

Select Lanjutan

Select Lanjutan

Setelah mempelajari select di materi sebelumnya sekarang kita akan masuk ke dalam materi select lanjutan, fungsi dari select select ini ialah untuk mendapatkan hasil yang lebih spesifik dan lebih luas, sekarang kita akan mempelajari 7 select lanjutan (**AND ,OR ,BETWEEN-AND ,NOT BETWEEN ,<= ,>= ,<> ATAU !=**) Untuk penjelasan lebih lanjutnya ialah seperti berikut :

 Isi Table yang akan digunakan : >

id_pelanggan	no_plat	no_mesin	warna	pemilik	peminjam	harga_rental
1	DD 2650 XY	ACX3568	Hitam	Ibrahim	Afdal	50000
2	DD 2440 AX	BCS1120	Merah	Ibrahim	Elia	100000
3	B 1611 QC	LSQ1112	Silver	Baim	Anty	50000
4	DD 2901 JK	UQL1029	Hitam	Ibe	NULL	150000
5	DD 2210 LS	CJH1011	Hitam	Ibe	NULL	100000

AND

untuk **AND** ini akan mengambil "data 1" *dan* "data 2", contoh kodenya adalah seperti berikut :

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

Dan hasilnya akan seperti berikut :

```
MariaDB [rental_nero]> SELECT warna,pemilik FROM mobil WHERE warna="Hitam" AND pemilik="ibrahim";
+-----+-----+
| warna | pemilik |
+-----+-----+
| Hitam | Ibrahim |
+-----+-----+
1 row in set (0.001 sec)
```

OR

Untuk **OR** ini akan mengambil "data 1" *atau* "data 2", contoh kodenya ialah seperti berikut :

```
SELECT warna,pemilik FROM mobil WHERE warna="Hitam" OR pemilik="ibrahim";
```

Dan hasilnya akan seperti berikut :

```
MariaDB [rental_nero]> SELECT warna,pemilik FROM mobil WHERE warna="Hitam" OR pemilik="ibrahim";
+-----+-----+
| warna | pemilik |
+-----+-----+
| Hitam | Ibrahim |
| Merah | Ibrahim |
| Hitam | Ibe     |
| Hitam | Ibe     |
+-----+-----+
4 rows in set (0.000 sec)
```

BETWEEN-AND

Untuk **BETWEEN-AND** ini akan mengambil antara "data 1" *sampai* "data 2" dibantu dengan **AND**, contoh kodenya ialah seperti berikut :

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

Dan hasilnya akan seperti berikut :

```
MariaDB [rental_nero]> SELECT * FROM mobil WHERE harga_rental BETWEEN 100000 AND 200000;
+-----+-----+-----+-----+-----+-----+-----+
| id_pelanggan | no_plat   | no_mesin | warna | pemilik | peminjam | harga_rental |
+-----+-----+-----+-----+-----+-----+-----+
|          2   | DD 2440 AX | BCS1120  | Merah | Ibrahim | Elia     |      100000  |
|          4   | DD 2901 JK | UQL1029  | Hitam | Ibe     | NULL     |      150000  |
|          5   | DD 2210 LS | CJH1011  | Hitam | Ibe     | NULL     |      100000  |
+-----+-----+-----+-----+-----+-----+-----+
3 rows in set (0.000 sec)
```

NOT BETWEEN

Untuk **NOT BETWEEN** ini akan mengambil "data" yang *bukan antara* "data 1" *dan* "data 2", contoh kodenya ialah seperti berikut :

```
MariaDB [rental_nero]> SELECT * FROM mobil WHERE harga_rental NOT BETWEEN 100000 AND 200000;
```

Dan hasilnya akan seperti berikut :

```
MariaDB [rental_nero]> SELECT * FROM mobil WHERE harga_rental NOT BETWEEN 100000 AND 200000;
+-----+-----+-----+-----+-----+-----+-----+
| id_pelanggan | no_plat   | no_mesin | warna  | pemilik | peminjam | harga_rental |
+-----+-----+-----+-----+-----+-----+-----+
|          1   | DD 2650 XY | ACX3568  | Hitam  | Ibrahim | Afdal    |      50000   |
|          3   | B 1611 QC  | LSQ1112  | Silver | Baim    | Anty     |      50000   |
+-----+-----+-----+-----+-----+-----+-----+
2 rows in set (0.006 sec)
```

<=

Untuk <= ini akan mengambil "data" lebih kecil atau sama dengan "nilai data", contoh kodenya ialah seperti berikut :

```
MariaDB [rental_nero]> SELECT * FROM mobil WHERE harga_rental <= 50000;
```

Dan hasilnya akan seperti berikut :

```
MariaDB [rental_nero]> SELECT * FROM mobil WHERE harga_rental <= 50000;
+-----+-----+-----+-----+-----+-----+-----+
| id_pelanggan | no_plat   | no_mesin | warna  | pemilik | peminjam | harga_rental |
+-----+-----+-----+-----+-----+-----+-----+
|          1   | DD 2650 XY | ACX3568  | Hitam  | Ibrahim | Afdal    |      50000   |
|          3   | B 1611 QC  | LSQ1112  | Silver | Baim    | Anty     |      50000   |
+-----+-----+-----+-----+-----+-----+-----+
2 rows in set (0.001 sec)
```

>=

Untuk **>=** ini akan mengambil "data" lebih besar atau sama dengan "nilai data", contoh kodenya ialah seperti berikut :

```
MariaDB [rental_nero]> SELECT * FROM mobil WHERE harga_rental >= 50000;
```

Dan hasilnya akan seperti berikut :

```
MariaDB [rental_nero]> SELECT * FROM mobil WHERE harga_rental >= 50000;
+-----+-----+-----+-----+-----+-----+-----+
| id_pelanggan | no_plat   | no_mesin | warna  | pemilik | peminjam | harga_rental |
+-----+-----+-----+-----+-----+-----+-----+
|          1   | DD 2650 XY | ACX3568  | Hitam  | Ibrahim | Afdal    |          50000 |
|          2   | DD 2440 AX | BCS1120  | Merah  | Ibrahim | Elia     |         100000 |
|          3   | B 1611 QC  | LSQ1112  | Silver | Baim    | Anty     |          50000 |
|          4   | DD 2901 JK | UQL1029  | Hitam  | Ibe     | NULL     |         150000 |
|          5   | DD 2210 LS | CJH1011  | Hitam  | Ibe     | NULL     |         100000 |
+-----+-----+-----+-----+-----+-----+-----+
5 rows in set (0.001 sec)
```

<> atau !=

Untuk **<>** atau **!=** ini akan mengambil "data" yang tidak sama dengan "nilai data", contoh kodenya ialah seperti berikut :

```
MariaDB [rental_nero]> SELECT * FROM mobil WHERE harga_rental <> 50000;
```

Dan hasilnya akan seperti berikut :

```
MariaDB [rental_nero]> SELECT * FROM mobil WHERE harga_rental <> 50000;
```

id_pelanggan	no_plat	no_mesin	warna	pemilik	peminjam	harga_rental
2	DD 2440 AX	BCS1120	Merah	Ibrahim	Elia	100000
4	DD 2901 JK	UQL1029	Hitam	Ibe	NULL	150000
5	DD 2210 LS	CJH1011	Hitam	Ibe	NULL	100000

3 rows in set (0.000 sec)

Tantangan I

Untuk tantangan saya akan mengambil nama pemilik "Ibe" dengan cara memanggilnya dengan syarat nomor pelatnya yaitu "DD 2901 JK" lalu hasilnya akan seperti berikut :

"Aset/IMG4/IMG4(9).png" could not be found.

Analisis >

"AND" : Mengambil data 1 **dan** data 2.

"OR" : Mengambil data antara data 1 **atau** data 2.

"BETWEEN-AND" : Mengambil data **antara** data 1 **sampai** data 2.

"NOT BETWEEN" : Mengambil data yang tidak **antara** data 1 **sampai** data 2.

"<=" : Mengambil data yang lebih kecil atau sama dengan nilai data.

">=" : Mengambil data yang lebih besar atau sama dengan nilai data.

"<> atau !=" : Mengambil data yang **tidak** sama dengan nilai data.

Kesimpulan :

Select ini memiliki cakupan yang luas dan bervariasi semuanya juga memiliki kelebihan dan keunikan masing masing sehingga dapat menampilkan hasil sebuah nilai yang diinginkan, keberagaman select ini mulai dari AND ,OR ,BETWEEN-AND ,NOT BETWEEN ,<= ,>= , <> ATAU !=.

Pertemuan 5

IN

Untuk select lanjutan berikutnya ialah IN yang berfungsi untuk menampilkan data yang disebutkan pada dalam kurung yang diminta oleh IN, contoh kodenya akan seperti dibawah ini :

```
select * from mobil WHERE warna IN ('Hitam','Silver');
```

Dan hasilnya akan seperti berikut :

```
MariaDB [rental_nero]> SELECT * FROM mobil WHERE warna IN ('Hitam','Silver');
+-----+-----+-----+-----+-----+-----+-----+
| id_pelanggan | no_plat   | no_mesin | warna  | pemilik | peminjam | harga_rental |
+-----+-----+-----+-----+-----+-----+-----+
| 1            | DD 2650 XY | ACX3568  | Hitam  | Ibrahim | Afdal    | 50000         |
| 3            | B 1611 QC  | LSQ1112  | Silver | Baim    | Anty     | 50000         |
| 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** adalah sebuah perintah query yang tugas memilih.

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

Kesimpulan :

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

IN+AND

Setelah mempelajari materi AND kita bisa menggabungkan IN dan AND untuk mendapat hasil yang lebih luas karna akan memiliki dua kondisi, untuk contoh kode programnya akan seperti dibawah ini :

```
Select * from mobil  
Where warna IN ("Hitam","Silver")  
And harga_rental = 50000;
```

Dan hasilnya akan seperti berikut :


```
MariaDB [rental_nero]> select * from mobil
-> where warna in ('hitam','silver')
-> and harga_rental = 50000;
```

id_pelanggan	no_plat	no_mesin	warna	pemilik	peminjam	harga_rental
1	DD 2650 XY	ACX3568	Hitam	Ibrahim	Afdal	50000
3	B 1611 QC	LSQ1112	Silver	Baim	Anty	50000

2 rows in set (0.001 sec)

Analisis: >

SELECT adalah sebuah perintah query yang tugas memilih.

- **FROM** adalah sebuah perintah query yang tugas nya untuk pemanggilan dari.
- **WHERE** adalah sebuah perintah query yang tugas untuk menentukan pemanggilan yang mana ditujukannya.
- **DATA_MOBIL** adalah sebuah nama tabel.
- **WARNA** adalah sebuah nama kolom.
- **HITAM DAN SILVER** adalah nama data pada sebuah kolom.
- **IN** adalah perintah query yang bertugas menentukan seluruh data yang sesuai dengan kondisi ditentukan, menghindari penulisan berulang query dengan kondisi yang sama.
- **AND** adalah sebuah perintah query DAN menunjukkan bahwa pemilihan data nya lebih dari 1
- **HARGA_RENTAL** adalah sebuah nama kolom.
- 6- **50000** adalah nama data pada sebuah kolom

Kesimpulan :

Query ini akan menampilkan data dimana mobil tersebut berwarna "Hitam","Silver" dan memiliki harga_rental = "50000", Seperti pada contoh di atas.

IN+OR

Selanjutnya kita akan mempelajari query IN+OR untuk contoh kodenya akan seperti berikut :

```
Select * from mobil
Where warna IN ("Hitam","Silver")
or harga_rental = 50000;
```

Dan hasilnya akan seperti berikut :

```
MariaDB [rental_nero]> select * from mobil
-> where warna in ('hitam','silver')
-> or harga_rental = 50000;
+-----+-----+-----+-----+-----+-----+-----+
| id_pelanggan | no_plat   | no_mesin | warna  | pemilik | peminjam | harga_rental |
+-----+-----+-----+-----+-----+-----+-----+
| 1 | DD 2650 XY | ACX3568 | Hitam  | Ibrahim | Afdal    | 50000 |
| 3 | B 1611 QC  | LSQ1112 | Silver | Baim    | Anty     | 50000 |
| 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: >

SELECT adalah sebuah perintah query yang tugas memilih.

- **FROM** adalah sebuah perintah query yang tugas nya untuk pemanggilan dari.
- **WHERE** adalah sebuah perintah query yang tugas untuk menentukan pemanggilan yang mana ditujukannya.
- **DATA_MOBIL** adalah sebuah nama tabel.
- **HARGA_RENTAL** adalah sebuah nama kolom.

- **50000** adalah nama data pada sebuah kolom
- **WARNA** adalah sebuah nama kolom.
- **HITAM DAN SILVER** adalah nama data pada sebuah kolom.
- **IN** adalah perintah query yang bertugas menentukan seluruh data yang sesuai dengan kondisi ditentukan, menghindari penulisan berulang query dengan kondisi yang sama.
- **OR** adalah perintah yang bertugas menentukan data dengan format "atau".

Kesimpulan :

Query ini akan mengambil data mobil yang berwarna "Hitam","Silver" atau yang memiliki harga rental = 50000.

IN + AND + OPERATOR

Selanjutnya kita akan menggabungkan antara IN, AND dan Operator. untuk kode program nya akan terlihat seperti berikut :

```
Select * from mobil
Where warna IN ("Hitam","Silver")
or harga_rental > 50000;
```

Dan hasilnya akan seperti berikut :

```
MariaDB [rental_nero]> select * from mobil
-> where warna IN ('Hitam','Silver')
-> And harga_rental < 100000;
```

id_pelanggan	no_plat	no_mesin	warna	pemilik	peminjam	harga_rental
1	DD 2650 XY	ACX3568	Hitam	Ibrahim	Afdal	50000
3	B 1611 QC	LSQ1112	Silver	Baim	Anty	50000

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

Analisis: >

SELECT adalah sebuah perintah query yang tugas memilih.

- **FROM** adalah sebuah perintah query yang tugas nya untuk pemanggilan dari.
- **WHERE** adalah sebuah perintah query yang tugas untuk menentukan pemanggilan yang mana ditujukannya.
- **DATA_MOBIL** adalah sebuah nama tabel.
- **WARNA** adalah sebuah nama kolom.
- **HITAM DAN SILVER** adalah nama data pada sebuah kolom.
- **IN** adalah perintah query yang bertugas menentukan seluruh data yang sesuai dengan kondisi ditentukan, menghindari penulisan berulang query dengan kondisi yang sama.
- **AND** adalah sebuah perintah query DAN menunjukkan bahwa pemilihan data nya lebih dari 1
- **HARGA_RENTAL** adalah sebuah nama kolom.
- **50000** adalah nama data pada sebuah kolom
 - > ialah sebuah operator yang bertugas untuk mengambil data khusus "di atas" 50000.

Kesimpulan :

Jadi query ini akan mengambil data dengan warna "Hitam","Silver" dan mobil dengan harga rental di atas 50000.

LIKE

1. MENCARI AWALAN

Struktur :

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

Contoh :

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

Hasil :

```
MariaDB [rental_nero]> select * from daftar_mobil  
-> where pemilik like 'ib%';
```

id_pelanggan	no_plat	no_mesin	warna	pemilik	peminjam	deadline	harga_rental
1	DD 2650 XY	ACX3568	Hitam	Ibrahim	Afdal	2024-04-24	50000
2	DD 2440 AX	BCS1120	Merah	Ibrahim	Elia	2024-04-24	100000
4	DD 2901 JK	UQL1029	Hitam	Ibe	NULL	NULL	150000
5	DD 2210 LS	CJH1011	Hitam	Ibe	NULL	NULL	100000

```
4 rows in set (0.003 sec)
```

Analisis :

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

Kesimpulan :

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

2. MENCARI AKHIRAN

Struktur :

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

Contoh :

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

Hasil :

```
MariaDB [rental_nero]> select * from daftar_mobil  
-> where pemilik like '%M';
```

id_pelanggan	no_plat	no_mesin	warna	pemilik	peminjam	deadline	harga_rental
1	DD 2650 XY	ACX3568	Hitam	Ibrahim	Afdal	2024-04-24	50000
2	DD 2440 AX	BCS1120	Merah	Ibrahim	Elia	2024-04-24	100000
3	B 1611 QC	LSQ1112	Silver	Baim	Anty	2024-04-24	50000

3 rows in set (0.000 sec)

Analisis :

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

Kesimpulan :

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

3. MENCARI AWALAN DAN AKHIRAN

Struktur :

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

Contoh :

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

Hasil :

```
MariaDB [rental_nero]> select * from daftar_mobil  
-> where pemilik like 'B%M';
```

id_pelanggan	no_plat	no_mesin	warna	pemilik	peminjam	deadline	harga_rental
3	B 1611 QC	LSQ1112	Silver	Baim	Anty	2024-04-24	50000

1 row in set (0.000 sec)

Analisis :

- **SELECT** adalah sebuah perintah query yang tugas memilih.
- **FROM** adalah sebuah perintah query yang tugas nya untuk pemanggilan dari.
- **WHERE** adalah sebuah perintah query yang tugas untuk menentukan pemanggilan yang mana ditujukannya.
- **DATA_MOBIL** adalah sebuah nama tabel.
- **PEMILIK** adalah sebuah nama kolom.
- **LIKE** adalah perintah query yang bertugas menentukan hasil berdasarkan pola teks tertentu.
- **B%M** adalah sebuah pola awalan dan akhiran tertentu yang ingin ditampilkan.

Kesimpulan :

query diatas adalah perintah untuk menampilkan data pada kolom pemilik yang berpola b%m artinya data dengan awalan B dan akhiran M pada kolom pemilik yang akan ditampilkan.

4. MENCARI BERDASARKAN TOTAL KARAKTER

Struktur :

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


Contoh :

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

Hasil :

```
MariaDB [rental_nero]> select * from daftar_mobil  
-> where pemilik like 'i__';
```

id_pelanggan	no_plat	no_mesin	warna	pemilik	peminjam	deadline	harga_rental
4	DD 2901 JK	UQL1029	Hitam	Ibe	NULL	NULL	150000
5	DD 2210 LS	CJH1011	Hitam	Ibe	NULL	NULL	100000

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

Analisis :

- **SELECT** adalah sebuah perintah query yang tugas memilih.
- **FROM** adalah sebuah perintah query yang tugas nya untuk pemanggilan dari.
- **WHERE** adalah sebuah perintah query yang tugas untuk menentukan pemanggilan yang mana ditujukannya.
- **DATA_MOBIL** adalah sebuah nama tabel.
- **PEMILIK** adalah sebuah nama kolom.
- **LIKE** adalah perintah query yang bertugas menentukan hasil berdasarkan pola teks tertentu.
- **'i__'** adalah pola karakter tertentu yang ingin ditampilkan.

Kesimpulan :

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

5. KOMBINASI

Struktur :

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

Contoh :

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

Hasil :

```
MariaDB [rental_nero]> select * from daftar_mobil  
-> WHERE PEMILIK LIKE '__R%';
```

id_pelanggan	no_plat	no_mesin	warna	pemilik	peminjam	deadline	harga_rental
1	DD 2650 XY	ACX3568	Hitam	Ibrahim	Afdal	2024-04-24	50000
2	DD 2440 AX	BCS1120	Merah	Ibrahim	Elia	2024-04-24	100000
6	DD 9056 DT	KJH1987	KUNING	NERO	NULL	NULL	200000

```
3 rows in set (0.000 sec)
```

Analisis :

- **SELECT** adalah sebuah perintah query yang tugas memilih.
- **FROM** adalah sebuah perintah query yang tugas nya untuk pemanggilan dari.
- **WHERE** adalah sebuah perintah query yang tugas untuk menentukan pemanggilan yang mana ditujukannya.
- **DATA_MOBIL** adalah sebuah nama tabel.

- **PEMILIK** adalah sebuah nama kolom.
- **LIKE** adalah perintah query yang bertugas menentukan hasil berdasarkan pola teks tertentu.
- **'__R%'** adalah pola karakter tertentu dari salah satu data yang ingin ditampilkan.

Kesimpulan :

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

6. NOT LIKE

Struktur :

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

Contoh :

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

Hasil :

```
MariaDB [rental_nero]> SELECT * FROM DAFTAR_MOBIL WHERE PEMINJAM NOT LIKE 'A%';
+-----+-----+-----+-----+-----+-----+-----+-----+
| id_pelanggan | no_plat   | no_mesin | warna | pemilik | peminjam | deadline | harga_rental |
+-----+-----+-----+-----+-----+-----+-----+-----+
|            2 | DD 2440 AX | BCS1120  | Merah | Ibrahim | Elia     | 2024-04-24 |          100000 |
+-----+-----+-----+-----+-----+-----+-----+-----+
1 row in set (0.000 sec)
```

Analisis :

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

- **WHERE** adalah sebuah perintah query yang tugas untuk menentukan pemanggilan yang mana ditujukannya.
- **DATA_MOBIL** adalah sebuah nama tabel.
- **PEMINJAM** adalah sebuah nama kolom.
- **NOT LIKE** adalah perintah query yang bertugas menentukan hasil yang tidak sesuai berdasarkan pola teks tertentu.
- **'A%'** adalah pola yang tidak boleh ditampilkan.

Kesimpulan :

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

NULL & NOT NULL

1. MENCARI DATA KOSONG

Struktur :

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

Contoh :

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

Hasil :

```
MariaDB [rental_nero]> SELECT * FROM DAFTAR_MOBIL WHERE PEMINJAM IS NOT NULL;
```

id_pelanggan	no_plat	no_mesin	warna	pemilik	peminjam	deadline	harga_rental
1	DD 2650 XY	ACX3568	Hitam	Ibrahim	Afdal	2024-04-24	50000
2	DD 2440 AX	BCS1120	Merah	Ibrahim	Elia	2024-04-24	100000
3	B 1611 QC	LSQ1112	Silver	Baim	Anty	2024-04-24	50000

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

Analisis :

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

Kesimpulan :

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

2. NOT NULL

Struktur :

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

Contoh :

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

Hasil :

```
MariaDB [rental_nero]> SELECT * FROM DAFTAR_MOBIL WHERE PEMINJAM IS NULL;
+-----+-----+-----+-----+-----+-----+-----+-----+
| id_pelanggan | no_plat | no_mesin | warna | pemilik | peminjam | deadline | harga_rental |
+-----+-----+-----+-----+-----+-----+-----+-----+
|          4 | DD 2901 JK | UQL1029 | Hitam | Ibe | NULL | NULL | 150000 |
|          5 | DD 2210 LS | CJH1011 | Hitam | Ibe | NULL | NULL | 100000 |
|          6 | DD 9056 DT | KJH1987 | KUNING | NERO | NULL | NULL | 200000 |
|          7 | DD 2536 TP | WN96K5 | Hitam | Rahmat | NULL | Ready | 250000 |
+-----+-----+-----+-----+-----+-----+-----+-----+
4 rows in set (0.000 sec)
```

Analisis :

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

Kesimpulan :

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

ORDER BY LIMIT

1. ASC

Struktur :

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

Contoh :

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

Hasil :

```
MariaDB [rental_nero]> SELECT * FROM DAFTAR_MOBIL ORDER BY PEMILIK ASC;
```

id_pelanggan	no_plat	no_mesin	warna	pemilik	peminjam	deadline	harga_rental
3	B 1611 QC	LSQ1112	Silver	Baim	Anty	2024-04-24	50000
4	DD 2901 JK	UQL1029	Hitam	Ibe	NULL	NULL	150000
5	DD 2210 LS	CJH1011	Hitam	Ibe	NULL	NULL	100000
1	DD 2650 XY	ACX3568	Hitam	Ibrahim	Afdal	2024-04-24	50000
2	DD 2440 AX	BCS1120	Merah	Ibrahim	Elia	2024-04-24	100000
6	DD 9056 DT	KJH1987	KUNING	NERO	NULL	NULL	200000
7	DD 2536 TP	WN96K5	Hitam	Rahmat	NULL	Ready	250000

7 rows in set (0.001 sec)

Analisis :

- **SELECT** adalah sebuah perintah query yang tugas memilih.
- **FROM** adalah sebuah perintah query yang tugas nya untuk pemanggilan dari.
- **ORDER BY** adalah sebuah perintah query yang untuk mengurutkan pemanggilan berdasarkan nilai.

- **DATA_MOBIL** adalah sebuah nama tabel.
- **PEMINJAM** adalah sebuah nama kolom.
- **ASC** adalah perintah query yang bertugas menampilkan suatu data secara berurutan ke atas.

Kesimpulan :

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

2. DESC

Struktur :

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

Contoh :

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

Hasil :

id_pelanggan	no_plat	no_mesin	warna	pemilik	peminjam	deadline	harga_rental
7	DD 2536 TP	WN96K5	Hitam	Rahmat	NULL	Ready	250000
6	DD 9056 DT	KJH1987	KUNING	NERO	NULL	NULL	200000
1	DD 2650 XY	ACX3568	Hitam	Ibrahim	Afdal	2024-04-24	50000
2	DD 2440 AX	BCS1120	Merah	Ibrahim	Elia	2024-04-24	100000
4	DD 2901 JK	UQL1029	Hitam	Ibe	NULL	NULL	150000
5	DD 2210 LS	CJH1011	Hitam	Ibe	NULL	NULL	100000
3	B 1611 QC	LSQ1112	Silver	Baim	Anty	2024-04-24	50000

7 rows in set (0.001 sec)

Analisis :

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

Kesimpulan :

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

3. LIMIT

Struktur :

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

Contoh :

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

Hasil :

```
MariaDB [rental_nero]> SELECT * FROM DAFTAR_MOBIL ORDER BY PEMILIK ASC;
```

id_pelanggan	no_plat	no_mesin	warna	pemilik	peminjam	deadline	harga_rental
3	B 1611 QC	LSQ1112	Silver	Baim	Anty	2024-04-24	50000
4	DD 2901 JK	UQL1029	Hitam	Ibe	NULL	NULL	150000
5	DD 2210 LS	CJH1011	Hitam	Ibe	NULL	NULL	100000
1	DD 2650 XY	ACX3568	Hitam	Ibrahim	Afdal	2024-04-24	50000
2	DD 2440 AX	BCS1120	Merah	Ibrahim	Elia	2024-04-24	100000
6	DD 9056 DT	KJH1987	KUNING	NERO	NULL	NULL	200000
7	DD 2536 TP	WN96K5	Hitam	Rahmat	NULL	Ready	250000

7 rows in set (0.001 sec)

Analisis :

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

KESIMPULAN:

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

3. LIMIT

Struktur :

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

Contoh :

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

Hasil :

```
MariaDB [rental_nero]> SELECT * FROM DAFTAR_MOBIL WHERE WARNA="HITAM" ORDER BY HARGA_RENTAL ASC LIMIT 2;
```

id_pelanggan	no_plat	no_mesin	warna	pemilik	peminjam	deadline	harga_rental
1	DD 2650 XY	ACX3568	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** adalah sebuah perintah query yang tugas memilih.
- **FROM** adalah sebuah perintah query yang tugas nya untuk pemanggilan dari.
- **ORDER BY** adalah sebuah perintah query yang untuk mengurutkan pemanggilan berdasarkan nilai.
- **DATA_MOBIL** adalah sebuah nama tabel.
- **WHERE** adalah sebuah perintah query yang tugas untuk menentukan pemanggilan yang mana ditujukannya.
- **PEMINJAM & WARNA** adalah sebuah nama kolom.
- **ASC** adalah perintah query yang bertugas menampilkan suatu data secara berurutan ke atas.
- **HITAM** adalah nama data pada sebuah kolom.
- **LIMIT 2** adalah perintah untuk membatasi jumlah baris yang ingin ditampilkan.

Kesimpulan :

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

DISTINCT

Struktur :

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

Contoh :

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

Hasil :

```
MariaDB [rental_nero]> SELECT DISTINCT(PEMILIK) FROM DAFTAR_MOBIL;
+-----+
| PEMILIK |
+-----+
| Ibrahim |
| Baim    |
| Ibe     |
| NERO    |
| Rahmat  |
+-----+
5 rows in set (0.001 sec)
```

Analisis :

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

Kesimpulan :

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

CONCAT, CONCAT_WS, AS

1. CONCAT

Struktur :

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

Contoh :

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

Hasil :

```
MariaDB [rental_nero]> SELECT CONCAT(PEMILIK, WARNA) FROM DAFTAR_MOBIL;
+-----+
| CONCAT(PEMILIK, WARNA) |
+-----+
| IbrahimHitam           |
| IbrahimMerah           |
| BaimSilver              |
| IbeHitam                |
| IbeHitam                |
| NEROKUNING              |
| RahmatHitam             |
+-----+
7 rows in set (0.000 sec)
```

Analisis :

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

KESIMPULAN:

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

2. CONCAT_WS

Struktur :

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

Contoh :

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

Hasil :

```

MariaDB [rental_nero]> SELECT CONCAT_WS ("-",NO_PLAT,NO_MESIN,ID_PELANGGAN) FROM DAFTAR_MOBIL;
+-----+
| CONCAT_WS ("-",NO_PLAT,NO_MESIN,ID_PELANGGAN) |
+-----+
| DD 2650 XY-ACX3568-1 |
| DD 2440 AX-BCS1120-2 |
| B 1611 QC-LSQ1112-3 |
| DD 2901 JK-UQL1029-4 |
| DD 2210 LS-CJH1011-5 |
| DD 9056 DT-KJH1987-6 |
| DD 2536 TP-WN96K5-7 |
+-----+
7 rows in set (0.000 sec)

```

Analisis :

- **SELECT** adalah perintah query yang tugas nya memilih.
- **CONCAT_WS** adalah perintah untuk menggabungkan nilai nilai sebuah kolom dan memiliki pemisah tertentu untuk ditampilkan.
- **NO_PLAT,NO_MESIN,ID_MOBIL** adalah sebuah nama kolom.
- **"-"** adalah pemisah dari kolom yang di satukan.
- **FROM** adalah sebuah pemanggilan dari data yang ditujukan.
- **DATA_MOBIL** adalah sebuah nama tabel.

Kesimpulan :

query di atas adalah perintah untuk menggabungkan kolom no_plat,no_mesin,id_mobil menggunakan "-" sebagai pemisah untuk ditampilkan.

3. AS

Struktur :

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

Contoh :

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

Hasil :

```
MariaDB [rental_nero]> SELECT CONCAT_WS ("+",PEMILIK,PEMINJAM) FROM DAFTAR_MOBIL;
+-----+
| CONCAT_WS ("+",PEMILIK,PEMINJAM) |
+-----+
| Ibrahim+Afdal                     |
| Ibrahim+Elia                      |
| Baim+Anty                         |
| Ibe                               |
| Ibe                               |
| NERO                             |
| Rahmat                           |
+-----+
7 rows in set (0.000 sec)
```

Analisis :

- **SELECT** adalah perintah query yang tugas nya memilih.
- **CONCAT_WS** adalah perintah untuk menggabungkan nilai nilai sebuah kolom dan memiliki pemisah tertentu untuk ditampilkan.
- **PEMILIK,PEMINJAM** adalah sebuah nama kolom.
- **"+"** adalah penghubung dari kolom yang di satukan.
- **FROM** adalah sebuah pemanggilan dari data yang ditujukan.
- **DATA_MOBIL** adalah sebuah nama tabel.

Kesimpulan :

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

VIEW

1. MEMBUAT TABEL VIRTUAL

Struktur :

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

Contoh :

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

Hasil :

```
MariaDB [rental_nero]> CREATE VIEW INFO_NO_PLAT AS
-> SELECT ID_PELANGGAN,NO_PLAT,PEMILIK,PEMINJAM
-> FROM DAFTAR_MOBIL
-> WHERE PEMILIK="IBRAHIM";
Query OK, 0 rows affected (0.014 sec)
```

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

```
+-----+
| Tables_in_rental_nero |
+-----+
| daftar_mobil          |
| info_no_plat          |
| pelanggan             |
+-----+
3 rows in set (0.003 sec)
```

Analisis :

- **SELECT** adalah perintah query yang tugas nya memilih.
- **CREATE VIEW** adalah perintah untuk membuat sebuah tabel virtual
- **ID_MOBIL,NO_PLAT,PEMILIK,PEMINJAM** adalah sebuah nama kolom.
- **FROM** adalah sebuah pemanggilan dari data yang ditujukan.
- **DATA_MOBIL** adalah sebuah nama tabel.
- **WHERE** adalah sebuah perintah query yang tugas untuk menentukan pemanggilan yang mana ditujukannya.
- **IBRAHIM** adalah nama data pada suatu kolom

Kesimpulan :

query diatas adalah perintah untuk membuat sebuah tabel virtual yang ditampilkan adalah kolom id_mobil,no_plat,pemilik,peminjam yang sesuai dengan data ibrahim pada kolom pemilik.

2. MENAMPILKAN TABEL VIRTUAL

Struktur :

```
SELECT * FROM NAMA_TABEL_VIRTUAL;
```

Contoh :

```
SELECT * FROM INFO_NO_PLAT;
```

Hasil :

```
MariaDB [rental_nero]> SELECT * FROM INFO_NO_PLAT;
+-----+-----+-----+-----+
| ID_PELANGGAN | 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** adalah perintah query yang tugas nya memilih.
- **FROM** adalah perintah query dari.
- **INFO_NO_PLAT** adalah sebuah nama tabel virtual yang telah dibuat

Kesimpulan :

query di atas adalah perintah untuk menampilkan tabel virtual yang dibuat tadi.

3. MENGHAPUS TABEL VIRTUAL

Struktur :

```
DROP VIEW NAMA_TABEL_VIRTUAL;
```

Contoh :

```
DROP VIEW INFO_NO_PLAT;
```

Hasil :

```
MariaDB [rental_nero]> SELECT * FROM INFO_NO_PLAT;
+-----+-----+-----+-----+
| ID_PELANGGAN | NO_PLAT   | PEMILIK | PEMINJAM |
+-----+-----+-----+-----+
|           1 | DD 2650 XY | Ibrahim | Afdal    |
|           2 | DD 2440 AX | Ibrahim | Elia     |
+-----+-----+-----+-----+
2 rows in set (0.002 sec)

MariaDB [rental_nero]> DROP VIEW INFO_NO_PLAT;
Query OK, 0 rows affected (0.001 sec)

MariaDB [rental_nero]> SHOW TABLES;
+-----+
| Tables_in_rental_nero |
+-----+
| daftar_mobil          |
| pelanggan             |
+-----+
2 rows in set (0.002 sec)
```

Analisis :

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

AGREGASI

1. MENGHITUNG TOTAL

Struktur :

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

Contoh :

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

Hasil :

```
MariaDB [rental_nero]> SELECT SUM(HARGA_RENTAL) AS TOTAL_HARGA FROM DAFTAR_MOBIL;
+-----+
| TOTAL_HARGA |
+-----+
|      900000 |
+-----+
1 row in set (0.000 sec)
```

Analisis :

- **SELECT** adalah perintah untuk memilih suatu kolom
- **SUM** adalah perintah untuk menghitung semua total jumlah
- **HARGA_RENTAL** adalah sebuah nama kolom

- **AS** adalah perintah untuk memberikan nama lain dari perintah tersebut.
- **TOTAL_HARGA** nama lain dari kolom pada perintah tersebut
- **DATA_MOBIL** adalah sebuah nama tabel.

Kesimpulan :

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

2. MENGHITUNG JUMLAH BARIS

Struktur :

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

Contoh :

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

Hasil :

```
MariaDB [rental_nero]> SELECT COUNT(PEMILIK) AS JUMLAH_PEMILIK FROM DAFTAR_MOBIL;
+-----+
| JUMLAH_PEMILIK |
+-----+
|                7 |
+-----+
1 row in set (0.000 sec)
```

Analisis :

- **SELECT** adalah perintah untuk memilih suatu kolom

- **COUNT** adalah perintah untuk menghitung semua jumlah baris.
- **JUMLAH_PEMILIK** adalah sebuah nama kolom
- **AS** adalah perintah untuk memberikan nama lain dari perintah tersebut.
- **PEMILIK** nama lain dari kolom pada perintah tersebut
- **DATA_MOBIL** adalah sebuah nama tabel.

Kesimpulan :

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

3. MENGHITUNG NILAI TERENDAH

Struktur :

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

Contoh :

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

Hasil :

```
MariaDB [rental_nero]> SELECT MIN(HARGA_RENTAL) AS MINIMUM FROM DAFTAR_MOBIL;
+-----+
| MINIMUM |
+-----+
|  50000  |
+-----+
1 row in set (0.002 sec)
```

Analisis :

- **SELECT** adalah perintah untuk memilih suatu kolom
- **MIN** adalah perintah untuk menghitung jumlah data yang paling terkecil.
- **HARGA_RENTAL** adalah sebuah nama kolom
- **AS** adalah perintah untuk memberikan nama lain dari perintah tersebut.
- **MINIMUM** nama lain dari kolom pada perintah tersebut
- **DATA_MOBIL** adalah sebuah nama tabel.

Kesimpulan :

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

4. MENGHITUNG NILAI TERTINGGI

Struktur :

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

Contoh :

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

Hasil :

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


Analisis :

- **SELECT** adalah perintah untuk memilih suatu kolom
- **MAX** adalah perintah untuk menghitung data yang paling tinggi.
- **HARGA_RENTAL** adalah sebuah nama kolom
- **AS** adalah perintah untuk memberikan nama lain dari perintah tersebut.
- **MAXIMUM** nama lain dari kolom pada perintah tersebut
- **DATA_MOBIL** adalah sebuah nama tabel.

Kesimpulan :

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

5. MENAMPILKAN NILAI RATA RATA

Struktur :

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

Contoh :

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

Hasil :

```
MariaDB [rental_nero]> SELECT AVG(HARGA_RENTAL) AS RERATA FROM DAFTAR_MOBIL;  
+-----+  
| RERATA |  
+-----+  
| 128571.4286 |  
+-----+  
1 row in set (0.001 sec)
```

Analisis :

- **SELECT** adalah perintah untuk memilih suatu kolom
- **AVG** adalah perintah untuk menghitung jumlah rata rata
- **HARGA_RENTAL** adalah sebuah nama kolom
- **AS** adalah perintah untuk memberikan nama lain dari perintah tersebut.
- **RERATA** nama lain dari kolom pada perintah tersebut
- **DATA_MOBIL** adalah sebuah nama tabel.

Kesimpulan :

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