

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 :

AND

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

```
SELECT id_guru,nama_depan FROM tabel_guru WHERE id_guru="3" AND na="Rusdyansyar";
```

Dan hasilnya akan seperti berikut :

```
MariaDB [rental_nafan]> SELECT warna,pemilik FROM data_mobil WHERE warna="Hitam" AND pemilik="ibrahim";
+-----+-----+
| warna | pemilik |
+-----+-----+
| Hitam | Ibrahim |
+-----+-----+
1 row in set (0.020 sec)
```

OR

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

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

Dan hasilnya akan seperti berikut :

```
MariaDB [rental_nafan]> SELECT warna,pemilik FROM data_mobil WHERE warna="Hitam" OR pemilik="ibrahim";
+-----+-----+
| warna | pemilik |
+-----+-----+
| Hitam | Ibrahim |
| Hitam | Ibe     |
| Hitam | Ibe     |
+-----+-----+
3 rows in set (0.006 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 tabel_guru WHERE tanggal_lahir BETWEEN 1982-06-29 AND 2000-09-21;
```

Dan hasilnya akan seperti berikut :

```
MariaDB [rental_nafan]> SELECT * FROM data_mobil WHERE harga_rental BETWEEN 100000 AND 200000;
+-----+-----+-----+-----+-----+-----+-----+
| 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)
```

NOT BETWEEN

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

```
SELECT * FROM data_mobil WHERE harga_rental NOT BETWEEN 100000 AND 200000;
```

Dan hasilnya akan seperti berikut :

```
MariaDB [rental_nafan]> SELECT * FROM data_mobil WHERE harga_rental NOT BETWEEN 100000 AND 200000;
+-----+-----+-----+-----+-----+-----+-----+
| id_mobil | no_plat | no_mesin | warna | pemilik | peminjam | harga_rental |
+-----+-----+-----+-----+-----+-----+-----+
| 1 | DD 2650 XY | ACX3568 | Hitam | Ibrahim | Afdal | 50000 |
| 3 | B 1611 QC | LSQ1112 | Silver | Baim | Anty | 50000 |
+-----+-----+-----+-----+-----+-----+-----+
2 rows in set (0.001 sec)
```

<=

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

```
MariaDB [rental_nafan]> SELECT * FROM data_mobil WHERE harga_rental <= 50000;
```

Dan hasilnya akan seperti berikut :

```
MariaDB [rental_nafan]> SELECT * FROM data_mobil WHERE harga_rental <= 50000;
+-----+-----+-----+-----+-----+-----+-----+
| id_mobil | no_plat | no_mesin | warna | pemilik | peminjam | harga_rental |
+-----+-----+-----+-----+-----+-----+-----+
| 1 | DD 2650 XY | ACX3568 | Hitam | Ibrahim | Afdal | 50000 |
| 3 | B 1611 QC | LSQ1112 | Silver | Baim | Anty | 50000 |
+-----+-----+-----+-----+-----+-----+-----+
2 rows in set (0.004 sec)
```

>=

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

```
MariaDB [rental_nafan]> SELECT * FROM data_mobil WHERE harga_rental >= 50000;
```

Dan hasilnya akan seperti berikut :

```
MariaDB [rental_nafan]> SELECT * FROM data_mobil WHERE harga_rental >= 50000;
+-----+-----+-----+-----+-----+-----+-----+
| id_mobil | no_plat | no_mesin | warna | pemilik | peminjam | harga_rental |
+-----+-----+-----+-----+-----+-----+-----+
| 1 | DD 2650 XY | ACX3568 | Hitam | Ibrahim | Afdal | 50000 |
| 2 | DD 2440 AX | BCS1120 | Merah | Ibrahim | Elia | 100000 |
| 3 | B 1611 QC | LSQ1112 | Silver | Baim | Anty | 50000 |
| 4 | DD 2901 JK | UQL1029 | Hitam | Ibe | NULL | 150000 |
| 5 | DD 2210 LS | CJH1011 | Hitam | Ibe | NULL | 100000 |
+-----+-----+-----+-----+-----+-----+-----+
5 rows in set (0.002 sec)
```

<> atau !=

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

```
MariaDB [rental_nafan]> SELECT * FROM data_mobil WHERE harga_rental <> 50000;
```

Dan hasilnya akan seperti berikut :

```
MariaDB [rental_nafan]> SELECT * FROM data_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 (0.002 sec)
```

Tantangan

Untuk tantangan saya akan mengambil nama pemilik "Ibrahim" dengan cara memanggilnya dengan syarat nomor pelatnya yaitu "DD 2440 AX" lalu hasilnya akan seperti berikut :

```
MariaDB [rental_nafan]> SELECT pemilik  
-> FROM data_mobil  
-> WHERE no_plat = 'DD 2440 AX';  
+-----+  
| pemilik |  
+-----+  
| Ibrahim |  
+-----+  
1 row in set (0.001 sec)
```

TANTANGAN LOGIN

Struktur

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

Contoh

```
SELECT nama FROM akun  
WHERE id = 1;
```

Hasil

```

MariaDB [nafan]> select * from users;
+-----+-----+-----+-----+
| id | nama          | username | password |
+-----+-----+-----+-----+
| 1  | M Nafan Nabil | admin    | 123      |
| 2  | Farhan        | user     | 123      |
+-----+-----+-----+-----+
2 rows in set (0.001 sec)

MariaDB [nafan]> Select nama From users WHERE id = 2;
+-----+
| nama |
+-----+
| Farhan |
+-----+
1 row in set (0.001 sec)

```

Analisis

- **SELECT** : digunakan untuk mengambil sebuah data pada tabel
- **nama** : nama dari kolom yang akan kita tampilkan
- **FROM akun** : merupakan nama tabel yang ingin kita ambil datanya
- **WHERE** : digunakan untuk menyaring data yang ada pada tabel
- **id = 2;** merupakan baris yang akan kita tampilkan yang memiliki id bernilai 2

Kesimpulan

perintah SQL tersebut adalah mengambil/menampilkan nama yang memiliki ID 2 dari tabel **akun** .

[i 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 di inginkan, keberagaman select ini mulai dari AND ,OR ,BETWEEN-AND ,NOT BETWEEN ,<= ,>= ,<> ATAU !=.

IN

Struktur Query

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

Contoh

```
SELECT * FROM mobil WHERE warna IN('Hitam', 'Merah');
```

Hasil

```
MariaDB [rental_nafan]> select * from mobil;
```

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

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



```
MariaDB [rental_nafan]> SELECT * FROM mobil WHERE warna IN('Hitam', 'Merah');
```

id_mobil	no_plat	no_mesin	warna	pemilik	peminjam	batas_peminjam	harga_rental
1	DD 2650 XY	ACX3568	Hitam	Ibrahim	Afdal	NULL	50000
2	DD 2440 AX	BCS1120	Merah	Ibrahim	Elia	NULL	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 (0.001 sec)
```

Analisis

- `SELECT * FROM` : Merupakan query awal untuk menampilkan sebuah tabel yang didalamnya berisikan kolom, di mana kolom tersebut menampung sebuah data.
- `mobil` nama dari tabel yang ingin kita tampilkan data-data nya.
- `WHERE` : digunakan untuk menyaring data yang ada pada tabel.
- `warna IN('Hitam', 'Merah');` : Query tersebut menampilkan sebuah data pada tabel dimana kolom warna yang bernilai Hitam dan Merah akan di tampilkan.

Kesimpulan

Perintah MySQL tersebut Menyaring sebuah data pada tabel mobil dimana akan hanya menampilkan warna Hitam dan Merah.

IN & AND

Struktur Query

```
SELECT * FROM [nama_tabel]
WHERE [nama_kolom] IN ('nilai')
AND [nama_kolom] = nilai;
```

Contoh

```
SELECT * FROM mobil
WHERE warna IN('Merah','silver')
AND harga_rental = 50000;
```

Hasil

```
MariaDB [rental_nafan]> select * from mobil;
```

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

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



```
MariaDB [rental_nafan]> SELECT * FROM mobil
-> WHERE warna IN('Merah','silver')
-> AND harga_rental = 50000;
```

id_mobil	no_plat	no_mesin	warna	pemilik	peminjam	batas_peminjam	harga_rental
3	B 1611 QC	LSQ1112	Silver	Baim	Anty	NULL	50000

```
1 row in set (0.001 sec)
```

Analisis

- `SELECT * FROM` : Merupakan query awal untuk menampilkan sebuah tabel yang didalamnya berisikan kolom, di mana kolom tersebut menampung sebuah data.
- `mobil` nama dari tabel yang ingin kita tampilkan data-data nya.
- `WHERE` : digunakan untuk menyaring data yang ada pada tabel.
- `warna IN('Merah','silver') AND harga_rental = 50000;` : Query tersebut mencari dimana kolom warna yang hanya memiliki nilai Merah atau silver, dan kolom harga_rental bernilai 50000 yang akan tampil.

Kesimpulan

Menggunakan operator AND, memastikan bahwa kedua kondisi/kolom harus benar untuk memilih baris yang akan diambil.

IN & OR

Struktur Query

```
SELECT * FROM [nama_tabel]
WHERE [nama_kolom] IN ('nilai')
OR [nama_kolom] = nilai;
```

Contoh

```
SELECT * FROM mobil
WHERE warna IN('Merah','silver')
OR harga_rental = 150000;
```

Hasil

```
MariaDB [rental_nafan]> select * from mobil;
```

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

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



```
MariaDB [rental_nafan]> SELECT * FROM mobil
-> WHERE warna IN('Merah','silver')
-> OR harga_rental = 150000;
```

id_mobil	no_plat	no_mesin	warna	pemilik	peminjam	batas_peminjam	harga_rental
2	DD 2440 AX	BCS1120	Merah	Ibrahim	Elia	NULL	100000
3	B 1611 QC	LSQ1112	Silver	Baim	Anty	NULL	50000
4	DD 2901 JK	UQL1029	Hitam	Ibe	NULL	NULL	150000

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

Analisis

- `SELECT * FROM` : Merupakan query awal untuk menampilkan sebuah tabel yang didalamnya berisikan kolom, di mana kolom tersebut menampung sebuah data.
- `mobil` nama dari tabel yang ingin kita tampilkan data-data nya.
- `WHERE` : digunakan untuk menyaring data yang ada pada tabel.

- `warna IN('Merah','silver') OR harga_rental = 150000;` : query tersebut mencari data pada kolom warna bernilai Merah dan silver atau kolom harga_rental yang bernilai 150000 yang akan ditampilkan.

Kesimpulan

Menggunakan operator logika OR memastikan bahwa salah satu dari 2 kondisi/kolom tersebut memenuhi syarat akan diambil.

IN + AND + OPERATOR

LEBIH BESAR

Struktur Query

```
SELECT * FROM [nama_tabel]
WHERE [nama_kolom] IN ('nilai1','nilai2')
AND [nama_kolom] > nilai;
```

Contoh

```
SELECT * FROM mobil
WHERE warna IN('Merah','silver')
AND harga_rental > 50000;
```

Hasil

```
MariaDB [rental_nafan]> select * from mobil;
```

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

```
5 rows in set (0.001 sec)

MariaDB [rental_nafan]> SELECT * FROM mobil
-> WHERE warna IN('Merah','silver')
-> AND harga_rental > 50000;
```

id_mobil	no_plat	no_mesin	warna	pemilik	peminjam	batas_peminjam	harga_rental
2	DD 2440 AX	BCS1120	Merah	Ibrahim	Elia	NULL	100000

```
1 row in set (0.001 sec)
```

Analisis

- `SELECT * FROM` : Merupakan query awal untuk menampilkan sebuah tabel yang didalamnya berisikan kolom, di mana kolom tersebut menampung sebuah data.
- `mobil` nama dari tabel yang ingin kita tampilkan data-data nya.
- `WHERE` : digunakan untuk menyaring data yang ada pada tabel.
- `warna IN('Merah','silver') AND harga_rental > 50000`; : query tersebut mencari data pada kolom warna yang bernilai Merah atau silver dengan kriteria harga_rental berada di atas 50000.

Kesimpulan

Query tersebut mencari data pada kolom warna yang bernilai Merah atau silver dan pada saat yang sama kolom harga_rental harus lebih besar dari 50000.

LEBIH KECIL

Struktur Query

```
SELECT * FROM [nama_tabel]
WHERE [nama_kolom] IN ('nilai1','nilai2')
AND [nama_kolom] > nilai;
```

Contoh

```
SELECT * FROM mobil
WHERE warna IN('Merah','silver')
AND harga_rental < 1000000;
```

Hasil

```
MariaDB [rental_nafan]> select * from mobil;
```

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

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



```
MariaDB [rental_nafan]> SELECT * FROM mobil
-> WHERE warna IN('Merah','silver')
-> AND harga_rental < 1000000;
```

id_mobil	no_plat	no_mesin	warna	pemilik	peminjam	batas_peminjam	harga_rental
2	DD 2440 AX	BCS1120	Merah	Ibrahim	Elia	NULL	100000
3	B 1611 QC	LSQ1112	Silver	Baim	Anty	NULL	50000

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

Analisis

- `SELECT * FROM` : Merupakan query awal untuk menampilkan sebuah tabel yang didalamnya berisikan kolom, di mana kolom tersebut menampung sebuah data.
- `mobil` nama dari tabel yang ingin kita tampilkan data-data nya.
- `WHERE` : digunakan untuk menyaring data yang ada pada tabel.
- `warna IN('Merah','silver') AND harga_rental < 50000`; : query tersebut mencari data pada kolom warna yang bernilai Merah atau silver dengan kriteria harga_rental berada dibawah 100000.

Kesimpulan

Query tersebut mencari data pada kolom warna yang bernilai Merah atau silver dan pada saat yang sama kolom harga_rental berada dibawah 100000.

LIKE

MENCARI AWALAN

Struktur Query

```
SELECT * FROM [nama_tabel]
```

```
WHERE [nama_kolom] LIKE 'nilai';
```

Contoh

```
SELECT * FROM mobil
WHERE pemilik LIKE 'i%';
```

Hasil

```
MariaDB [rental_nafan]> select * from mobil;
```

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

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

```
MariaDB [rental_nafan]> SELECT * FROM mobil
-> WHERE pemilik LIKE 'i%';
```

id_mobil	no_plat	no_mesin	warna	pemilik	peminjam	batas_peminjam	harga_rental
1	DD 2650 XY	ACX3568	Hitam	Ibrahim	Afdal	NULL	50000
4	DD 2901 JK	UQL1029	Hitam	Ibe	NULL	NULL	150000
5	DD 2210 LS	CJH1011	Hitam	Ibe	NULL	NULL	100000

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

Analisis

- `SELECT * FROM` : Merupakan query awal untuk menampilkan sebuah tabel yang didalamnya berisikan kolom, di mana kolom tersebut menampung sebuah data.
- `mobil` nama dari tabel yang ingin kita tampilkan data-data nya.
- `WHERE` : digunakan untuk menyaring data yang ada pada tabel.
- `pemilik LIKE 'T%'` : query tersebut mencari data pada kolom pemilik yang berawalan kata huruf T akan diambil.

Kesimpulan

`SELECT * FROM mobil WHERE pemilik LIKE 'T%'` : Query tersebut mencari data pada tabel mobil yang memiliki awalan kata huruf T pada kolom pemilik.

MENCARI AKHIRAN

Struktur Query

```
SELECT * FROM [nama_tabel]
WHERE [nama_kolom] LIKE 'nilai%';
```

Contoh

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

Hasil

```
MariaDB [rental_nafan]> select * from mobil;
```

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

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



```
MariaDB [rental_nafan]> SELECT * FROM mobil
-> WHERE pemilik LIKE '%m';
```

id_mobil	no_plat	no_mesin	warna	pemilik	peminjam	batas_peminjam	harga_rental
1	DD 2650 XY	ACX3568	Hitam	Ibrahim	Afdal	NULL	50000
2	DD 2440 AX	BCS1120	Merah	Ibrahim	Elia	NULL	100000
3	B 1611 QC	LSQ1112	Silver	Baim	Anty	NULL	50000

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

Analisis

- `SELECT * FROM` : Merupakan query awal untuk menampilkan sebuah tabel yang didalamnya berisikan kolom, di mana kolom tersebut menampung sebuah data.
- `mobil` nama dari tabel yang ingin kita tampilkan data-data nya.
- `WHERE` : digunakan untuk menyaring data yang ada pada tabel.
- `pemilik LIKE '%a'` : Query tersebut mencari data pada kolom pemilik yang akhiran katanya adalah huruf a yang akan diambil

Kesimpulan

`SELECT * FROM mobil WHERE pemilik LIKE '%a'` : Perintah query tersebut akan menampilkan sebuah data pada tabel mobil dimana kolom pemilik akhiran katanya huruf a.

MENCARI AWALAN & AKHIRAN

Struktur Query

```
SELECT * FROM [nama_tabel]
WHERE [nama_kolom] LIKE 'nilai_awal%nilai_akhir';
```

Contoh

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

Hasil

```
MariaDB [rental_nafan]> SELECT * FROM mobil
-> ;
```

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

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

```
MariaDB [rental_nafan]> SELECT * FROM mobil
-> WHERE pemilik LIKE 'i%m';
```

id_mobil	no_plat	no_mesin	warna	pemilik	peminjam	batas_peminjam	harga_rental
1	DD 2650 XY	ACX3568	Hitam	Ibrahim	Afdal	NULL	50000

```
1 row in set (0.001 sec)
```

Analisis

- `SELECT * FROM` : Merupakan query awal untuk menampilkan sebuah tabel yang didalamnya berisikan kolom, di mana kolom tersebut menampung sebuah data.
- `mobil` nama dari tabel yang ingin kita tampilkan data-data nya.
- `WHERE` : digunakan untuk menyaring data yang ada pada tabel.
- `pemilik LIKE 'T%q'` : Perintah query tersebut akan menampilkan sebuah data pada kolom pemilik yang berawalan huruf T dan akhiran katanya q.

Kesimpulan

`SELECT * FROM mobil WHERE pemilik LIKE 'T%q'` : Perintah query tersebut mencari data pada tabel dimana kolom pemilik yang awalan katanya huruf T dan akhiran katanya q yang akan diambil.

MENCARI BERDASARKAN TOTAL KARAKTER

Struktur Query

```
SELECT * FROM [nama_tabel]
WHERE [nama_kolom] LIKE 'Nilai__';

SELECT * FROM [nama_tabel]
WHERE [nama_kolom] LIKE 'underscore';
```

Contoh

```
SELECT * FROM mobil
WHERE pemilik LIKE 'B___';

SELECT * FROM mobil
WHERE warna LIKE '_____';
```

Hasil

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

MariaDB [rental_nafan]> SELECT * FROM mobil
-> WHERE pemilik LIKE 'B___';
+-----+-----+-----+-----+-----+-----+-----+-----+
| id_mobil | no_plat | no_mesin | warna | pemilik | peminjam | batas_peminjam | harga_rental |
+-----+-----+-----+-----+-----+-----+-----+-----+
| 3 | B 1611 QC | LSQ1112 | Silver | Baim | Anty | NULL | 50000 |
+-----+-----+-----+-----+-----+-----+-----+-----+
1 row in set (0.001 sec)
```

```

MariaDB [rental_nafan]> select * from mobil;
+-----+-----+-----+-----+-----+-----+-----+-----+
| id_mobil | no_plat | no_mesin | warna | pemilik | peminjam | batas_peminjam | harga_rental |
+-----+-----+-----+-----+-----+-----+-----+-----+
| 1 | DD 2650 XY | ACX3568 | Hitam | Ibrahim | Afdal | NULL | 50000 |
| 2 | DD 2440 AX | BCS1120 | Merah | Ibrahim | Elia | NULL | 100000 |
| 3 | B 1611 QC | LSQ1112 | Silver | Baim | Anty | NULL | 50000 |
| 4 | DD 2901 JK | UQL1029 | Hitam | Ibe | NULL | NULL | 150000 |
| 5 | DD 2210 LS | CJH1011 | Hitam | Ibe | NULL | NULL | 100000 |
+-----+-----+-----+-----+-----+-----+-----+-----+
5 rows in set (0.000 sec)

MariaDB [rental_nafan]> SELECT * FROM mobil
-> WHERE warna LIKE '_____';
+-----+-----+-----+-----+-----+-----+-----+-----+
| id_mobil | no_plat | no_mesin | warna | pemilik | peminjam | batas_peminjam | harga_rental |
+-----+-----+-----+-----+-----+-----+-----+-----+
| 1 | DD 2650 XY | ACX3568 | Hitam | Ibrahim | Afdal | NULL | 50000 |
| 2 | DD 2440 AX | BCS1120 | Merah | Ibrahim | Elia | NULL | 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 (0.001 sec)

```

Analisis

- `SELECT * FROM` : Merupakan query awal untuk menampilkan sebuah tabel yang didalamnya berisikan kolom, di mana kolom tersebut menampung sebuah data.
- `mobil` nama dari tabel yang ingin kita tampilkan data-data nya.
- `WHERE` : digunakan untuk menyaring data yang ada pada tabel.
- `pemilik LIKE 'T_____';` : perintah query ini akan mencari data dimana kolom pemilik memiliki awalan kata huruf T serta panjang karakternya sebanyak 6.
- `warna LIKE '_____';` : perintah ini mencari data, dimana kolom warna yang memiliki jumlah karakter sebanyak 6 yang akan diambil.

Kesimpulan

- `SELECT * FROM mobil WHERE pemilik LIKE 'T_____';` : perintah query tersebut akan mencari data pada tabel mobil, dimana kolom pemilik memiliki awalan kata T dan jumlah karakter sebanyak 6 yang akan diambil.
- `SELECT * FROM mobil WHERE warna LIKE '_____';` : dan kalau perintah query ini akan mencari data pada tabel mobil, dimana kolom warna memiliki jumlah karakter sebanyak 6 maka akan diambil.

KOMBINASI

```

SELECT * FROM [nama_tabel]
WHERE [nama_kolom] LIKE 'nilai%'

```

```

SELECT * FROM [nama_tabel]

```



```
WHERE [nama_kolom] LIKE 'nilai%';
```

Contoh

```
SELECT * FROM mobil  
WHERE pemilik LIKE '_d%';
```

```
SELECT * FROM mobil  
WHERE pemilik LIKE '__i%';
```

Hasil

```
MariaDB [rental_nafan]> select * from mobil;
```

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

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

```
MariaDB [rental_nafan]> SELECT * FROM mobil  
-> WHERE pemilik LIKE '_i%';
```

id_mobil	no_plat	no_mesin	warna	pemilik	peminjam	batas_peminjam	harga_rental
2	DD 2440 AX	BCS1120	Merah	Ibrahim	Elia	NULL	100000

```
1 row in set (0.001 sec)
```

```
MariaDB [rental_nafan]> select * from mobil;
```

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

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

```
MariaDB [rental_nafan]> SELECT * FROM mobil  
-> WHERE pemilik LIKE '__i%';
```

id_mobil	no_plat	no_mesin	warna	pemilik	peminjam	batas_peminjam	harga_rental
3	B 1611 QC	LSQ1112	Silver	Baim	Anty	NULL	50000

```
1 row in set (0.002 sec)
```

Analisis

- `SELECT * FROM` : Merupakan query awal untuk menampilkan sebuah tabel yang didalamnya berisikan kolom, di mana kolom tersebut menampung sebuah data.
- `mobil` nama dari tabel yang ingin kita tampilkan data-data nya.
- `WHERE` : digunakan untuk menyaring data yang ada pada tabel.
- `pemilik LIKE '_d%'` ; perintah ini akan mencari data pada kolom pemilik, dimana underscore nya sebagai pengganti karakter awal dan karakter keduanya bernilai d.
- `pemilik LIKE '__i%'` ; perintah ini akan mencari data pada kolom pemilik, yang dimana 2 underscore nya sebagai pengganti nilai asli dan karakter ketiganya bernilai i.

Kesimpulan

- `SELECT * FROM mobil WHERE pemilik LIKE '_d%'` ; Perintah query tersebut akan mencari data pada tabel mobil, dimana kolom pemilik yang bernilai d pada karakter keduanya akan diambil.
 - `SELECT * FROM mobil WHERE pemilik LIKE '__i%'` ; Perintah query tersebut akan mencari data pada tabel mobil, dimana kolom pemilik yang bernilai i pada karakter ketiga yang akan diambil.
-

NOT LIKE

Struktur Query

```
SELECT * FROM [nama_tabel]
WHERE [nama_kolom] NOT LIKE 'nilai%';
```

Contoh

```
SELECT * FROM mobil
WHERE pemilik NOT LIKE 'nafan%';
```

Hasil

```
MariaDB [rental_nafan]> select * from mobil;
```

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

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



```
MariaDB [rental_nafan]> SELECT * FROM mobil
-> WHERE pemilik NOT LIKE 'nafan%';
```

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

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

Analisis

- `SELECT * FROM` : Merupakan query awal untuk menampilkan sebuah tabel yang didalamnya berisikan kolom, di mana kolom tersebut menampung sebuah data.
- `mobil` nama dari tabel yang ingin kita tampilkan data-data nya.
- `WHERE` : digunakan untuk menyaring data yang ada pada tabel.
- `pemilik NOT LIKE 'opi%'` : Perintah query ini tidak akan menampilkan yang bernilai Opi pada kolom pemilik

Kesimpulan

`SELECT * FROM mobil WHERE pemilik NOT LIKE 'opi%'` : Jadi Perintah tersebut mengecualikan sebuah data pada tabel mobil dimana kolom pemilik yang bernilai Opi tidak akan ditampilkan.

NULL & NOT NULL

NULL

Struktur Query

```
SELECT * FROM [nama_tabel] WHERE [nama_kolom] IS NULL;
```

Contoh

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

Hasil

```
MariaDB [rental_nafan]> select * from mobil;
```

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

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



```
MariaDB [rental_nafan]> SELECT * FROM mobil WHERE peminjam IS NULL;
```

id_mobil	no_plat	no_mesin	warna	pemilik	peminjam	batas_peminjam	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 (0.003 sec)
```

Analisis

- `SELECT * FROM` : Merupakan query awal untuk menampilkan sebuah tabel yang didalamnya berisikan kolom, di mana kolom tersebut menampung sebuah data.
- `mobil` nama dari tabel yang ingin kita tampilkan data-data nya.
- `WHERE` : digunakan untuk menyaring data yang ada pada tabel.
- `peminjam IS NULL` pada kolom peminjam akan menampilkan data yang kosong.

Kesimpulan

Null digunakan ketika kita ingin menampilkan kolom yang kosong/belum ada isinya.

NOT NULL

Struktur Query

```
SELECT * FROM [nama_tabel] WHERE [nama_kolom] IS NOT NULL;
```

Contoh

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

Hasil

```
MariaDB [rental_nafan]> select * from mobil;
```

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

5 rows in set (0.000 sec)

```
MariaDB [rental_nafan]> SELECT * FROM mobil WHERE peminjam IS NOT NULL;
```

id_mobil	no_plat	no_mesin	warna	pemilik	peminjam	batas_peminjam	harga_rental
1	DD 2650 XY	ACX3568	Hitam	Ibrahim	Afdal	NULL	50000
2	DD 2440 AX	BCS1120	Merah	Ibrahim	Elia	NULL	100000
3	B 1611 QC	LSQ1112	Silver	Baim	Anty	NULL	50000

3 rows in set (0.003 sec)

Analisis

- `SELECT * FROM` : Merupakan query awal untuk menampilkan sebuah tabel yang didalamnya berisikan kolom, di mana kolom tersebut menampung sebuah data.
- `mobil` nama dari tabel yang ingin kita tampilkan data-data nya.
- `WHERE` : digunakan untuk menyaring data yang ada pada tabel.
- `peminjam IS NOT NULL` pada kolom peminjam akan ditampilkan data yang terisi/tidak kosong.

Kesimpulan

Not null di gunakan ketika kita ingin menampilkan data yang terisi atau tidak kosong.

ORDER BY & LIMIT

Mengurutkan data dari data terkecil

Struktur Query

```
SELECT * FROM [nama_tabel] ORDER BY [nama_kolom] ASC;
```

Contoh

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

Hasil

```
MariaDB [rental_nafan]> select * from mobil;
```

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

5 rows in set (0.001 sec)

```
MariaDB [rental_nafan]> SELECT * FROM mobil ORDER BY pemilik ASC;
```

id_mobil	no_plat	no_mesin	warna	pemilik	peminjam	batas_peminjam	harga_rental
2	DD 2440 AX	BCS1120	Merah	Ibrahim	Elia	NULL	100000
3	B 1611 QC	LSQ1112	Silver	Baim	Anty	NULL	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	NULL	50000

5 rows in set (0.032 sec)

Analisis

- `SELECT * FROM` : Merupakan query awal untuk menampilkan sebuah tabel yang didalamnya berisikan kolom, di mana kolom tersebut menampung sebuah data.
- `mobil` nama dari tabel yang ingin kita tampilkan data-data nya.
- `ORDER BY` digunakan untuk mengurutkan hasil query berdasarkan hasil tertentu.
- `ASC` digunakan untuk mengurutkan hasil secara berurutan dari nilai terkecil ke nilai terbesar, seperti urutan alfabetis dari A ke Z.

Kesimpulan

`ORDER BY ASC` digunakan untuk mengurutkan nilai dari yang terkecil.

Mengurutkan data dari data terbesar

Struktur Query

```
SELECT * FROM [nama_tabel] ORDER BY [nama_kolom] DESC;
```

Contoh

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

Hasil

```
MariaDB [rental_nafan]> select * from mobil;
```

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

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



```
MariaDB [rental_nafan]> SELECT * FROM mobil ORDER BY peminjam DESC;
```

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

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

Analisis

- `SELECT * FROM` : Merupakan query awal untuk menampilkan sebuah tabel yang didalamnya berisikan kolom, di mana kolom tersebut menampung sebuah data.
- `mobil` nama dari tabel yang ingin kita tampilkan data-data nya.
- `ORDER BY` digunakan untuk mengurutkan hasil query berdasarkan hasil tertentu.
- `DESC` digunakan untuk mengurutkan hasil secara berurutan dari nilai terbesar ke nilai terkecil, seperti urutan alfabetis dari Z ke A.

Kesimpulan

`ORDER BY DESC` digunakan untuk mengurutkan nilai dari yang terbesar.

Membatasi data yang tampil

Struktur Query

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

Contoh

```
SELECT * FROM mobil
```

```
WHERE warna = 'Hitam' ORDER BY
```

```
harga_rental ASC LIMIT 2;
```

Hasil

```
MariaDB [rental_nafan]> select * from mobil;
```

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

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



```
MariaDB [rental_nafan]> SELECT * FROM mobil
-> WHERE warna = 'Hitam' ORDER BY harga_rental ASC LIMIT 2;
```

id_mobil	no_plat	no_mesin	warna	pemilik	peminjam	batas_peminjam	harga_rental
1	DD 2650 XY	ACX3568	Hitam	Ibrahim	Afdal	NULL	50000
5	DD 2210 LS	CJH1011	Hitam	Ibe	NULL	NULL	100000

```
2 rows in set (0.036 sec)
```

Analisis

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

Kesimpulan

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

DISTINCT

Contoh Pertama

Struktur Query


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

Contoh

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

Hasil

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

MariaDB [rental_nafan]> SELECT DISTINCT(warna) FROM mobil;
+-----+
| warna |
+-----+
| Hitam |
| Merah |
| Silver |
+-----+
3 rows in set (0.008 sec)
```

Analisis

- `SELECT DISTINCT (warna)` digunakan untuk mengambil nilai unik dari kolom warna dalam tabel, dan menghapus data duplikat sehingga hanya menampilkan data-data yang berbeda.
- `FROM mobil` Query tersebut memerintahkan untuk mengambil data pada tabel mobil.

Kesimpulan

Distinct digunakan untuk menampilkan data-data yang asli atau tidak duplikat

Kombinasi Distinct dan Order By

Struktur Query

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

Contoh

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

Hasil

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

MariaDB [rental_nafan]> SELECT DISTINCT(harga_rental) FROM mobil ORDER BY harga_rental DESC;
+-----+
| harga_rental |
+-----+
| 150000 |
| 100000 |
| 50000 |
+-----+
3 rows in set (0.004 sec)
```

Analisis

- `SELECT DISTINCT (harga_rental)` digunakan untuk mengambil nilai unik dari kolom `harga_rental` dalam tabel, dan menghapus data duplikat sehingga hanya menampilkan data-data yang berbeda.
- `FROM mobil` Query tersebut memerintahkan untuk mengambil data pada tabel mobil.
- `ORDER BY harga_rental DESC` Query tersebut Mengurutkan isi dalam kolom `harga_rental` dari yang tertinggi.

Kesimpulan

Kombinasi Distinct dan Order By digunakan ketika ingin mengetahui jumlah data asli yang ada pada kolom secara berurutan, baik dari yang tertinggi ataupun yang terendah.

CONCAT, CONCAT_WS, AS

Menggabungkan kolom tanpa pemisah

Struktur Query

```
SELECT CONCAT(kolom1,kolom2) FROM [nama_tabel];
```

Contoh

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

Hasil

```
MariaDB [rental_nafan]> select * from mobil;
+-----+-----+-----+-----+-----+-----+-----+-----+
| id_mobil | no_plat | no_mesin | warna | pemilik | peminjam | batas_peminjam | harga_rental |
+-----+-----+-----+-----+-----+-----+-----+-----+
| 1 | DD 2650 XY | ACX3568 | Hitam | Ibrahim | Afdal | NULL | 50000 |
| 2 | DD 2440 AX | BCS1120 | Merah | Ibrahim | Elia | NULL | 100000 |
| 3 | B 1611 QC | LSQ1112 | Silver | Baim | Anty | NULL | 50000 |
| 4 | DD 2901 JK | UQL1029 | Hitam | Ibe | NULL | NULL | 150000 |
| 5 | DD 2210 LS | CJH1011 | Hitam | Ibe | NULL | NULL | 100000 |
+-----+-----+-----+-----+-----+-----+-----+-----+
5 rows in set (0.002 sec)

MariaDB [rental_nafan]> SELECT CONCAT(pemilik, warna) FROM mobil;
+-----+
| CONCAT(pemilik, warna) |
+-----+
| IbrahimHitam |
| IbrahimMerah |
| BaimSilver |
| IbeHitam |
| IbeHitam |
+-----+
5 rows in set (0.004 sec)
```

Analisis

- `SELECT CONCAT` digunakan untuk mengambil dan menggabungkan 2 kolom atau lebih menjadi satu string tunggal.
- `(pemilik, warna)` Merupakan kolom yang akan di gabungkan menjadi satu.
- `FROM mobil` query tersebut akan mengambil data dari tabel mobil.

kesimpulan

`CONCAT` digunakan ketika kita ingin menggabungkan suatu kolom menjadi satu.

Menggabungkan kolom dengan pemisah

Struktur Query

```
SELECT CONCAT_WS(Separator, kolom1, kolom2,
kolom3) FROM [nama_tabel];
```

Contoh

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

Hasil

```
MariaDB [rental_nafan]> select * from mobil;  
+-----+-----+-----+-----+-----+-----+-----+-----+  
| id_mobil | no_plat | no_mesin | warna | pemilik | peminjam | batas_peminjam | harga_rental |  
+-----+-----+-----+-----+-----+-----+-----+-----+  
| 1 | DD 2650 XY | ACX3568 | Hitam | Ibrahim | Afdal | NULL | 50000 |  
| 2 | DD 2440 AX | BCS1120 | Merah | Ibrahim | Elia | NULL | 100000 |  
| 3 | B 1611 QC | LSQ1112 | Silver | Baim | Anty | NULL | 50000 |  
| 4 | DD 2901 JK | UQL1029 | Hitam | Ibe | NULL | NULL | 150000 |  
| 5 | DD 2210 LS | CJH1011 | Hitam | Ibe | NULL | NULL | 100000 |  
+-----+-----+-----+-----+-----+-----+-----+-----+  
5 rows in set (0.002 sec)  
  
MariaDB [rental_nafan]> SELECT CONCAT_WS("-",no_plat,no_mesin,  
-> id_mobil) FROM mobil;  
+-----+  
| CONCAT_WS("-",no_plat,no_mesin,  
id_mobil) |  
+-----+  
| DD 2650 XY-ACX3568-1 |  
| DD 2440 AX-BCS1120-2 |  
| B 1611 QC-LSQ1112-3 |  
| DD 2901 JK-UQL1029-4 |  
| DD 2210 LS-CJH1011-5 |  
+-----+  
5 rows in set (0.002 sec)
```

Analisis

- `SELECT CONCAT_WS` digunakan untuk mengambil dan menggabungkan 2 kolom atau lebih menjadi satu string tunggal dengan pemisah.
- `("-",no_plat,no_mesin,id_mobil)` tanda "-" adalah pemisah yang akan digunakan. `no_plat,no_mesin,id_mobil` nama kolom-kolom yang dipilih dari tabel untuk di gabungkan.
- `FROM mobil` query tersebut akan mengambil data dari tabel mobil.

kesimpulan

`CONCAT_WS` digunakan untuk menggabungkan nilai kolom-kolom dengan pemisah agar data-data yang di gabung terlihat rapi.

Memberikan nama kolom alias

Struktur Query

```
SELECT CONCAT_WS(Separator,kolom1,kolom2) AS [nama_alias] FROM mobil;
```

Contoh

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

Hasil

```
MariaDB [rental_nafan]> select * from mobil;
```

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

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

```
MariaDB [rental_nafan]> SELECT CONCAT_WS("+",pemilik,peminjam) AS COLLAB FROM mobil;
```

COLLAB
Ibrahim+Afdal
Ibrahim+Elia
Baim+Anty
Ibe
Ibe

```
5 rows in set (0.004 sec)
```

Analisis

- `SELECT CONCAT_WS` digunakan untuk mengambil dan menggabungkan 2 kolom atau lebih menjadi satu string tunggal dengan pemisah.
- `("+",pemilik,peminjam)` tanda "+" sebagai pemisah `pemilik,peminjam` merupakan kolom yang akan digabungkan nilainya, setiap nilai dari kolom `pemilik` dan `peminjam` akan digabungkan menjadi satu string dengan tanda "+" di antara mereka.
- `AS COLLAB` `AS` merupakan alias sedangkan `collab` nama dari penggabungan 2 kolom yaitu `pemilik` dan `peminjam`.
- `FROM mobil` query tersebut akan mengambil data dari tabel `mobil`.

kesimpulan

Kita bisa menggunakan Query tersebut untuk menggabungkan sebuah kolom dengan memberi nama alias, agar lebih mudah di mengerti.

VIEW

Membuat tabel virtual

Struktur Query

```
CREATE VIEW [nama_tabel_virtual] AS SELECT id_mobil,no_plat,pemilik,peminjam FROM [nama_tabel] WHERE [nama_kolom] = "nilai";
```

Contoh

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

Hasil

```
MariaDB [rental_nafan]> CREATE VIEW info_no_plat AS SELECT id_mobil, no_plat, pemilik, peminjam FROM mobil WHERE pemilik = "Baim";
Query OK, 0 rows affected (0.013 sec)
```

Analisis

- `CREATE VIEW info_no_plat AS` Perintah query tersebut membuat sebuah tabel virtual dengan nama `info_no_plat`.
- `SELECT id_mobil, no_plat, pemilik, peminjam FROM mobil` digunakan untuk memilih kolom-kolom dari tabel `mobil` untuk di masukkan ke dalam tabel virtual yang dibuat.
- `WHERE pemilik = "Baim";` perintah tersebut akan menampilkan data-data yang pemiliknya Dani.

kesimpulan

Tabel virtual digunakan untuk mengelompokkan sebuah kolom tertentu dari tabel yang sudah ada.

Menampilkan tabel virtual

Struktur Query

Contoh

```
SELECT * FROM info_no_plat;
```

Hasil

```

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

MariaDB [rental_nafan]> select * from info_no_plat;
+-----+-----+-----+-----+
| id_mobil | no_plat | pemilik | peminjam |
+-----+-----+-----+-----+
| 3 | B 1611 QC | Baime | Anty |
+-----+-----+-----+-----+
1 row in set (0.001 sec)

```

Analisis

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

kesimpulan

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

Menghapus tabel virtual

Struktur Query

Contoh

```
DROP VIEW info_no_plat;
```

Hasil

```

MariaDB [rental_nafan]> DROP VIEW info_no_plat;
Query OK, 0 rows affected (0.003 sec)

MariaDB [rental_nafan]> select * from no_plat;
ERROR 1146 (42S02): Table 'rental_nafan.no_plat' doesn't exist
MariaDB [rental_nafan]>

```

Analisis

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

kesimpulan

Jika ingin menghapus tabel virtual cukup dengan perintah `DROP VIEW`.

Tantangan

1

Struktur Query

```
CREATE VIEW [nama_tabel_virtual] AS SELECT [kolom_pilihan] FROM [nama_tabel] WHERE [kolom] = nilai;
```

Contoh

```
CREATE VIEW tantangan AS SELECT no_plat, warna, peminjam FROM mobil WHERE peminjam is null;
```

Hasil

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

MariaDB [rental_nafan]> select * from tantangan;
+-----+-----+-----+
| no_plat | warna | peminjam |
+-----+-----+-----+
| DD 2901 JK | Hitam | NULL |
| DD 2210 LS | Hitam | NULL |
+-----+-----+-----+
2 rows in set (0.018 sec)
```

Analisis

- `CREATE VIEW` perintah MySQL yang digunakan untuk membuat tabel view yang akan tersimpan di dalam database.

- tantangan AS nama alias dari tabel virtual yang kita buat.
- SELECT no_plat, warna, peminjam Kolom yang kita pilih untuk di masukkan ke dalam sebuah tabel view yang dibuat.
- FROM mobil kolom yang kita pilih untuk tabel view merupakan dari tabel mobil yang sudah ada.
- WHERE peminjam is null; data yang akan tampil dimana peminjamnya null atau tidak ada.

Kesimpulan

Program MySQL tersebut membuat sebuah view yang disebut tantangan yang berisi daftar mobil yang belum dipinjam, dengan kolom-kolom no_plat, warna, dan peminjam dari tabel mobil di mana nilai peminjam belum diisi (NULL). View ini menyediakan informasi tentang mobil-mobil yang tersedia untuk dipinjam.

2

Struktur Query

```
UPDATE [nama_tabel] SET [nama_kolom] = nilai WHERE [nama_kolom] = nilai;
```

Contoh

```
UPDATE mobil SET peminjam = Farhan WHERE id_mobil = 4;
```

Hasil

```
MariaDB [rental_nafan]> UPDATE mobil SET peminjam = 'Farhan' WHERE id_mobil = 4;
Query OK, 1 row affected (0.010 sec)
Rows matched: 1  Changed: 1  Warnings: 0
```

```
MariaDB [rental_nafan]> select * from mobil;
```

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

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

```
MariaDB [rental_nafan]> select * from tantangan;
```

no_plat	warna	peminjam
DD 2210 LS	Hitam	NULL

```
1 row in set (0.001 sec)
```

Analisis

- `UPDATE mobil` Query tersebut akan mengupdate data pada tabel mobil.
- `SET peminjam = NULL` perintah tersebut akan mengosongkan kolom peminjam dengan `NULL`.
- `WHERE id_mobil = 4;` data yang akan di update yang memiliki `Id_mobil = 4`.

Kesimpulan

Perintah MySQL tersebut akan mengubah nilai kolom `peminjam` menjadi `NULL` untuk `mobil` yang memiliki `id_mobil = 4` dalam tabel `mobil`.

AGREGASI

Menghitung total nilai numerik suatu kolom (Sum)

Struktur Query

```
SELECT SUM(nama_kolom) AS [nama_alias] FROM [nama_tabel]
```

Contoh

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

Hasil

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

MariaDB [rental_nafan]> SELECT SUM(harga_rental) AS total_harga FROM mobil;
+-----+
| total_harga |
+-----+
| 450000 |
+-----+
1 row in set (0.004 sec)
```

Analisis

- `SELECT` digunakan untuk mengambil/menampilkan data

- `SUM` digunakan untuk menjumlahkan nilai numerik dalam kolom.
- `harga_rental` nama kolom yang ingin di jumlahkan nilainya.
- `AS total_harga` sebagai nama alias kolom
- `FROM mobil` query tersebut akan mengambil data dari tabel mobil.

Kesimpulan

`SUM` digunakan untuk menjumlahkan nilai numerik dalam kolom

Menghitung jumlah baris/data, biasanya berdasarkan kriteria tertentu (Count)

Struktur Query

```
SELECT COUNT(nama_kolom) AS [nama_alias] FROM
[nama_tabel];
```

Contoh

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

Hasil

```
MariaDB [rental_nafan]> select * from mobil;
+-----+-----+-----+-----+-----+-----+-----+-----+
| id_mobil | no_plat | no_mesin | warna | pemilik | peminjam | batas_peminjam | harga_rental |
+-----+-----+-----+-----+-----+-----+-----+-----+
| 1 | DD 2650 XY | ACX3568 | Hitam | Ibrahim | Afdal | NULL | 50000 |
| 2 | DD 2440 AX | BCS1120 | Merah | Ibrahim | Elia | NULL | 100000 |
| 3 | B 1611 QC | LSQ1112 | Silver | Baim | Anty | NULL | 50000 |
| 4 | DD 2901 JK | UQL1029 | Hitam | Ibe | Farhan | NULL | 150000 |
| 5 | DD 2210 LS | CJH1011 | Hitam | Ibe | NULL | NULL | 100000 |
+-----+-----+-----+-----+-----+-----+-----+-----+
5 rows in set (0.003 sec)

MariaDB [rental_nafan]> SELECT COUNT(pemilik) AS total_pemilik FROM mobil;
+-----+
| total_pemilik |
+-----+
| 5 |
+-----+
1 row in set (0.006 sec)
```

Analisis

- `SELECT` digunakan untuk mengambil/menampilkan data.
- `COUNT` digunakan untuk menghitung jumlah baris(nilai non-NULL).

- `pemilik` nama kolom yang ingin di hitung jumlah barisnya.
- `AS total_pemilik` sebagai nama alias kolom
- `FROM mobil` query tersebut akan mengambil data dari tabel mobil.

Kesimpulan

`COUNT` digunakan ketika ingin menghitung jumlah baris dalam suatu kolom(atau jumlah non-NULL).

Struktur Query

```
SELECT COUNT(nama_kolom) AS [nama_alias] FROM
[nama_tabel];
```

Contoh

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

Hasil

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

MariaDB [rental_nafan]> SELECT COUNT(peminjam) AS total_peminjam FROM mobil;
+-----+
| total_peminjam |
+-----+
| 4 |
+-----+
1 row in set (0.001 sec)
```

Analisis

- `SELECT` digunakan untuk mengambil/menampilkan data.
- `COUNT` digunakan untuk menghitung jumlah baris(nilai non-NULL).
- `peminjam` nama kolom yang ingin di hitung jumlah barisnya.
- `AS total_peminjam` sebagai nama alias kolom
- `FROM mobil` query tersebut akan mengambil data dari tabel mobil.

Kesimpulan

`COUNT` digunakan untuk menghitung jumlah baris dalam kolom yang hanya memiliki (bukan NULL)

Menampilkan nilai terendah (Min)

Struktur Query

```
SELECT MIN(nama_kolom) AS [nama_alias] FROM [nama_tabel];
```

Contoh

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

Hasil

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

MariaDB [rental_nafan]> SELECT MIN(harga_rental) AS minimum FROM mobil;
+-----+
| minimum |
+-----+
| 50000 |
+-----+
1 row in set (0.005 sec)
```

Analisis

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

Kesimpulan

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

Menampilkan nilai tertinggi (Max)

Struktur Query

```
SELECT MAX(nama_kolom) AS [nama_alias] FROM [nama_tabel];
```

Contoh

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

Hasil

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

MariaDB [rental_nafan]> SELECT MAX(harga_rental) AS maximum FROM mobil;
+-----+
| maximum |
+-----+
| 150000 |
+-----+
1 row in set (0.001 sec)
```

Analisis

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

Kesimpulan

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

Menampilkan nilai rata-rata (Avg)

Struktur Query

```
SELECT AVG(nama_kolom) AS [nama_alias] FROM [nama_tabel];
```

Contoh

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

Hasil

```
MariaDB [rental_nafan]> select * from mobil;
+-----+-----+-----+-----+-----+-----+-----+-----+
| id_mobil | no_plat | no_mesin | warna | pemilik | peminjam | batas_peminjam | harga_rental |
+-----+-----+-----+-----+-----+-----+-----+-----+
| 1 | DD 2650 XY | ACX3568 | Hitam | Ibrahim | Afdal | NULL | 50000 |
| 2 | DD 2440 AX | BCS1120 | Merah | Ibrahim | Elia | NULL | 100000 |
| 3 | B 1611 QC | LSQ1112 | Silver | Baim | Anty | NULL | 50000 |
| 4 | DD 2901 JK | UQL1029 | Hitam | Ibe | Farhan | NULL | 150000 |
| 5 | DD 2210 LS | CJH1011 | Hitam | Ibe | NULL | NULL | 100000 |
+-----+-----+-----+-----+-----+-----+-----+-----+
5 rows in set (0.009 sec)

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

Analisis

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

Kesimpulan

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

Tantangan 7 NOMOR

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

Struktur Query

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

Query

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

Hasil

```
MariaDB [rental_nafan]> select warna,COUNT(id_mobil) AS Jumlah_Data_Mobil FROM mobil GROUP BY warna;
+-----+-----+
| warna | Jumlah_Data_Mobil |
+-----+-----+
| Hitam | 3 |
| Merah | 1 |
| Silver | 1 |
+-----+-----+
3 rows in set (0.065 sec)
```

Analisis

- **SELECT Klausa :**

warna: Memilih kolom warna dari tabel mobil.

- **``COUNT(id_mobil) AS Jumlah_Data_Mobil:**

Menghitung jumlah baris (mobil) untuk setiap warna unik dan memberi alias Jumlah_Data_Mobil pada hasil hitungan tersebut.

- **FROM Klausa:**

mobil: Menentukan tabel data_mobil sebagai sumber data.

- **GROUP BY Klausa:**

warna: Mengelompokkan hasil query berdasarkan nilai di kolom warna. Setiap nilai unik dalam kolom warna akan menjadi satu grup.

Kesimpulan

1. Mengelompokkan Data Berdasarkan Warna: Data dalam tabel mobil dikelompokkan berdasarkan kolom warna.

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

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

Struktur Query

```
select nama_data,COUNT(nama_data) AS nama_sementara from nama_tabel GROUP BY  
nama_data HAVING COUNT(nama_data) >= 2;
```

Query

```
select pemilik,COUNT(id_mobil) AS jumlah_mobil from mobil GROUP BY pemilik HAVING  
COUNT(id_mobil) >= 2;
```

Hasil

```
MariaDB [rental_nafan]> select pemilik,COUNT(id_mobil) AS jumlah_mobil from mobil GROUP BY pemilik HAVING COUNT(id_mobil) >= 2;  
+-----+-----+  
| pemilik | jumlah_mobil |  
+-----+-----+  
| Ibe     | 2            |  
+-----+-----+  
1 row in set (0.001 sec)
```

Analisis

1. SELECT Klausa

pemilik: Kolom ini dipilih dari tabel mobil. Kolom pemilik berisi data tentang pemilik mobil.

2. COUNT(id_mobil) AS jumlah_mobil: Fungsi agregat COUNT digunakan untuk menghitung jumlah baris dalam setiap grup yang memiliki pemilik yang sama. Hasil hitungan ini diberi alias jumlah_mobil, sehingga dalam hasil akhir, kolom ini akan diberi nama jumlah_mobil.
3. FROM Klausa data_mobil: Tabel ini merupakan sumber data dari query. Tabel ini diasumsikan berisi data mobil, termasuk kolom pemilik dan id_mobil.
4. GROUP BY pemilik: Pernyataan ini mengelompokkan baris-baris data berdasarkan nilai dalam kolom pemilik. Semua baris yang memiliki nilai pemilik yang sama akan dimasukkan ke dalam grup yang sama.

5. **HAVING COUNT(id_mobil) >= 2:** Pernyataan ini menyaring grup-grup yang terbentuk berdasarkan hasil agregat. Hanya grup yang memiliki jumlah baris (mobil) setidaknya 2 yang akan dimasukkan dalam hasil akhir. HAVING digunakan setelah pengelompokan data, berbeda dengan WHERE yang digunakan sebelum pengelompokan.

Kesimpulan

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

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

Struktur Query

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

Query

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

Hasil

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

pemilik	jumlah_mobil
Ibrahim	1
Baim	1
Ibe	2
Ibrahim	1

4 rows in set (0.001 sec)

Analisis

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

Kesimpulan

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

4.berdasarkan query yang ada pada praktikum 5 bagian 7 tampilkan data pada table mobil dengan mengelompokkan

berdasarkan pemiliknya.hitung menggunakan sum total pendapatan pemilik berdasarkan harga rental

Struktur Query

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

Query

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

Hasil

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

pemilik	jumlah_pendapatan
Ibrahim	100000
Baim	50000
Ibe	250000
Ibrahim	50000

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

Analisis

- **SELECT** merupakan perintah yang digunakan untuk memilih data dari database.
- **pemilik** adalah nama kolom yang akan diambil dari tabel **mobil**.
- **SUM(harga_rental)** adalah fungsi yang digunakan untuk menghitung total nilai dari kolom **harga_rental**.
- **AS jumlah_pendapatan** memberikan alias pada hasil perhitungan **SUM(harga_rental)** sehingga hasilnya akan diberi nama **jumlah_pendapatan**.
- **FROM data_mobil** menentukan tabel **mobil** sebagai sumber data.
- **GROUP BY pemilik** mengelompokkan data berdasarkan kolom **pemilik** dan melakukan perhitungan **SUM** untuk setiap kelompok.

Kesimpulan

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

5. Berdasarkan praktikum 5 query no 8 tampilkan jumlah pemasukan pemilik berdasarkan harga rental kelompokkan berdasarkan pemiliknya dan seleksi yang total pemasukannya atau harga rentalnya mencapai lebih besar atau sama dengan 250k

Struktur Query

```
select data_mobil,SUM(data_mobil) AS nama_sementara from nama_tabel GROUP BY  
data_mobil HAVING SUM(data_mobil) >= 250000;
```

Query

```
select pemilik,SUM(harga_rental) AS jumlah_pemasukan from mobil GROUP BY pemilik  
HAVING SUM(harga_rental) >= 250000;
```

Hasil

```
MariaDB [rental_nafan]> select pemilik,SUM(harga_rental) AS jumlah_pemasukan from mobil GROUP BY pemilik HAVING SUM(harga_rental) >= 250000;  
+-----+-----+  
| pemilik | jumlah_pemasukan |  
+-----+-----+  
| Ibe    |          250000  |  
+-----+-----+  
1 row in set (0.001 sec)
```

Analisis

- **SELECT** merupakan perintah yang digunakan untuk memilih data dari database.
- **pemilik** adalah nama kolom yang akan diambil dari tabel `mobil`.
- **SUM(harga_rental)** adalah fungsi yang digunakan untuk menghitung total nilai dari kolom `harga_rental`.
- **AS jumlah_pemasukan** memberikan alias pada hasil perhitungan `SUM(harga_rental)` sehingga hasilnya akan diberi nama `jumlah_pemasukan`.
- **FROM data_mobil** menentukan tabel `mobil` sebagai sumber data.
- **GROUP BY pemilik** mengelompokkan data berdasarkan kolom `pemilik` dan melakukan perhitungan `SUM` untuk setiap kelompok.
- **HAVING SUM(harga_rental) >= 250000** merupakan klausa yang digunakan untuk menyaring kelompok hasil perhitungan `SUM(harga_rental)` yang nilainya lebih besar atau sama dengan 250000.

Kesimpulan

Perintah SQL ini akan menghasilkan daftar pemilik mobil beserta total pendapatan dari harga rental yang mereka miliki, tetapi hanya untuk pemilik yang total pendapatannya sama dengan atau lebih dari 250000. Hasil query akan menampilkan dua kolom: `pemilik` yang berisi nama pemilik, dan `jumlah_pemasukan` yang berisi total pendapatan dari harga rental mobil untuk setiap pemilik yang memenuhi kriteria `HAVING` tersebut.

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

Struktur Query

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

Query

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

Hasil

```
MariaDB [rental_nafan]> select pemilik,AVG(harga_rental) AS rata_pemasukam from mobil GROUP BY pemilik;
+-----+-----+
| pemilik | rata_pemasukam |
+-----+-----+
| Ibrahim | 100000.0000 |
| Baïm | 50000.0000 |
| Ibe | 125000.0000 |
| Ibrahim | 50000.0000 |
+-----+-----+
4 rows in set (0.001 sec)
```

Analisis

- **SELECT** merupakan perintah yang digunakan untuk memilih data dari database.
- **pemilik** adalah nama kolom yang akan diambil dari tabel `mobil`. Kolom ini menyimpan informasi tentang pemilik mobil.
- **AVG(harga_rental)** adalah fungsi yang digunakan untuk menghitung nilai rata-rata dari kolom `harga_rental`.

- **AS rata_pemasukan** memberikan alias pada hasil perhitungan `AVG(harga_rental)` sehingga hasilnya akan diberi nama `rata_pemasukan`.
- **FROM data_mobil** menentukan tabel `mobil` sebagai sumber data.
- **GROUP BY pemilik** mengelompokkan data berdasarkan kolom `pemilik` dan melakukan perhitungan `AVG` untuk setiap kelompok.

Kesimpulan

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

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

Struktur Query

```
select nama_data,MAX(nama_data) AS nama_sementara,MIN(nama_data) AS nama_sementara
from nama_tabel GROUP BY nama_data HAVING COUNT(nama_data) >= 1;
```

Query

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

Hasil

```
MariaDB [rental_nafan]> select pemilik,MAX(harga_rental) AS Pemasukan_Terbesar ,MIN(harga_rental) AS pemasukan_terkecil from mobil GR
OUP BY pemilik HAVING COUNT(harga_rental) > 1;
+-----+-----+-----+
| pemilik | Pemasukan_Terbesar | pemasukan_terkecil |
+-----+-----+-----+
| Ibe     | 150000             | 100000             |
+-----+-----+-----+
1 row in set (0.001 sec)
```

Analisis

- **SELECT** merupakan perintah yang digunakan untuk memilih data dari database.

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

Kesimpulan

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

Relasi Tabel

latihan_rpl_1	pengawas
id_pengawas	: int(100)
nama	: varchar(50)

latihan_rpl_1	tabel cabang
Nocab	: varchar(11)
nama	: varchar(100)
alamat	: varchar(100)
Kabupaten	: varchar(100)
no_telp	: int(100)
kota	: varchar(100)
email	: text

latihan_rpl_1	nilai
id_siswa	: int(15)
id_pengawas	: int(5)
nilai	: int(3)

latihan_rpl_1	tabel pegawai
idpeg	: int(7)
nama	: varchar(100)
namakel	: varchar(100)
jk	: varchar(100)
alamat	: varchar(100)
kota	: varchar(100)
telp	: int(20)
nocab	: varchar(10)

latihan_rpl_1	murid
nis	: int(15)
nama	: varchar(50)