

Select Lanjutan

AND

Struktur

```
select kolom1,kolom2 from nama_table where kolom1='nilai_kolom1' and  
kolom2='nilai_kolom2';
```

Contoh

```
select warna,pemilik from mobil where warna='hitam' and pemilik='ibrahim'
```

Hasil

```
+-----+-----+  
| warna | pemilik |  
+-----+-----+  
| Hitam | Ibrahim |  
+-----+-----+
```

Analisis

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

Kesimpulan

jika ingin menampilkan data yang telah di seleksi dengan cara memberikan syarat yang semuanya harus di penuhi kalian bisa menggunakan query dengan struktur

```
select kolom1,kolom2 from nama_table where kolom1='nilai_kolom1' and  
kolom2='nilai_kolom2';
```

OR

Struktur

```
select kolom1,kolom2 from nama_table where kolom1='nilai_kolom1' or kolom2='nilai_kolom2';
```

Contoh

```
select warna,pemilik from mobil where warna='Hitam' or pemilik='Ibrahim';
```

Hasil

```
MariaDB [rental_angga]> select * from mobil where warna='hitam' or pemilik='ibrahim';
```

id_mobil	no_plat	no_mesin	warna	pemilik	peminjam	harga_rental
1	DD 2650 XY	ACX3560	Hitam	Ibrahim	Afdal	50000
2	DD 2440 AX	BCS1120	Merah	Ibrahim	Elia	100000
4	DD 2981 JK	UQL1029	Hitam	Ibe	NULL	150000
5	DD 2210 LS	CJH1011	Hitam	Ibe	NULL	100000

Analisis

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

Kesimpulan

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

Between

Struktur

```
select * from nama_table where nama_kolom between nilai1 and nilai2;
```

Contoh

```
select * from mobil where harga_rental between 100000 and 150000;
```

Hasil

```
MariaDB [rental_angga]> select * from mobil where harga_rental between 100000 and 150000;
+-----+-----+-----+-----+-----+-----+-----+
| id_mobil | no_plat | no_mesin | warna | pemilik | peminjam | harga_rental |
+-----+-----+-----+-----+-----+-----+-----+
| 2 | DD 2440 AX | BCS1120 | Merah | Ibrahim | Elia | 100000 |
| 4 | DD 2981 JK | UQL1029 | Hitam | Ibe | NULL | 150000 |
| 5 | DD 2210 LS | CJH1011 | Hitam | Ibe | NULL | 100000 |
+-----+-----+-----+-----+-----+-----+-----+
```

Analisis

1. `select` query yang digunakan untuk menampilkan masukan dari `insert`
2. `*` berarti semua kolom akan di tampilkan
3. `from` untuk memberikan tanda bahwa table mana yang akan di tampilkan
4. `mobil` nama table yang akan di tampilkan
5. `where` untuk memberikan sebuah kondisi
6. `harga_rental` nama kolom yang digunakan untuk mengkondisikan sebuah table
7. `between` Ini adalah operator yang digunakan untuk memilih rentang nilai
8. `100000 and 150000` Ini adalah nilai rentang yang digunakan dalam kriteria pemilihan data

Kesimpulan

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

Not Between

Struktur

```
select * from nama_table where nama_kolom not between nilai1 and nilai2;
```

Contoh

```
select * from mobil where harga_rental not between 100000 and 150000;
```

Hasil

```
MariaDB [rental_angga]> select * from mobil where harga_rental not between 100000 and 150000;
+-----+-----+-----+-----+-----+-----+-----+
| id_mobil | no_plat | no_mesin | warna | pemilik | peminjam | harga_rental |
+-----+-----+-----+-----+-----+-----+-----+
| 1 | DD 2650 XY | ACX3560 | Hitam | Ibrahim | Afdal | 50000 |
| 3 | B 1611 QC | LSQ1112 | Silver | Baim | Anty | 50000 |
+-----+-----+-----+-----+-----+-----+-----+
```

Analisis

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

Kesimpulan

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

<=

Struktur

```
select * from nama_table where nama_kolom<=nilai;
```

Contoh

```
select * from mobil where harga_rental<=50000;
```

Hasil

```
MariaDB [rental_angga]> select * from mobil where harga_rental <= 50000;
```

id_mobil	no_plat	no_mesin	warna	pemilik	peminjam	harga_rental
1	DD 2650 XY	ACX3560	Hitam	Ibrahim	Afdal	50000
3	B 1611 QC	LSQ1112	Silver	Baim	Anty	50000

Analisis

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

Kesimpulan

jika ingin menampilkan table dengan menggunakan hasil seleksi yang dimana jika dia lebih kecil dari nilai yang di tentukan maka dia akan tampil, yaitu dengan cara menggunakan query dengan struktur

```
select * from nama_table where nama_kolom<=nilai;
```

>=

Struktur

```
select * from nama_table where nama_kolom>=nilai;
```

Contoh

```
select * from mobil where harga_rental>=50000
```

Hasil

```
MariaDB [rental_angga]> select * from mobil where harga_rental >= 50000;
```

id_mobil	no_plat	no_mesin	warna	pemilik	peminjam	harga_rental
1	DD 2650 XY	ACX3560	Hitam	Ibrahim	Afdal	50000
2	DD 2440 AX	BCS1120	Merah	Ibrahim	Elia	100000
3	B 1611 QC	LSQ1112	Silver	Baim	Anty	50000
4	DD 2981 JK	UQL1029	Hitam	Ibe	NULL	150000
5	DD 2210 LS	CJH1011	Hitam	Ibe	NULL	100000

Analisis

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

Kesimpulan

jika ingin menampilkan table dengan menggunakan hasil seleksi yang dimana jika dia lebih besar dari nilai yang di tentukan maka dia akan tampil, yaitu dengan cara menggunakan query dengan struktur

```
select * from nama_table where nama_kolom<=nilai;
```

<> atau !=

Struktur1

```
select * from nama_table where nama_kolom<>nilai;
```

Struktur2

```
select * from nama_table where nama_kolom!=nilai;
```

Contoh1

```
select * from mobil where harga_rental<>50000;
```

Contoh2

```
select * from mobil where harga_rental!=50000;
```

Hasil1

```
MariaDB [rental_angga]> select * from mobil where harga_rental <> 50000;
```

id_mobil	no_plat	no_mesin	warna	pemilik	peminjam	harga_rental
2	DD 2440 AX	BCS1120	Merah	Ibrahim	Elia	100000
4	DD 2981 JK	UQL1029	Hitam	Ibe	NULL	150000
5	DD 2210 LS	CJH1011	Hitam	Ibe	NULL	100000

Hasil2

```
MariaDB [rental_angga]> select * from mobil where harga_rental != 50000;
```

id_mobil	no_plat	no_mesin	warna	pemilik	peminjam	harga_rental
2	DD 2440 AX	BCS1120	Merah	Ibrahim	Elia	100000
4	DD 2981 JK	UQL1029	Hitam	Ibe	NULL	150000
5	DD 2210 LS	CJH1011	Hitam	Ibe	NULL	100000

Analisis1

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

Analisis2

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

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

Tantangan Login

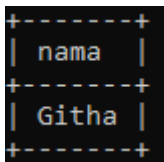
Struktur

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

Contoh

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

Hasil



nama
Githa

Analisis

1. `SELECT` query yang digunakan untuk menampilkan sebuah table
2. `nama` nama kolom yang dimana hanya isi dari kolom ini yang akan di tampilkan
3. `from` query yang digunakan untuk memberikan sebuah tanda ke table yang akan di tampilkan
4. `akun` nama table yang akan di tampilkan
5. `where` query yang digunakan untuk memberikan sebuah kondisi
6. `password="12345"` sebuah kondisi yang telah diberikan. `password` adalah nama kolom, `=` adalah operator, dan `"12345"` adalah nilai.

Summary

jika ingin menampilkan dari hasil seleksi yang dimana hanya ada satu nilai dari satu kolom aka di tampilkan, yaitu dengan cara menggunakan query dengan struktur `select nama_kolom1 from nama_table where nama_kolom2=nilai;`

Catatan

`nama_kolom2` harus kolom yang menggunakan constraint unique

IN

IN

Struktur

```
select * from nama_table where nama_kolom in('nilai_kolom');
```

Contoh

```
select * from mobil where warna in('Hitam','Silver');
```

Hasil

```
MariaDB [rental_angga]> select * from mobil where warna in('merah', 'hitam');
```

id_mobil	no_plat	no_mesin	warna	pemilik	peminjam	harga_rental
1	DD 2650 XY	ACX3560	Hitam	Ibrahim	Afdal	50000
2	DD 2440 AX	BCS1120	Merah	Ibrahim	Elia	100000
4	DD 2981 JK	UQL1029	Hitam	Ibe	NULL	150000
5	DD 2210 LS	CJH1011	Hitam	Ibe	NULL	100000

Analisis

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

Kesimpulan >

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

IN + AND

Struktur

```
select * from nama_table
where nama_kolom in('nilai_kolom')
and nama_kolom = nilai_kolom;
```

Contoh

```
select * from mobil
where warna in('Hitam','Merah')
and harga_rental = 50000;
```

Hasil

```
MariaDB [rental_angga]> select * from mobil
-> where warna in('hitam', 'merah')
-> or harga_rental = 50000;
```

id_mobil	no_plat	no_mesin	warna	pemilik	peminjam	harga_rental
1	DD 2650 XY	ACX3560	Hitam	Ibrahim	Afdal	50000
2	DD 2440 AX	BCS1120	Merah	Ibrahim	Elia	100000
3	B 1611 QC	LSQ1112	Silver	Baim	Anty	50000
4	DD 2981 JK	UQL1029	Hitam	Ibe	NULL	150000
5	DD 2210 LS	CJH1011	Hitam	Ibe	NULL	100000

Analisis

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

Kesimpulan >

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

IN+OR

Struktur

```
select * from nama_table
where nama_kolom in('nilai_kolom')
or nama_kolom = nilai_kolom;
```

Contoh

```
select * from mobil
where warna in('Hitam','Silver')
or harga_rental = 50000;
```

Hasil

```
MariaDB [(none)]> use rental_angga
Database changed
MariaDB [rental_angga]> select * from mobil
-> where warna in('Hitam','Silver')
-> or harga_rental = 50000;
```

id_mobil	no_plat	no_mesin	warna	pemilik	peminjam	harga_rental
1	DD 2650 XY	ACX3560	Hitam	Ibrahim	Afdal	50000
3	B 1611 QC	LSQ1112	Silver	Baim	Anty	50000
4	DD 2981 JK	UQL1029	Hitam	Ibe	NULL	150000
5	DD 2210 LS	CJH1011	Hitam	Ibe	NULL	100000

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

Analisis

- `SELECT *`: Memilih semua kolom dari tabel.
- `FROM mobil`: Menunjukkan bahwa data diambil dari tabel `mobil`.
- `WHERE warna IN ('Hitam', 'Silver') OR harga_rental = 50000`: Menggunakan klausa `WHERE` untuk memfilter baris berdasarkan kondisi bahwa nilai kolom `warna` adalah 'Hitam' atau 'Silver', atau nilai kolom `harga_rental` adalah 50000.

Kesimpulan >

Perintah `SELECT *` digunakan untuk memilih semua kolom dari tabel. Selanjutnya, `FROM mobil` menunjukkan bahwa data diambil dari tabel `mobil`. Klausa `WHERE warna IN ('Hitam', 'Silver') OR harga_rental = 50000` digunakan untuk memfilter baris berdasarkan kondisi bahwa nilai kolom `warna` adalah 'Hitam' atau 'Silver', atau nilai kolom `harga_rental` adalah 50000. Dengan demikian, query ini akan mengembalikan baris-baris di mana nilai kolom `warna` adalah 'Hitam' atau 'Silver', atau nilai kolom `harga_rental` adalah 50000.

IN+AND+OPERATOR

Struktur1

```
select * from nama_table
where nama_kolom in('nilai_kolom')
or nama_kolom > nilai_kolom;
```

Contoh1

```
select * from mobil
where warna in('Hitam','Silver')
or harga_rental > 50000;
```

Hasil1

```
MariaDB [rental_angga]> select * from mobil
-> where warna in('hitam', 'merah')
-> or harga_rental > 50000;
```

id_mobil	no_plat	no_mesin	warna	pemilik	peminjam	harga_rental
1	DD 2650 XY	ACX3560	Hitam	Ibrahim	Afdal	50000
2	DD 2440 AX	BCS1120	Merah	Ibrahim	Elia	100000
4	DD 2981 JK	UQL1029	Hitam	Ibe	NULL	150000
5	DD 2210 LS	CJH1011	Hitam	Ibe	NULL	100000

Analisis1

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

Kesimpulan >

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

Struktur2

```
select * from nama_table
where nama_kolom in('nilai_kolom')
or nama_kolom < nilai_kolom;
```

Contoh2

```
select * from mobil
where warna in('Hitam','Merah')
or harga_rental < 100000;
```

Hasil

```
MariaDB [rental_angga]> select * from mobil
-> where warna in('hitam', 'merah')
-> or harga_rental < 100000;
```

id_mobil	no_plat	no_mesin	warna	pemilik	peminjam	harga_rental
1	DD 2650 XY	ACX3560	Hitam	Ibrahim	Afdal	50000
2	DD 2440 AX	BCS1120	Merah	Ibrahim	Elia	100000
3	B 1611 QC	LSQ1112	Silver	Baim	Anty	50000
4	DD 2981 JK	UQL1029	Hitam	Ibe	NULL	150000
5	DD 2210 LS	CJH1011	Hitam	Ibe	NULL	100000

Analisis

1. `SELECT *` artinya kita akan mengambil semua kolom dari tabel "mobil".
2. `FROM mobil` artinya kita akan mengambil data dari tabel "mobil".
3. `WHERE warna IN ('Hitam', 'Merah')` artinya kita hanya akan mengambil baris-baris di mana nilai kolom "warna" adalah "Hitam" atau "Silver".
4. `OR harga_rental < 100000` artinya kita juga akan mengambil baris-baris di mana nilai kolom "harga_rental" kurang dari 100000.

Kesimpulan >

Perintah `SELECT *` digunakan untuk mengambil semua kolom dari tabel "mobil". Selanjutnya, `FROM mobil` menunjukkan bahwa data diambil dari tabel "mobil". Klausa `WHERE warna IN ('Hitam', 'Merah')` digunakan untuk memfilter baris-baris di mana nilai kolom "warna" adalah "Hitam" atau "Merah". Selain itu, `OR harga_rental < 100000` digunakan untuk juga memfilter baris-baris di mana nilai kolom "harga_rental" kurang dari 100000. Jadi, pernyataan ini akan mengambil baris-baris di mana nilai kolom "warna" adalah "Hitam" atau "Merah", atau di mana nilai kolom "harga_rental" kurang dari 100000.

LIKE

Mencari Awalan

Struktur

```
select * from nama_table
where nama_kolom like 'awalan nilai kolom';
```

Contoh

```
select * from mobil
where pemilik like 'Ib%';
```

Hasil

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

id_mobil	no_plat	no_mesin	warna	pemilik	peminjam	harga_rental
1	DD 2650 XY	ACX3560	Hitam	Ibrahim	Afdal	50000
2	DD 2440 AX	BCS1120	Merah	Ibrahim	Elia	100000
4	DD 2981 JK	UQL1029	Hitam	Ibe	NULL	150000
5	DD 2210 LS	CJH1011	Hitam	Ibe	NULL	100000

4 rows in set (0.338 sec)

Analisis

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

Kesimpulan >

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

Mencari Akhiran

Struktur

```
select * from nama_table  
where nama_kolom like 'ahiran nilai kolom';
```

Contoh

```
select * from mobil  
where pemilik like '%m';
```

Hasil

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

id_mobil	no_plat	no_mesin	warna	pemilik	peminjam	harga_rental
1	DD 2650 XY	ACX3560	Hitam	Ibrahim	Afdal	50000
2	DD 2440 AX	BCS1120	Merah	Ibrahim	Elia	100000
3	B 1611 QC	LSQ1112	Silver	Baim	Anty	50000

3 rows in set (0.001 sec)

Analisis

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

Kesimpulan >

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

Mencari Awalan & Akhiran

Struktur

```
select * from nama_table
```

```
where nama_kolom like 'awalan nilai kolom'
```

Contoh

```
select * from mobil  
where pemilik like 'b%m'
```

Hasil

```
MariaDB [rental_angga]> select * from mobil  
-> where pemilik like 'b%m';  
+-----+-----+-----+-----+-----+-----+-----+  
| id_mobil | no_plat | no_mesin | warna | pemilik | peminjam | harga_rental |  
+-----+-----+-----+-----+-----+-----+-----+  
| 3 | B 1611 QC | LSQ1112 | Silver | Baim | Anty | 50000 |  
+-----+-----+-----+-----+-----+-----+-----+  
1 row in set (0.001 sec)
```

Analisis

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

Kesimpulan >

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

Mencari Berdasarkan Total Karakter

Struktur1

```
select * from nama_table  
where nama_kolom like 'jumlah karakter pada nilai kolom'
```

Contoh1


```
select * from mobil
where pemilik like 'I__';
```

Hasil1

```
MariaDB [rental_angga]> select * from mobil
-> where pemilik like 'I__';
```

id_mobil	no_plat	no_mesin	warna	pemilik	peminjam	harga_rental
4	DD 2981 JK	UQL1029	Hitam	Ibe	NULL	150000
5	DD 2210 LS	CJH1011	Hitam	Ibe	NULL	100000

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

Analisis1

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

Kesimpulan >

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

Struktur2

```
select * from nama_table
      where nama_kolom like 'jumlah karakter pada nilai kolom'
```

Contoh2

```
select * from mobil
where pemilik like '___';
```

Hasil2

```
MariaDB [rental_angga]> select * from mobil  
-> where pemilik like '___';
```

id_mobil	no_plat	no_mesin	warna	pemilik	peminjam	harga_rental
4	DD 2981 JK	UQL1029	Hitam	Ibe	NULL	150000
5	DD 2210 LS	CJH1011	Hitam	Ibe	NULL	100000

2 rows in set (0.001 sec)

Analisis

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

Kesimpulan >

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

Kombinasi

Struktur1

```
select * from nama_table  
where nama_kolom like 'nilai kolom'
```

Contoh1

```
select * from mobil  
where pemilik like '__r%'
```

Hasil1

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

id_mobil	no_plat	no_mesin	warna	pemilik	peminjam	harga_rental
1	DD 2650 XY	ACX3560	Hitam	Ibrahim	Afdal	50000
2	DD 2440 AX	BCS1120	Merah	Ibrahim	Elia	100000

2 rows in set (0.001 sec)

Analisis1

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

Kesimpulan >

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

Struktur2

```
select * from nama_table  
where nama_kolom like 'nilai kolom'
```

Contoh

```
select * from mobil  
where pemilik like '_b%'
```

Hasil

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

id_mobil	no_plat	no_mesin	warna	pemilik	peminjam	harga_rental
1	DD 2650 XY	ACX3560	Hitam	Ibrahim	Afdal	50000
2	DD 2440 AX	BCS1120	Merah	Ibrahim	Elia	100000
4	DD 2981 JK	UQL1029	Hitam	Ibe	NULL	150000
5	DD 2210 LS	CJH1011	Hitam	Ibe	NULL	100000

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

Analisis

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

Kesimpulan >

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

Not Like

Struktur

```
select * from nama_table where nama_kolom not like 'nilai kolom'
```

Contoh

```
select * from mobil where peminjam not like 'A%';
```

Analisis

1. `SELECT *`: Memilih semua kolom dari tabel 'mobil'.
2. `FROM mobil`: Menentukan tabel yang digunakan untuk mengambil data, dalam hal ini tabel 'mobil'.

3. `WHERE peminjam NOT LIKE 'A%'`: Menggunakan klausa `WHERE` untuk memfilter baris-baris yang akan diambil. Kondisi `peminjam NOT LIKE 'A%'` digunakan untuk memeriksa apakah nilai kolom 'peminjam' tidak dimulai dengan huruf 'A'. Operator `NOT LIKE` digunakan untuk memeriksa apakah nilai tidak cocok dengan pola yang diberikan, dalam hal ini, pola 'A%' berarti dimulai dengan 'A'. Jadi, pernyataan ini akan mengembalikan baris-baris di mana nilai kolom 'peminjam' tidak dimulai dengan huruf 'A'.

Kesimpulan >

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

Null & Not Null

Null

Struktur

```
SELECT * FROM nama_table WHERE nama_kolom IS NULL
```

Kode Program

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

Hasil

```
MariaDB [rental_angga]> SELECT * FROM mobil WHERE peminjam IS NULL;
+-----+-----+-----+-----+-----+-----+-----+
| id_mobil | no_plat | no_mesin | warna | pemilik | peminjam | harga_rental |
+-----+-----+-----+-----+-----+-----+-----+
| 4 | DD 2901 JK | UQL1029 | Hitam | Ibe | NULL | 150000 |
| 5 | DD 2210 LS | CJH1011 | Hitam | Ibe | NULL | 100000 |
+-----+-----+-----+-----+-----+-----+-----+
2 rows in set (0.001 sec)
```

Analisis

1. `SELECT *`: Memilih semua kolom dari tabel 'mobil'.
2. `FROM mobil`: Menentukan tabel yang digunakan untuk mengambil data, yaitu tabel 'mobil'.

3. `WHERE peminjam IS NULL` : Menggunakan klausa `WHERE` untuk memfilter baris-baris yang akan diambil. Kondisi `peminjam IS NULL` digunakan untuk memeriksa apakah nilai kolom 'peminjam' adalah `NULL`. Operator `IS NULL` digunakan untuk memeriksa apakah nilai kolom adalah `NULL`.

Kesimpulan >

Perintah `SELECT *` digunakan untuk memilih semua kolom dari tabel 'mobil'. Kemudian, `FROM mobil` menunjukkan tabel yang digunakan untuk mengambil data, yaitu tabel 'mobil'. Selanjutnya, `WHERE peminjam IS NULL` menggunakan klausa `WHERE` untuk memfilter baris-baris yang akan diambil. Kondisi `peminjam IS NULL` digunakan untuk memeriksa apakah nilai kolom 'peminjam' adalah `NULL`. Operator `IS NULL` digunakan untuk memeriksa apakah nilai kolom adalah `NULL`. Dengan demikian, query ini akan mengembalikan baris-baris di mana kolom 'peminjam' memiliki nilai `NULL`.

Not Null

Struktur

```
SELECT * FROM nama_table WHERE nama_kolom IS NOT NULL
```

Kode Program

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

Hasil

```
MariaDB [rental_angga]> SELECT * FROM mobil WHERE peminjam IS NOT NULL;
+-----+-----+-----+-----+-----+-----+-----+
| id_mobil | no_plat | no_mesin | warna | pemilik | peminjam | harga_rental |
+-----+-----+-----+-----+-----+-----+-----+
| 1 | DD 2650 XY | ACX3560 | Hitam | Ibrahim | Afdal | 50000 |
| 2 | DD 2440 AX | BCS1120 | Merah | Ibrahim | Elia | 100000 |
| 3 | B 1611 QC | LSQ1112 | Silver | Baim | Anty | 50000 |
+-----+-----+-----+-----+-----+-----+-----+
3 rows in set (0.001 sec)
```

Analisis

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

Kesimpulan >

Perintah `SELECT *` digunakan untuk memilih semua kolom dari tabel 'mobil'. Kemudian, `FROM mobil` menunjukkan tabel yang digunakan untuk mengambil data, yaitu tabel 'mobil'. Selanjutnya, `ORDER BY pemilik ASC` menggunakan klausa `ORDER BY` untuk mengurutkan hasil query berdasarkan kolom 'pemilik' secara ascending (ASC), yang berarti data akan diurutkan dari nilai paling rendah ke nilai paling tinggi berdasarkan abjad.

Order By

Struktur

```
SELECT * FROM nama_table ORDER BY nama_kolom ASC
```

Kode Program

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

Hasil

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

id_mobil	no_plat	no_mesin	warna	pemilik	peminjam	harga_rental
3	B 1611 QC	LSQ1112	Silver	Baim	Anty	50000
4	DD 2901 JK	UQL1029	Hitam	Ibe	NULL	150000
5	DD 2210 LS	CJH1011	Hitam	Ibe	NULL	100000
1	DD 2650 XY	ACX3560	Hitam	Ibrahim	Afdal	50000
2	DD 2440 AX	BCS1120	Merah	Ibrahim	Elia	100000

5 rows in set (0.001 sec)

Analisis

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

Kesimpulan >

Perintah `SELECT *` digunakan untuk memilih semua kolom dari tabel 'mobil'. Kemudian, `FROM mobil` menunjukkan tabel yang digunakan untuk mengambil data, yaitu tabel 'mobil'. Selanjutnya, `ORDER BY pemilik ASC` menggunakan klausa `ORDER BY` untuk mengurutkan hasil query berdasarkan kolom

'pemilik' secara ascending (ASC), yang berarti data akan diurutkan dari nilai paling rendah ke nilai paling tinggi berdasarkan abjad.

Struktur 2

```
SELECT * FROM Nama_table ORDER BY nama_kolom DESC
```

Kode Program 2

```
SELECT * FROM mobil ORDER BY pemilik DESC
```

Hasil 2

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

id_mobil	no_plat	no_mesin	warna	pemilik	peminjam	harga_rental
2	DD 2440 AX	BCS1120	Merah	Ibrahim	Elia	100000
3	B 1611 QC	LSQ1112	Silver	Baim	Anty	50000
1	DD 2650 XY	ACX3560	Hitam	Ibrahim	Afdal	50000
4	DD 2901 JK	UQL1029	Hitam	Ibe	NULL	150000
5	DD 2210 LS	CJH1011	Hitam	Ibe	NULL	100000

5 rows in set (0.001 sec)

Analisis 2

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

Kesimpulan >

Perintah `SELECT *` digunakan untuk memilih semua kolom dari tabel 'mobil'. Kemudian, `FROM mobil` menunjukkan tabel yang digunakan untuk mengambil data, yaitu tabel 'mobil'. Selanjutnya, `ORDER BY peminjam DESC` menggunakan klausa `ORDER BY` untuk mengurutkan hasil query berdasarkan kolom 'peminjam' secara descending (DESC), yang berarti data akan diurutkan dari nilai paling tinggi ke nilai paling rendah berdasarkan abjad.

Distinct

Struktur


```
SELECT DISTINCT(nama_kolom) FROM nama_table;
```

Kode Program

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

Hasil

```
MariaDB [rental_angga]> SELECT DISTINCT(pemilik) FROM mobil;
+-----+
| pemilik |
+-----+
| Ibrahim |
| Baim    |
| Ibe     |
+-----+
3 rows in set (0.001 sec)
```

Analisis

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

Kesimpulan >

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

Struktur 2

```
SELECT DISTINCT(nama_kolom) FROM nama_table ORDER BY nama_kolom DESC;
```

Kode Program 2

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

Hasil 2

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

Analisis 2

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

Kesimpulan >

Perintah `SELECT DISTINCT(harga_rental)` digunakan untuk memilih nilai unik dari kolom 'harga_rental'. Penggunaan `DISTINCT` menghilangkan nilai yang sama dan hanya mengembalikan nilai yang unik. Selanjutnya, `FROM mobil` menunjukkan bahwa data diambil dari tabel 'mobil'. Kemudian, `ORDER BY harga_rental DESC` mengurutkan hasil berdasarkan kolom 'harga_rental' secara descending, artinya data akan diurutkan dari nilai tertinggi ke terendah berdasarkan harga rental.

CONCAT, CONCAT_WS, AS

Concat

Struktur

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

Kode Program

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

Hasil

```
MariaDB [rental_angga]> SELECT CONCAT(pemilik,warna) FROM mobil;
+-----+
| CONCAT(pemilik,warna) |
+-----+
| IbrahimHitam          |
| IbrahimMerah          |
| BaimSilver             |
| IbeHitam              |
| IbeHitam              |
+-----+
5 rows in set (0.005 sec)
```

Analisis

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

Kesimpulan >

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

Concat_Ws

Struktur

```
SELECT CONCAT_WS("pemisah",nama_kolom) AS nama_hasil FROM nama_table;
```

Kode Program

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

Hasil

```
MariaDB [rental_angga]> SELECT CONCAT_WS("-",no_plat,no_mesin,id_mobil) FROM mobil;
+-----+
| CONCAT_WS("-",no_plat,no_mesin,id_mobil) |
+-----+
| DD 2650 XY-ACX3560-1 |
| DD 2440 AX-BCS1120-2 |
| B 1611 QC-LSQ1112-3 |
| DD 2901 JK-UQL1029-4 |
| DD 2210 LS-CJH1011-5 |
+-----+
5 rows in set (0.001 sec)
```

Analisis

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

Kesimpulan >

Perintah `SELECT CONCAT_WS("+", pemilik, peminjam) AS COLLAB` menggabungkan nilai dari kolom 'pemilik' dan 'peminjam' dengan menggunakan fungsi `CONCAT_WS()`, di mana setiap nilai dipisahkan oleh tanda "+". Hasil penggabungan ini diberi alias 'COLLAB'. Selanjutnya, query tersebut menunjukkan bahwa data diambil dari tabel 'mobil'.

Concat_As

Struktur

```
SELECT CONCAT_WS("pemisah",nama_kolom) AS nama_hasil FROM nama_table;
```

Kode Program

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

Hasil

```
MariaDB [rental_angga]> SELECT * FROM info_no_plat;
+-----+-----+-----+-----+
| id_mobil | no_plat | pemilik | peminjam |
+-----+-----+-----+-----+
| 1 | DD 2650 XY | Ibrahim | Afdal |
| 2 | DD 2440 AX | Ibrahim | Elia |
+-----+-----+-----+-----+
2 rows in set (0.003 sec)
```

Analisis

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

View

Struktur

```
CREATE VIEW nama_table_virtual AS SELECT nama_kolom FROM nama_table WHERE nama_kolom = "nilai_kolom";
```

Kode Program

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

Hasil

```
MariaDB [rental_angga]> CREATE VIEW info_no_plat AS SELECT id_mobil, no_plat, pemilik, peminjam FROM mobil WHERE pemilik = "Ibrahim";
Query OK, 0 rows affected (0.136 sec)
```

Analisis

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

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

Struktur 2

```
SELECT * FROM nama_table_virtual;
```

Kode Program 2

```
SELECT * FROM info_no_plat
```

Hasil 2

```
MariaDB [rental_angga]> SELECT * FROM info_no_plat;
+-----+-----+-----+-----+
| id_mobil | no_plat | pemilik | peminjam |
+-----+-----+-----+-----+
| 1 | DD 2650 XY | Ibrahim | Afdal |
| 2 | DD 2440 AX | Ibrahim | Elia |
+-----+-----+-----+-----+
2 rows in set (0.003 sec)
```

Analisis 2

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

Kesimpulan >

Perintah `CREATE VIEW info_no_plat AS` digunakan untuk membuat view baru dengan nama `info_no_plat`. View ini akan berisi data mobil yang dimiliki oleh pemilik dengan nama "Ibrahim". Query `SELECT id_mobil, no_plat, pemilik, peminjam FROM mobil WHERE pemilik = "Ibrahim";` akan menjadi isi dari view `info_no_plat`. Query ini mengambil kolom `id_mobil`, `no_plat`, `pemilik`, dan `peminjam` dari tabel `mobil` hanya untuk baris-baris di mana `pemilik` adalah "Ibrahim".

Struktur 3

```
DROP VIEW nama_table_virtual;
```

Kode Program 3

```
DROP VIEW info_no_plat;
```

Hasil

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

Analisis

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

Kesimpulan >

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

Tantangan Table Virtual

Soal no 1

No 1 >

buat kan table virtual yang peminjamnya tidak ada dan tampilkan datanya

Kode Program

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

Hasil

```
MariaDB [rental_angga]> select * From mobil_Tanpa_peminjam ;
```

no_plat	peminjam
DD 2901 JK	NULL
DD 2210 LS	NULL

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

Analisis

1. `CREATE VIEW mobil_Tanpa_peminjam AS`: Perintah ini digunakan untuk membuat view baru dengan nama `mobil_Tanpa_peminjam`.
2. `SELECT no_plat, peminjam FROM mobil WHERE peminjam IS NULL;`: Query ini akan mengambil nilai dari kolom `no_plat` dan `peminjam` dari tabel `mobil` dimana nilai `peminjam` adalah `NULL`.

Kesimpulan >

Perintah `CREATE VIEW mobil_Tanpa_peminjam AS` digunakan untuk membuat view baru dengan nama `mobil_Tanpa_peminjam`. View ini akan menampilkan semua data mobil yang tidak memiliki peminjam. Query `SELECT no_plat, peminjam FROM mobil WHERE peminjam IS NULL;` akan mengambil nilai dari kolom `no_plat` dan `peminjam` dari tabel `mobil` dimana nilai `peminjam` adalah `NULL`.

Soal No 2

No 2 >

ganti salah satu data peminjam dari table mobil dengan nilai null dan tampilkan isi data pada table virtual hasil nya akan tiga data yang tampil

Kode Program

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

Hasil


```
MariaDB [rental_angga]> select * From mobil_Tanpa_peminjam ;
```

no_plat	peminjam
DD 2440 AX	NULL
DD 2901 JK	NULL
DD 2210 LS	NULL

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

Analisis

1. `UPDATE mobil` : Perintah ini digunakan untuk mengupdate data dalam tabel `mobil`.
2. `SET peminjam = NULL` : Bagian ini menentukan bahwa nilai kolom `peminjam` akan diubah menjadi `NULL`.
3. `WHERE peminjam = 'Elia'` : Kondisi ini membatasi pengubahan hanya untuk baris-baris dimana nilai kolom `peminjam` adalah 'Elia'.

Kesimpulan >

perintah `UPDATE mobil SET peminjam = NULL WHERE peminjam = 'Elia'` akan mengubah nilai kolom `peminjam` menjadi `NULL` hanya untuk baris-baris di mana nilai kolom `peminjam` adalah 'Elia' dalam tabel `mobil`

Soal No 3

No 3 >

berikan kesimpulan mengapa table virtual ini di buat

Kesimpulan >

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

Dengan membuat view dapat membatasi akses ke data sensitif atau kolom tertentu dari tabel yang mungkin tidak perlu diakses oleh semua pengguna.

Dengan membuat view untuk kueri yang sering digunakan, Anda dapat menghindari pengulangan kode SQL yang sama di beberapa tempat dalam aplikasi atau prosedur penyimpanan.

Agregasi

Menghitung total nilai numerik suatu kolom

Struktur query

```
SELECT SUM(nama_kolom) AS nama_hasil FROM nama_table;
```

Kode Program

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

Hasil

```
MariaDB [rental_angga]> SELECT SUM(harga_rental) AS total_harga FROM mobil;
+-----+
| total_harga |
+-----+
|         450000 |
+-----+
1 row in set (0.137 sec)
```

Analisis

- `SELECT SUM(harga_rental)` : Perintah ini digunakan untuk menghitung jumlah total dari nilai kolom `harga_rental` di dalam tabel `mobil`.
- `AS total_harga` : Menggunakan `AS` untuk memberi alias pada hasil perhitungan, sehingga hasilnya akan disebut sebagai `total_harga`.

Kesimpulan >

Query tersebut menghitung jumlah total dari semua nilai dalam kolom `harga_rental` dari tabel `mobil` dan memberikan hasilnya dengan nama `total_harga`. Hasilnya adalah jumlah total harga rental dari semua mobil yang terdapat dalam tabel.

Menghitung jumlah baris/data, biasanya berdasarkan kriteria tertentu

Struktur query

```
SELECT COUNT(nama_kolom) AS nama_hasil FROM nama_table;
```

Kode Program

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

Hasil

```
MariaDB [rental_angga]> SELECT COUNT(pemilik) AS total_pemilik FROM mobil;
+-----+
| total_pemilik |
+-----+
|              5 |
+-----+
1 row in set (0.001 sec)
```

Analisis

- `SELECT COUNT(pemilik)` : Perintah ini menghitung jumlah baris dalam tabel `mobil` di mana kolom `pemilik` tidak NULL.
- `AS total_pemilik` : Memberi nama alias `total_pemilik` pada hasil perhitungan.

Kesimpulan >

Query tersebut menghitung jumlah total baris dalam tabel `mobil` di mana kolom `pemilik` memiliki nilai yang tidak NULL, dan hasilnya diberikan dengan nama `total_pemilik`. Ini memberikan jumlah total pemilik mobil yang terdaftar dalam tabel.

struktur query 2

```
SELECT COUNT(nama_kolom) AS nama_hasil FROM nama_table;
```

Kode program 2

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

Hasil 2

```
MariaDB [rental_angga]> SELECT COUNT(peminjam) AS total_peminjam FROM mobil;
+-----+
| total_peminjam |
+-----+
|              2 |
+-----+
1 row in set (0.001 sec)
```

Analisis 2

- `SELECT COUNT(peminjam)` : Perintah ini menghitung jumlah baris dalam tabel `mobil` di mana kolom `peminjam` tidak `NULL`.
- `AS total_peminjam` : Memberi nama alias `total_peminjam` pada hasil perhitungan.

Kesimpulan >

Query tersebut menghitung jumlah total baris dalam tabel `mobil` di mana kolom `peminjam` memiliki nilai yang tidak `NULL`, dan hasilnya diberikan dengan nama `total_peminjam`. Ini memberikan jumlah total peminjam mobil yang terdaftar dalam tabel.

Menampilkan nilai terendah

Struktur query

```
SELECT MIN(nama_kolom) AS nama_hasil FROM nama_table;
```

Kode program

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

Hasil

```
MariaDB [rental_angga]> SELECT MIN(harga_rental) AS minimum FROM mobil;
+-----+
| minimum |
+-----+
|   50000 |
+-----+
1 row in set (0.006 sec)
```

Analisis

- `SELECT MIN(harga_rental)` : Perintah ini mengambil nilai minimum dari kolom `harga_rental` di tabel `mobil`.
- `AS minimum` : Memberikan alias `minimum` pada hasil nilai minimum yang diambil.

Kesimpulan >

Query tersebut mengambil nilai minimum dari kolom `harga_rental` di tabel `mobil` dan memberikan hasilnya dengan nama alias `minimum`. Ini akan memberikan informasi tentang harga rental terendah dari semua mobil yang terdaftar dalam tabel.

Menampilkan nilai tertinggi

Struktur query

```
SELECT MAX(nama_kolom) AS nama_hasil FROM nama_table;
```

Kode program

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

Hasil

```
MariaDB [rental_angga]> SELECT MAX(harga_rental) AS maximum FROM mobil;
+-----+
| maximum |
+-----+
| 150000 |
+-----+
1 row in set (0.069 sec)
```

Analisis

- `SELECT MAX(harga_rental)` : Perintah ini mengambil nilai maksimum dari kolom `harga_rental` di tabel `mobil`.
- `AS maximum` : Memberikan alias `maximum` pada hasil nilai maksimum yang diambil.

Kesimpulan >

Query tersebut mengambil nilai maksimum dari kolom `harga_rental` di tabel `mobil` dan memberikan hasilnya dengan nama alias `maximum`. Ini akan memberikan informasi tentang harga rental tertinggi dari semua mobil yang terdaftar dalam tabel.

Menampilkan nilai rata-rata

Struktur query

```
SELECT AVG(nama_kolom) AS nama_hasil FROM nama_table;
```

Kode program

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

Hasil

```
MariaDB [rental_angga]> SELECT AVG(harga_rental) AS rerata FROM mobil;
+-----+
| rerata |
+-----+
| 90000.0000 |
+-----+
1 row in set (0.003 sec)
```

Analisis

- `SELECT AVG(harga_rental)` : Perintah ini menghitung nilai rata-rata dari kolom `harga_rental` di tabel `mobil`.
- `AS rerata` : Memberikan alias `rerata` pada hasil perhitungan rata-rata.

Kesimpulan >

Query tersebut menghitung nilai rata-rata dari kolom `harga_rental` di tabel `mobil` dan memberikan hasilnya dengan nama alias `rerata`. Ini memberikan informasi tentang rata-rata harga rental dari semua mobil yang terdaftar dalam tabel.

Tantangan Groub & Having

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

Jawab

Kode Program

```
SELECT warna, COUNT(*) AS jumlah_mobil
FROM mobil
GROUP BY warna;
```

Hasil

warna	jumlah_mobil
Hitam	3
Merah	1
Silver	1

Analisis

- `SELECT warna, COUNT(*) AS jumlah_mobil`: Memilih kolom `warna` dan menghitung jumlah mobil untuk setiap warna.
- `FROM mobil`: Data diambil dari tabel `mobil`.
- `GROUP BY warna`: Data dikelompokkan berdasarkan warna mobil.

Kesimpulan

Query ini menghitung jumlah mobil yang ada untuk setiap warna yang berbeda di tabel `mobil`.

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

Jawab

Kode Program

```
SELECT pemilik
FROM mobil
GROUP BY pemilik
HAVING COUNT(*) >= 3;
```

Hasil

pemilik	total_pendapatan	jumlah_mobil
Ibrahim	200000	3

Analisis

- `SELECT pemilik`: Memilih kolom `pemilik`.
- `FROM mobil`: Data diambil dari tabel `mobil`.
- `GROUP BY pemilik`: Data dikelompokkan berdasarkan pemilik mobil.
- `HAVING COUNT(*) >= 3`: Memilih hanya pemilik yang memiliki 3 mobil atau lebih.

Kasimpulan

Query ini menampilkan daftar pemilik yang memiliki setidaknya 3 mobil.

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

Jawab

Kode Program

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

Hasil

pemilik	total_pendapatan	jumlah_mobil
Ibrahim	200000	3

Analisis

- `SELECT pemilik, COUNT(*) AS jumlah_mobil`: Memilih kolom `pemilik` dan menghitung jumlah mobil untuk setiap pemilik.
- `FROM mobil`: Data diambil dari tabel `mobil`.
- `GROUP BY pemilik`: Data dikelompokkan berdasarkan pemilik mobil.
- `HAVING COUNT(*) >= 3`: Memilih hanya pemilik yang memiliki 3 mobil atau lebih.

Kesimpulan

Query ini menampilkan daftar pemilik beserta jumlah mobil yang mereka miliki, dengan syarat pemilik tersebut memiliki setidaknya 3 mobil.










4. Mengelompokkan berdasarkan pemilik dan menghitung total pendapatan berdasarkan harga rental

Jawab

Kode Program

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

Hasil

	pemilik	total_pendapatan
<input type="checkbox"/>  Edit  Copy  Delete	Baim	50000
<input type="checkbox"/>  Edit  Copy  Delete	Ibe	250000
<input type="checkbox"/>  Edit  Copy  Delete	Ibrahim	150000

Analisis

- `SELECT pemilik, SUM(harga_rental) AS total_pendapatan`: Memilih kolom `pemilik` dan menghitung total pendapatan berdasarkan harga rental untuk setiap pemilik.
- `FROM mobil`: Data diambil dari tabel `mobil`.
- `GROUP BY pemilik`: Data dikelompokkan berdasarkan pemilik mobil.

Kesimpulan

Query ini menghitung total pendapatan dari harga rental mobil untuk setiap pemilik.




5. Mengelompokkan berdasarkan pemilik dan menampilkan jumlah pemasukan yang mencapai lebih besar atau sama dengan 300k

Jawab

Kode Program

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

Hasil

	pemilik	total_pemasukan
<input type="checkbox"/>	 Edit	 Copy
	 Delete	ibe 300000

Analisis

- `SELECT pemilik, SUM(harga_rental) AS total_pemasukan`: Memilih kolom `pemilik` dan menghitung total pemasukan berdasarkan harga rental untuk setiap pemilik.
- `FROM mobil`: Data diambil dari tabel `mobil`.
- `GROUP BY pemilik`: Data dikelompokkan berdasarkan pemilik mobil.
- `HAVING SUM(harga_rental) >= 300000`: Memilih hanya pemilik yang total pemasukan dari harga rental mobilnya mencapai atau lebih besar dari 300k.

Kesimpulan

Query ini menampilkan daftar pemilik yang total pemasukan dari harga rental mobilnya mencapai atau lebih besar dari 300k.







6. Menampilkan rata-rata pemasukan pemilik mobil dikelompokkan berdasarkan pemiliknya

Jawab

Kode Program

```
SELECT pemilik, AVG(harga_rental) AS rata_rata_pemasukan
FROM mobil
GROUP BY pemilik;
```

Hasil

				pemilik	rata_rata_pemasukan
<input type="checkbox"/>	 Edit	 Copy	 Delete	ibe	100000.0000
<input type="checkbox"/>	 Edit	 Copy	 Delete	Ibrahim	75000.0000

Analisis

- `SELECT pemilik, AVG(harga_rental) AS rata_rata_pemasukan`: Memilih kolom `pemilik` dan menghitung rata-rata pemasukan berdasarkan harga rental untuk setiap pemilik.
- `FROM mobil`: Data diambil dari tabel `mobil`.
- `GROUP BY pemilik`: Data dikelompokkan berdasarkan pemilik mobil.

Kesimpulan

Query ini menghitung rata-rata pemasukan dari harga rental mobil untuk setiap pemilik.

7. Menampilkan pemasukan terbesar dan terkecil dikelompokkan berdasarkan pemilik dan menyeleksi data pemilik yang memiliki jumlah mobil lebih dari 1

Jawab

Kode Program

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

Hasil

		pemilik	pemasukan_terbesar	pemasukan_terkecil
<input type="checkbox"/>	Edit Copy Delete	ibe	150000	50000
<input type="checkbox"/>	Edit Copy Delete	Ibrahim	100000	50000

Analisis

- `SELECT pemilik, MAX(harga_rental) AS pemasukan_terbesar, MIN(harga_rental) AS pemasukan_terkecil`: Memilih kolom `pemilik` dan menghitung pemasukan terbesar dan terkecil berdasarkan harga rental untuk setiap pemilik.
- `FROM mobil`: Data diambil dari tabel `mobil`.
- `GROUP BY pemilik`: Data dikelompokkan berdasarkan pemilik mobil.
- `HAVING COUNT(*) > 1`: Memilih hanya pemilik yang memiliki lebih dari 1 mobil.

Kesimpulan

Query ini menampilkan pemasukan terbesar dan terkecil dari harga rental mobil untuk setiap pemilik yang memiliki lebih dari 1 mobil.

Hasil Tantangan

Data Tabel

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

☐ Pilih Semua Dengan pilihan: Ubah Salin Hapus Ekspor

☐ Tampilkan semua | Jumlah baris: 25 | Saring baris: Cari di tabel ini | Sort by key: Tidak ada

Perubahan Struktur Tabel

Before

Struktur tabel

Tampilan hubungan

#	Nama	Jenis	Penyortiran	Atribut	Tak Ternilai	Bawaan	Komentar	Ekstra	Tindakan
<input type="checkbox"/>	1	IdPeg	int(5)		Tidak	Tidak ada			Ubah Hapus Lainnya
<input type="checkbox"/>	2	Nama	varchar(100)	utf8mb4_general_ci	Tidak	Tidak ada			Ubah Hapus Lainnya
<input type="checkbox"/>	3	NamaKel	varchar(100)	utf8mb4_general_ci	Tidak	Tidak ada			Ubah Hapus Lainnya
<input type="checkbox"/>	4	JK	enum('P', 'L', '', '')	utf8mb4_general_ci	Tidak	Tidak ada			Ubah Hapus Lainnya
<input type="checkbox"/>	5	Alamat	varchar(100)	utf8mb4_general_ci	Tidak	Tidak ada			Ubah Hapus Lainnya
<input type="checkbox"/>	6	Kota	varchar(100)	utf8mb4_general_ci	Tidak	Tidak ada			Ubah Hapus Lainnya
<input type="checkbox"/>	7	Telp	varchar(100)	utf8mb4_general_ci	Tidak	Tidak ada			Ubah Hapus Lainnya
<input type="checkbox"/>	8	NoCab	varchar(4)	utf8mb4_general_ci	Tidak	Tidak ada			Ubah Hapus Lainnya

↑

☐ Pilih Semua

Dengan pilihan:

Jelajahi

Ubah

Hapus

Utama

Unik

Indeks

Spasial

Teks penuh

Add to central columns

Remove from central columns

After

Struktur tabel

Tampilan hubungan

#	Nama	Jenis	Penyortiran	Atribut	Tak Ternilai	Bawaan	Komentar	Ekstra	Tindakan
<input type="checkbox"/>	1	IdPeg	int(5)		Tidak	Tidak ada			Ubah Hapus Lainnya
<input type="checkbox"/>	2	Nama	varchar(100)	utf8mb4_general_ci	Tidak	Tidak ada			Ubah Hapus Lainnya
<input type="checkbox"/>	3	NamaKel	varchar(100)	utf8mb4_general_ci	Tidak	Tidak ada			Ubah Hapus Lainnya
<input type="checkbox"/>	4	JK	varchar(5)	utf8mb4_general_ci	Tidak	Tidak ada			Ubah Hapus Lainnya
<input type="checkbox"/>	5	Alamat	varchar(100)	utf8mb4_general_ci	Tidak	Tidak ada			Ubah Hapus Lainnya
<input type="checkbox"/>	6	Kota	varchar(100)	utf8mb4_general_ci	Tidak	Tidak ada			Ubah Hapus Lainnya
<input type="checkbox"/>	7	Telp	varchar(100)	utf8mb4_general_ci	Tidak	Tidak ada			Ubah Hapus Lainnya
<input type="checkbox"/>	8	NoCab	varchar(4)	utf8mb4_general_ci	Tidak	Tidak ada			Ubah Hapus Lainnya

↑

☐ Pilih Semua

Dengan pilihan:

Jelajahi

Ubah

Hapus

Utama

Unik

Indeks

Spasial

Teks penuh

Add to central columns

Remove from central columns

Perubahan Data Tabel

Before pegawai

← T →

▼ IdPeg

Nama

NamaKel

JK

Alamat

Kota

Telp

NoCab

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

↑

☐ Pilih Semua

Dengan pilihan:

Ubah

Salin

Hapus

Ekspor

☐ Tampilkan semua
 Jumlah baris:
 Saring baris:
 Sort by key:

After pegawai

				IdPeg	Nama	NamaKel	JK	Alamat	Kota	Telp	NoCab
<input type="checkbox"/>				10107	Ahmad Anugrah	Satya	L	Jl. Traktor 4 no 38	Makassar	022-555768	C102
<input type="checkbox"/>				10176	Diah	Wahyuni	P	Jl. Maluku 56	Bandung	022-555934	C101
<input type="checkbox"/>				10246	Dian	Anggraini	P	Jl. Mawar 5	Semarang	024-555102	C103
<input type="checkbox"/>				10252	Antoni	Irawan	L	Jl. A. Yani 15	Jakarta	021-555888	C102
<input type="checkbox"/>				10307	Erik	Andrian	L	Jl. Manggis 155	Semarang	024-555236	C103
<input type="checkbox"/>				10314	Ayu	Rahmadani	P	Jl. Malaka 342	Jakarta	021-555098	C102
<input type="checkbox"/>				10324	Martin	Susanto	L	Jl. Bima 51	Jakarta	021-555785	C102
<input type="checkbox"/>				10407	Rio	Gunawan	L	Jl. Melati 356	Surabaya	031-555231	C104
<input type="checkbox"/>				10415	Susan	Sumantri	P	Jl. Pahlawan 24	Surabaya	031-555120	C104

☐ Pilih Semua
 Dengan pilihan:
 Ubah
 Salin
 Hapus
 Ekspor

Before cabang

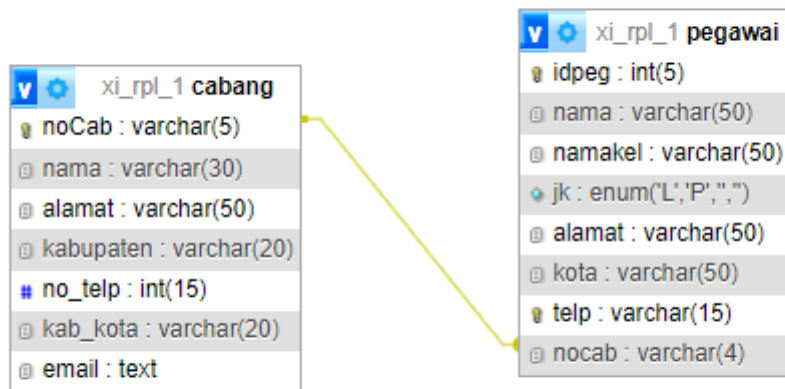
Extra options

After cabang

				NoCab	Nama	Alamat	Kabupaten	No_Telp	Kab_Kota	Email
<input type="checkbox"/>				C101	Muh.Daud Reski Jayadi	Jl. Kalumpang	Kota Makassar	812345678	Makassar	dauddaud123@gmail.com
<input type="checkbox"/>				C102	Muh. Agis	Jl. Galangan Kapal	Kota Makassar	819876543	Makassar	agisagis123@gmail.com
<input type="checkbox"/>				C104	Ahmad Anugrah Satya	Jl. Traktor 4 no38 depan SMPN 26 Makassar di KOMP P...	Kota Makassar	895275291	Makassar	anggaangga123@gmail.com
<input type="checkbox"/>				C103	Nur Reksy Alfatih	Jl. Muhammad Jufri	Kota Makassar	898127634	Makassar	fatirfatir123@gmail.com

☐ Pilih Semua
 Dengan pilihan:
 Ubah
 Salin
 Hapus
 Ekspor

Hasil Relasi



Query Relasi dan Hasil

Kode Program

```
SELECT s.nama AS Nama_Siswa, n.nilai AS Nilai
FROM nilai AS n
INNER JOIN siswa AS s ON s.id = n.id_siswa
WHERE n.nilai > 75;
```

Hasil

Nama_Siswa	Nilai
Alkawsar	90
Angga	80