HAVING` digunakan setelah `GROUP BY` untuk memfilter hasil berdasarkan total pemasukan (`SUM(harga_rental)`) yang dimiliki oleh setiap `pemilik`. Dalam kasus ini, hanya grup `pemilik` yang memiliki total pemasukan dari rental mobil sebesar atau lebih dari 300000 yang akan disertakan dalam hasil akhir.

Kesimpulan

Query ini digunakan `GROUP BY pemilik` dan `HAVING SUM(harga_rental) >= 300000` dan dapat mengidentifikasi pemilik yang menghasilkan pendapatan signifikan dari rental mobil.

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

Query

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

Hasil

pemilik	total_pemasukan
alice	5550000.0000
fni	100000.0000
gusion	150000.0000
mada	166666.6667
yuzhong	100000.0000

5 rows in set (0.039 sec)

Analisis

- `SELECT`: Memilih kolom `pemilik` dari tabel `mobil`.
- `AVG(harga_rental) AS total_pemasukan`: Menghitung rata-rata `harga_rental` untuk setiap `pemilik`. Hasil perhitungan disimpan dalam kolom baru yang dinamai `total_pemasukan`.
- `FROM mobil`: Menunjukkan bahwa data diambil dari tabel `mobil`.
- `GROUP BY pemilik`: Mengelompokkan hasil berdasarkan nilai unik pada kolom `pemilik`. Ini berarti semua baris dengan nilai `pemilik` yang sama dikelompokkan menjadi satu baris dalam hasil akhir.

Kesimpulan

Query ini digunakan untuk melihat rata-rata harga rental mobil yang diperoleh setiap pemilik dalam tabel `mobil`. Dengan menggunakan `GROUP BY pemilik`, kita dapat mengetahui rata-rata pendapatan per rental 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

Query

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

Hasil

pemilik	pemasukan_terbesar	pemasukan_terkecil
mada	250000	50000

1 row in set (0.039 sec)

Analisis

- `SELECT`: Memilih kolom `pemilik` dari tabel `mobil`.
- `MAX(harga_rental) AS pemasukan_terbesar`: Menghitung nilai maksimal dari `harga_rental` untuk setiap `pemilik`. Hasil perhitungan disimpan dalam kolom baru yang dinamai `pemasukan_terbesar`.
- `MIN(harga_rental) AS pemasukan_terkecil`: Menghitung nilai minimal dari `harga_rental` untuk setiap `pemilik`. Hasil perhitungan disimpan dalam kolom baru yang dinamai `pemasukan_terkecil`.
- `FROM mobil`: Menunjukkan bahwa data diambil dari tabel `mobil`.
- `GROUP BY pemilik`: Mengelompokkan hasil berdasarkan nilai unik pada kolom `pemilik`. Ini berarti semua baris dengan nilai `pemilik` yang sama dikelompokkan menjadi satu baris dalam hasil akhir.
- `HAVING COUNT(id_mobil) > 1`: Kondisi `HAVING` digunakan setelah `GROUP BY` untuk memfilter hasil berdasarkan jumlah mobil (`COUNT(id_mobil)`) yang dimiliki oleh setiap `pemilik`. Dalam kasus ini, hanya grup `pemilik` yang memiliki lebih dari satu mobil yang akan disertakan dalam hasil akhir.

Kesimpulan

Query ini digunakan untuk melihat pemilik mobil yang memiliki lebih dari satu mobil dan untuk mengetahui nilai maksimal dan minimal dari harga rental mobil yang dimiliki oleh setiap pemilik dalam tabel `mobil`. menggunakan `GROUP BY pemilik` dan `HAVING COUNT(id_mobil) > 1`, dapat melihat pemilik yang memiliki lebih dari satu mobil beserta rentang harga rentalnya.