

SELECT LANJUTAN

Tabel desc_mobil

```
insert into desc_mobil
values (1, "DD 2650 XY","ACX3568","Hitam","Ibrahim","Afdal",50000),(2, "DD 2440
AX","BCS1120","Merah"," Ibrahim","Elia",100000),
(3, "B 1611 QC"," LSQ1112","Silver","Baim","Anty",50000),
(4, "DD 2901 JK","UQL1029","Hitam","Ibe", NULL, 150000),
(5, "DD 2210 LS", "CJH1011","Hitam","Ibe", NULL, 100000) ;
```

AND

Struktur

```
select nama_kolom1,nama_kolom2 from nama_tabel where nama_kolom="nilai_kolom1" and
nama_kolom="nilai_kolom2";
```

Contoh

```
select warna,pemilik from desc_mobil where warna="Hitam" and pemilik="ibrahim";
```

Hasil

```
MariaDB [rental_mobil]> select warna,pemilik from desc_mobil where warna="Hitam" and pemilik="ibrahim";
+-----+-----+
| warna | pemilik |
+-----+-----+
| Hitam | Ibrahim |
+-----+-----+
1 row in set (0.008 sec)

MariaDB [rental_mobil]>
```

Analisis

1. `SELECT warna, pemilik FROM desc_mobil`: Ini adalah klausa SELECT dalam SQL yang digunakan untuk memilih dua kolom, yaitu "warna" dan "pemilik", dari tabel "desc_mobil". Dengan pernyataan ini, kita menginstruksikan basis data untuk mengambil nilai kolom "warna" dan "pemilik" dari setiap baris di tabel "desc_mobil".

2. `WHERE warna="Hitam" AND pemilik="ibrahim"`: Ini adalah klausa WHERE yang digunakan untuk memberikan kriteria pemfilteran pada hasil query. Dalam hal ini, kita membatasi hasil query hanya pada baris-baris di mana nilai kolom "warna" sama dengan "Hitam" dan nilai kolom "pemilik" sama dengan "ibrahim". Kedua kondisi harus terpenuhi secara bersamaan (dengan menggunakan operator logika AND) untuk mendapatkan baris-baris yang sesuai.

Kesimpulan

Kesimpulan, digunakan untuk mengambil informasi dari tabel "desc_mobil". Memilih data mobil berdasarkan kriteria warna "Hitam" dan pemilik "ibrahim". Dengan kata lain, program ini mencari mobil-mobil dengan warna hitam yang dimiliki oleh seseorang yang bernama "ibrahim" dari database mobil

OR

Struktur

```
select nama_kolom1,nama_kolom2 from nama_tabel where nama_kolom1="nilai_kolom1" or
nama_kolom2="nilai_kolom2";
```

Contoh

```
select warna,pemilik from desc_mobil where warna="Hitam" or pemilik="ibrahim";
```

Hasil

```
MariaDB [rental_mobil]> select warna,pemilik from desc_mobil where warna="Hitam" or pemilik="ibrahim";
+-----+-----+
| warna | pemilik |
+-----+-----+
| Hitam | Ibrahim |
| Hitam | Ibe     |
| Hitam | Ibe     |
+-----+-----+
3 rows in set (0.001 sec)

MariaDB [rental_mobil]>
```

Analisis

1. `SELECT warna, pemilik FROM desc_mobil`: Ini adalah klausa SELECT dalam SQL yang digunakan untuk memilih kolom "warna" dan "pemilik" dari tabel "desc_mobil". Dengan

pernyataan ini, kita menginstruksikan basis data untuk mengambil nilai kedua kolom tersebut dari setiap baris di tabel "desc_mobil".

2. `WHERE warna="Hitam" OR pemilik="Ibrahim"`: Ini adalah klausa WHERE yang digunakan untuk memberikan kriteria pemfilteran pada hasil query. Dalam hal ini, kita membatasi hasil query hanya pada baris-baris di mana nilai kolom "warna" sama dengan "Hitam" atau nilai kolom "pemilik" sama dengan "Ibrahim".

Kesimpulan

kesimpulan, digunakan untuk mengambil informasi dari tabel "desc_mobil". Query tersebut memilih data mobil berdasarkan kriteria warna "Hitam" atau pemilik "ibrahim" dari database mobil. Dengan kata lain, program ini mencari mobil-mobil dengan warna hitam atau dimiliki oleh seseorang yang bernama "ibrahim" dari database mobil.

BETWEEN - AND

Struktur

```
select * from nama_tabel where nama_kolom1 between nilai_kolom1 and nilai_kolom2;
```

Contoh

```
select * from desc_mobil where harga_rental between 100000 and 200000;
```

Hasil

"ASET SELECT/tabelbetweend-and.png" could not be found.

Analisis

1. `SELECT * FROM desc_mobil`: Ini adalah klausa SELECT dalam SQL yang digunakan untuk memilih semua kolom (ditandai dengan *) dari tabel "desc_mobil". Dengan pernyataan ini, kita menginstruksikan basis data untuk mengambil semua kolom dari setiap baris di tabel "desc_mobil".
2. `WHERE harga_rental BETWEEN 100000 AND 200000`: Ini adalah klausa WHERE yang digunakan untuk memberikan kriteria pemfilteran pada hasil query. Dalam hal ini, kita membatasi hasil query hanya pada baris-baris di mana nilai kolom "harga_rental" berada dalam rentang antara 100000 dan 200000, termasuk kedua batas tersebut.

Kesimpulan

Kesimpulan dari program ini adalah bahwa hasilnya akan berisi baris-baris dari tabel "desc_mobil" di mana harga rental mobil berada dalam rentang antara 100.000 dan 200.000.

NOT BETWEEN

Struktur

```
select * from nama_tabel where nama_kolom1 not between nilai_kolom1 and  
nilai_kolom2;
```

Contoh

```
select * from desc_mobil where harga_rental not between 100000 and 200000;
```

Hasil

"ASET SELECT/tabelnotbetween.png" could not be found.

Analisis

1. `SELECT * FROM desc_mobil`: Ini adalah klausa SELECT dalam SQL yang digunakan untuk memilih semua kolom (ditandai dengan tanda bintang `*`) dari tabel `desc_mobil`. Dengan pernyataan ini, kita menginstruksikan basis data untuk mengambil semua kolom dari setiap baris di tabel `desc_mobil`.
2. `WHERE harga_rental NOT BETWEEN 100000 AND 200000`: Ini adalah klausa WHERE yang digunakan untuk memberikan kriteria pemfilteran pada hasil query. Dalam hal ini, kita memfilter baris-baris di mana nilai kolom `harga_rental` tidak berada di antara (NOT BETWEEN) 100.000 dan 200.000.

Kesimpulan

kesimpulan dari program ini adalah bahwa hasilnya akan berisi baris-baris dari tabel "desc_mobil" di mana harga rental mobil tidak berada dalam rentang antara 100.000 dan 200.000.

<=

Struktur

```
select * from nama_tabel where nama_kolom1 <= nilaikolom1;
```

Contoh

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

Hasil

```
MariaDB [rental_mobil]> select * from desc_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 (0.001 sec)

MariaDB [rental_mobil]>
```

Analisis

1. `SELECT * FROM desc_mobil`: Ini adalah klausa SELECT dalam SQL yang digunakan untuk memilih semua kolom dari tabel "desc_mobil". Dengan pernyataan ini, kita menginstruksikan basis data untuk mengambil semua kolom dari setiap baris di tabel "desc_mobil".
2. `WHERE harga_rental <= 50000`: Ini adalah klausa WHERE yang digunakan untuk memberikan kriteria pemfilteran pada hasil query. Dalam hal ini, kita membatasi hasil query hanya pada baris-baris di mana nilai kolom "harga_rental" kurang dari atau sama dengan 50000.

Kesimpulan

kesimpulan dari program ini adalah bahwa hasilnya akan berisi baris-baris dari tabel "desc_mobil" di mana harga rental mobil kurang dari atau sama dengan 50.000.

>=

Struktur

```
select * from nama_tabel where nama_kolom1 <= nilaikolom1;
```

Contoh

```
select * from desc_mobil where harga_rental >= 50000;
```

Hasil

```
MariaDB [rental_mobil]> select * from desc_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)

MariaDB [rental_mobil]>
```

Analisis

1. `SELECT * FROM desc_mobil`: Ini adalah klausa SELECT dalam SQL yang digunakan untuk memilih semua kolom dari tabel "desc_mobil". Dengan menggunakan tanda asterisk (*), kita memilih semua kolom yang ada dalam tabel tersebut.
2. `WHERE harga_rental >= 50000`: Ini adalah klausa WHERE yang digunakan untuk memberikan kriteria pemfilteran pada hasil query. Dalam hal ini, kita membatasi hasil query hanya pada baris-baris di mana nilai dalam kolom "harga_rental" setidaknya sebesar 50000. Artinya, hanya baris-baris dengan harga rental setidaknya 50000 yang akan dimasukkan dalam hasil query.

Kesimpulan

kesimpulan dari program ini adalah bahwa hasilnya akan berisi baris-baris dari tabel "desc_mobil" di mana harga rental mobil lebih besar dari atau sama dengan 50.000.

<> atau !=

Struktur

```
select * from nama_tabel where nama_kolom <> nilai_kolom;
```

Contoh

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

Hasil

"ASET SELECT/besar kecil.png" could not be found.

Analisis

1. `SELECT * FROM desc_mobil`: Ini adalah klausa SELECT dalam SQL yang digunakan untuk memilih semua kolom dari tabel "desc_mobil". Dengan menggunakan tanda bintang (*) sebagai argumen SELECT, kita memilih untuk mengambil semua kolom yang ada dalam tabel.
2. `WHERE harga_rental <> 50000`: Ini adalah klausa WHERE yang digunakan untuk memberikan kriteria pemfilteran pada hasil query. Dalam hal ini, kita membatasi hasil query hanya pada baris-baris di mana nilai kolom "harga_rental" tidak sama dengan 50000.

Kesimpulan

kesimpulan dari program ini adalah bahwa hasilnya akan berisi baris-baris dari tabel "desc_mobil" di mana harga rental mobil tidak sama dengan 50.000.

Tantangan

Penjelasan

1. `SELECT nama FROM Frhan`: Ini adalah klausa SELECT dalam SQL yang digunakan untuk memilih kolom "nama" dari tabel "Frhan". Dengan pernyataan ini, kita menginstruksikan basis data untuk mengambil nilai kolom "nama" dari setiap baris di tabel "Frhan".
2. `WHERE nama="FARHAN"`: Ini adalah klausa WHERE yang digunakan untuk memberikan kriteria pemfilteran pada hasil query. Dalam hal ini, kita membatasi hasil query hanya pada baris-baris di mana nilai kolom "nama" sama dengan "FARHAN".

Struktur

```
select nama from Frhan
-> where nama="FARHAN";
```

Hasil

```
MariaDB [rental_mobil]> select nama from Frhan
-> where nama="FARHAN";
+-----+
| nama  |
+-----+
| FARHAN |
+-----+
1 row in set (0.001 sec)
```

Kesimpulan

Jadi, secara keseluruhan, query ini bertujuan untuk mengambil nilai kolom "nama" dari tabel "Frhan" di mana nilai kolom "nama" sama dengan "FARHAN". Ini akan mengembalikan baris-baris yang memenuhi kriteria tersebut.

MATERI IN

IN

struktur

```
select * from nama_tabel where nama_kolom in ("nilai1","nilai2");
```

Contoh

```
select * from pelanggan where nama_depan in ("farhan","radit");
```

Hasil


```

MariaDB [rental_farhan]> select * from pelanggan;
+-----+-----+-----+-----+
| id_pelanggan | nama_depan | nama_belakang | no_telp |
+-----+-----+-----+-----+
|          1 | farhan    | mln           | 088247291854 |
|          2 | rehan     | alfa          | 087868449445 |
|          4 | ilham     | vp            | 085657199460 |
|          5 | radit     | NULL          | NULL          |
+-----+-----+-----+-----+
4 rows in set (0.009 sec)

MariaDB [rental_farhan]> select * from pelanggan where nama_depan in ("farhan","radit");
+-----+-----+-----+-----+
| id_pelanggan | nama_depan | nama_belakang | no_telp |
+-----+-----+-----+-----+
|          1 | farhan    | mln           | 088247291854 |
|          5 | radit     | NULL          | NULL          |
+-----+-----+-----+-----+
2 rows in set (0.001 sec)

MariaDB [rental_farhan]>

```

Analisis

- `SELECT *`: Ini adalah klausa yang digunakan untuk memilih semua kolom dari tabel.
- `FROM pelanggan`: Menunjukkan bahwa data diambil dari tabel bernama "pelanggan".
- `WHERE nama_depan IN ("farhan", "radit")`: Ini adalah klausa yang digunakan untuk memfilter baris yang memiliki nilai kolom "nama_depan" yang sama dengan "farhan" atau "radit".

Kesimpulan

Kesimpulan: Query ini akan mengembalikan semua baris dari tabel "pelanggan" di mana nilai kolom "nama_depan" adalah "farhan" atau "radit". Dengan kata lain, query ini akan memberikan daftar pelanggan yang memiliki nama depan "farhan" atau "radit".

IN & AND

Struktur

```

select * from Nama_tabel
  where nama_kolom in ("nilai_kolom1","nilai_kolom2")
  and nama_kolom = nilai_kolom;

```

Contoh

```
select * from desc_mobil
  where warna in ("Hitam","Silver")
 and harga_rental = 50000;
```

Hasil

```
MariaDB [rental_mobil]> select * from desc_mobil;
+-----+-----+-----+-----+-----+-----+-----+
| 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 (0.003 sec)

MariaDB [rental_mobil]> select * from desc_mobil
-> where warna in ("Hitam","Silver")
-> and harga_rental = 50000;
+-----+-----+-----+-----+-----+-----+-----+
| id_mobil | no_plat | no_mesin | warna | pemilik | peminjam | harga_rental |
+-----+-----+-----+-----+-----+-----+-----+
| 1 | DD 2650 XY | ACX3568 | Hitam | Ibrahim | Afdal | 50000 |
| 3 | B 1611 QC | LSQ1112 | Silver | Baim | Anty | 50000 |
+-----+-----+-----+-----+-----+-----+-----+
2 rows in set (0.068 sec)

MariaDB [rental_mobil]> _
```

Analisis

- `SELECT *`: Ini adalah klausa yang digunakan untuk memilih semua kolom dari tabel.
- `FROM desc_mobil`: Menunjukkan bahwa data diambil dari tabel bernama "desc_mobil".
- `WHERE warna IN ("Hitam", "Silver")`: Ini adalah klausa yang digunakan untuk memfilter baris yang memiliki nilai kolom "warna" yang sama dengan "Hitam" atau "Silver".
- `AND harga_rental = 50000`: Ini adalah klausa yang digunakan untuk memastikan bahwa nilai kolom "harga_rental" adalah 50000.

Kesimpulan

Kesimpulan: Query ini akan mengembalikan semua baris dari tabel "desc_mobil" di mana warnanya adalah "Hitam" atau "Silver" dan harga rentalnya adalah 50000. Dengan kata lain, query ini akan memberikan daftar mobil yang tersedia untuk disewa dengan warna Hitam atau Silver dengan harga rental 50000.

IN & OR

Struktur

```
select * from nama_tabel
  where nama_kolom1 in ("nilai_kolom1","nilai_kolom2")
 or nama_kolom2 = nilai_kolom;
```

Contoh

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

Hasil

"IN & OR.png" could not be found.

(ASET%20SELECT/IN&OR.png)

Analisis

- **SELECT ***: Ini adalah klausa yang digunakan untuk memilih semua kolom dari tabel.
- **FROM desc_mobil**: Menunjukkan bahwa data diambil dari tabel bernama "desc_mobil".
- **WHERE warna IN ("Hitam", "Silver")**: Ini adalah klausa yang digunakan untuk memfilter baris yang memiliki nilai kolom "warna" yang sama dengan "Hitam" atau "Silver".
- **OR harga_rental = 50000**: Ini adalah klausa yang digunakan untuk memastikan bahwa nilai kolom "harga_rental" adalah 50000. Karena menggunakan operator OR, baris akan dipilih jika nilai kolom "warna" adalah Hitam atau Silver, atau jika harga rentalnya adalah 50000.

Kesimpulan

Kesimpulan: Query ini akan mengembalikan semua baris dari tabel "desc_mobil" di mana warnanya adalah "Hitam" atau "Silver", serta baris di mana harga rentalnya adalah 50000. Dengan kata lain, query ini akan memberikan daftar mobil yang tersedia untuk disewa dengan warna Hitam atau Silver, serta mobil dengan harga rental 50000, tanpa memperhatikan warnanya.

IN + AND + OPERATOR

Struktur

```
select * from nama_tabel
  where nama_kolom in ("nilai_kolom1","nilai_kolom2")
 and nama_kolom > nilai_kolom;
```

```
select * from nama_tabel
  where nama_kolom in ("nilai_kolom1","nilai_kolom2")
 and nama_kolom < nilai_kolom;
```

Contoh

```
select * from desc_mobil
  where warna in ("Hitam","Silver")
 and harga_rental > 50000;
```

```
select * from desc_mobil
  where warna in ("Hitam","Silver")
 and harga_rental < 100000;
```

Hasil

```
MariaDB [rental_mobil]> select * from desc_mobil
-> where warna in ("Hitam","Silver")
-> and harga_rental > 50000;
```

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

```
2 rows in set (0.030 sec)
```

```
MariaDB [rental_mobil]>
```

(ASET%20SELECT/inandoperator.png)

"in and operator 2.png" could not be found.

(ASET%20SELECT/inandoperator2.png)

Analisis

1.

- **SELECT ***: Ini adalah klausa yang digunakan untuk memilih semua kolom dari tabel.

- `FROM desc_mobil` : Menunjukkan bahwa data diambil dari tabel bernama "desc_mobil".
- `WHERE warna IN ("Hitam", "Silver")` : Ini adalah klausa yang digunakan untuk memfilter baris yang memiliki nilai kolom "warna" yang sama dengan "Hitam" atau "Silver".
- `AND harga_rental > 50000` : Ini adalah klausa yang digunakan untuk memastikan bahwa nilai kolom "harga_rental" lebih besar dari 50000.

2.

- `SELECT *` : Ini adalah klausa yang digunakan untuk memilih semua kolom dari tabel.
- `FROM desc_mobil` : Menunjukkan bahwa data diambil dari tabel bernama "desc_mobil".
- `WHERE warna IN ("Hitam", "Silver")` : Ini adalah klausa yang digunakan untuk memfilter baris yang memiliki nilai kolom "warna" yang sama dengan "Hitam" atau "Silver".
- `AND harga_rental < 100000` : Ini adalah klausa yang digunakan untuk memastikan bahwa nilai kolom "harga_rental" lebih kecil dari 100000.

Keismpulan

1. Kesimpulan: Query ini akan mengembalikan semua baris dari tabel "desc_mobil" di mana warnanya adalah "Hitam" atau "Silver", dan harga rental mobil tersebut lebih besar dari 50000. Dengan kata lain, query ini akan memberikan daftar mobil dengan warna Hitam atau Silver yang memiliki harga rental di atas 50000.
2. Kesimpulan: Query ini akan mengembalikan semua baris dari tabel "desc_mobil" di mana warnanya adalah "Hitam" atau "Silver", dan harga rental mobil tersebut lebih kecil dari 100000. Dengan kata lain, query ini akan memberikan daftar mobil dengan warna Hitam atau Silver yang memiliki harga rental di atas 100000.

LIKE

Mencari Awalan

Struktur

```
select * from nama_tabel
      where nama_kolom like 'nilai_kolom';
```

Contoh

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

Hasil

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

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

```
3 rows in set (0.071 sec)
```

```
MariaDB [rental_mobil]> _
```

(ASET%20SELECT/likemencariawalan.png)

Analisis

- `SELECT *`: Ini adalah klausa yang digunakan untuk memilih semua kolom dari tabel.
- `FROM desc_mobil`: Menunjukkan bahwa data diambil dari tabel bernama "desc_mobil".
- `WHERE pemilik LIKE 'Ib%'`: Ini adalah klausa yang digunakan untuk memfilter baris yang memiliki nilai kolom "pemilik" dimulai dengan 'Ib' dan diikuti oleh nol atau lebih karakter lainnya. Penggunaan operator LIKE dengan '%', yang merupakan wildcard, mengindikasikan bahwa pencarian dilakukan terhadap string yang dimulai dengan 'Ib' dan diikuti oleh karakter apa pun.

Kesimpulan

Kesimpulan: Query ini akan mengembalikan semua baris dari tabel "desc_mobil" di mana nama pemilik mobil dimulai dengan huruf 'Ib'. Dengan kata lain, query ini akan memberikan daftar mobil yang dimiliki oleh pemilik yang namanya dimulai dengan 'Ib'.

Mencari Akhiran

Struktur

```
select * from nama_tabel  
where nama_kolom like 'nilai_kolom';
```

Contoh

```
select * from desc_mobil
```

```
where pemilik like '%m';
```

Hasil

“like mencari akhiran.png” could not be found.

(ASET%20SELECT/likemencariakhiran.png)

Analisis

- `SELECT *`: Ini adalah klausa yang digunakan untuk memilih semua kolom dari tabel.
- `FROM desc_mobil`: Menunjukkan bahwa data diambil dari tabel bernama "desc_mobil".
- `WHERE pemilik LIKE '%m'`: Ini adalah klausa yang digunakan untuk memfilter baris yang memiliki nilai kolom "pemilik" berakhir dengan huruf 'm'. Penggunaan operator LIKE dengan '%', yang merupakan wildcard, mengindikasikan bahwa pencarian dilakukan terhadap string yang diakhiri dengan 'm' dan dimulai oleh nol atau lebih karakter lainnya.

Kesimpulan

Kesimpulan: Query ini akan mengembalikan semua baris dari tabel "desc_mobil" di mana nama pemilik mobil berakhir dengan huruf 'm'. Dengan kata lain, query ini akan memberikan daftar mobil yang dimiliki oleh pemilik yang namanya berakhir dengan huruf 'm'.

Mencari awalan dan akhiran

Struktur

```
select * from nama_tabel
where nama_kolom like 'nilai_kolom';
```

Contoh

```
select * from desc_mobil
where pemilik like 'b%m';
```

Hasil

“mencari awal akhir.png” could not be found.

(ASET%20SELECT/mencariawalakhir.png)

Analisis

- `SELECT *`: Ini adalah klausa yang digunakan untuk memilih semua kolom dari tabel.
- `FROM desc_mobil`: Menunjukkan bahwa data diambil dari tabel bernama "desc_mobil".
- `WHERE pemilik LIKE 'b%m'`: Ini adalah klausa yang digunakan untuk memfilter baris yang memiliki nilai kolom "pemilik" dimulai dengan huruf 'b', diikuti oleh nol atau lebih karakter, dan diakhiri dengan huruf 'm'. Penggunaan operator LIKE dengan '%', yang merupakan wildcard, mengindikasikan bahwa pencarian dilakukan terhadap string yang dimulai dengan 'b' dan diakhiri dengan 'm', dengan karakter apa pun di antaranya.

Kesimpulan

Kesimpulan: Query ini akan mengembalikan semua baris dari tabel "desc_mobil" di mana nama pemilik mobil dimulai dengan huruf 'b' dan diakhiri dengan huruf 'm', dengan karakter apa pun di antara keduanya. Dengan kata lain, query ini akan memberikan daftar mobil yang dimiliki oleh pemilik yang namanya dimulai dengan 'b' dan berakhir dengan 'm'.

Mencari Berdasarkan Total Karakter

Struktur

```
select * from nama_tabel
where nama_kolom like 'nilai_kolom';
```

Contoh

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

```
select * from desc_mobil
where pemilik like '____';
```

Hasil


```

MariaDB [rental_mobil]> select * from desc_mobil
-> where pemilik like 'I__';
+-----+-----+-----+-----+-----+-----+-----+
| id_mobil | no_plat | no_mesin | warna | pemilik | peminjam | harga_rental |
+-----+-----+-----+-----+-----+-----+-----+
| 4 | DD 2901 JK | UQL1029 | Hitam | Ibe | NULL | 150000 |
| 5 | DD 2210 LS | CJH1011 | Hitam | Ibe | NULL | 100000 |
+-----+-----+-----+-----+-----+-----+-----+
2 rows in set (0.001 sec)

MariaDB [rental_mobil]> 

```

(ASET%20SELECT/mencariberdasarkantotalkarakter.png)

"mencari berdasarkan total karakter 2.png" could not be found.

(ASET%20SELECT/mencariberdasarkantotalkarakter2.png)

Analisis

1.

- `SELECT *`: Ini adalah klausa yang digunakan untuk memilih semua kolom dari tabel.
- `FROM desc_mobil`: Menunjukkan bahwa data diambil dari tabel bernama "desc_mobil".
- `WHERE pemilik LIKE 'I__'`: Ini adalah klausa yang digunakan untuk memfilter baris yang memiliki nilai kolom "pemilik" dimulai dengan huruf 'I' dan diikuti oleh dua karakter lainnya. Penggunaan tanda garis bawah '_' sebagai wildcard menunjukkan bahwa dua karakter setelah huruf 'I' bisa apa saja, asalkan ada dua karakter tersebut.

2.

- `SELECT *`: Ini adalah klausa yang digunakan untuk memilih semua kolom dari tabel.
- `FROM desc_mobil`: Menunjukkan bahwa data diambil dari tabel bernama "desc_mobil".
- `WHERE pemilik LIKE '____'`: Ini adalah klausa yang digunakan untuk memfilter baris yang memiliki nilai kolom "pemilik" memiliki panjang tepat empat karakter. Penggunaan tanda garis bawah '_' sebagai wildcard menunjukkan bahwa empat karakter harus ada, dengan karakter apa pun di tempat masing-masing.

Kesimpulan

1. Query ini akan mengembalikan semua baris dari tabel "desc_mobil" di mana nama pemilik mobil memiliki panjang tepat tiga karakter, dimulai dengan huruf 'I'. Dengan kata lain, query ini akan memberikan daftar mobil yang dimiliki oleh pemilik yang namanya dimulai dengan 'I' dan terdiri dari total tiga karakter.

2. Query ini akan mengembalikan semua baris dari tabel "desc_mobil" di mana nama pemilik mobil memiliki panjang tepat empat karakter. Dengan kata lain, query ini akan memberikan daftar mobil yang dimiliki oleh pemilik yang namanya terdiri dari total empat karakter.

Kombinasi

Struktur

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

Contoh

```
select * from desc_mobil
  where pemilik like '__r%';
```

```
select * from desc_mobil
  where pemilik like '_b%';
```

Hasil

```
MariaDB [rental_mobil]> select * from desc_mobil
-> where pemilik like '__r%';
+-----+-----+-----+-----+-----+-----+-----+
| id_mobil | no_plat   | no_mesin | warna | pemilik | peminjam | harga_rental |
+-----+-----+-----+-----+-----+-----+-----+
|          1 | DD 2650 XY | ACX3568  | Hitam | Ibrahim | Afdal    |          50000 |
+-----+-----+-----+-----+-----+-----+-----+
1 row in set (0.000 sec)

MariaDB [rental_mobil]> _
```

(ASET%20SELECT/kombinasi.png)

"kombinasi 2.png" could not be found.

(ASET%20SELECT/kombinasi2.png)

Analisis

- 1.

- **SELECT ***: Ini adalah klausa yang digunakan untuk memilih semua kolom dari tabel.

- `FROM desc_mobil` : Menunjukkan bahwa data diambil dari tabel bernama "desc_mobil".
- `WHERE pemilik LIKE '__r%'` : Ini adalah klausa yang digunakan untuk memfilter baris yang memiliki nilai kolom "pemilik" memiliki panjang tepat tiga karakter, diikuti oleh huruf 'r', dan diikuti oleh nol atau lebih karakter lainnya. Penggunaan tanda garis bawah '_' sebagai wildcard menunjukkan bahwa dua karakter harus ada sebelum huruf 'r', dan tanda '%' menunjukkan bahwa karakter setelah 'r' bisa apa saja.

2.

- `SELECT *` : Ini adalah klausa yang digunakan untuk memilih semua kolom dari tabel.
- `FROM desc_mobil` : Menunjukkan bahwa data diambil dari tabel bernama "desc_mobil".
- `WHERE pemilik LIKE '_b%'` : Ini adalah klausa yang digunakan untuk memfilter baris yang memiliki nilai kolom "pemilik" memiliki panjang tepat dua karakter, diikuti oleh huruf 'b', dan diikuti oleh nol atau lebih karakter lainnya. Penggunaan tanda garis bawah '_' sebagai wildcard menunjukkan bahwa satu karakter harus ada sebelum huruf 'b', dan tanda '%' menunjukkan bahwa karakter setelah 'b' bisa apa saja.

Kesimpulan

1. Kesimpulan: Query ini akan mengembalikan semua baris dari tabel "desc_mobil" di mana nama pemilik mobil memiliki panjang tepat tiga karakter, diikuti oleh huruf 'r', dan diikuti oleh nol atau lebih karakter lainnya. Dengan kata lain, query ini akan memberikan daftar mobil yang dimiliki oleh pemilik yang namanya terdiri dari total tiga karakter, dimulai dengan dua karakter apa pun, diikuti oleh 'r', dan diikuti oleh karakter apa pun.
2. Kesimpulan: Query ini akan mengembalikan semua baris dari tabel "desc_mobil" di mana nama pemilik mobil memiliki panjang tepat dua karakter, dimulai dengan satu karakter apa pun, diikuti oleh huruf 'b', dan diikuti oleh karakter apa pun. Dengan kata lain, query ini akan memberikan daftar mobil yang dimiliki oleh pemilik yang namanya terdiri dari total dua karakter, dimulai dengan karakter apa pun, diikuti oleh 'b', dan diikuti oleh karakter apa pun.

Not Like

Struktur

```
select * from nama_tabel where nama_kolom not like 'nilai_kolom';
```

Contoh

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

Hasil

"not like.png" could not be found.

(ASET%20SELECT/notlike.png)

Analisis

- `SELECT *`: Ini adalah klausa yang digunakan untuk memilih semua kolom dari tabel.
- `FROM desc_mobil`: Menunjukkan bahwa data diambil dari tabel bernama "desc_mobil".
- `WHERE peminjam NOT LIKE 'A%'`: Ini adalah klausa yang digunakan untuk memfilter baris yang memiliki nilai kolom "peminjam" tidak dimulai dengan huruf 'A'. Penggunaan operator NOT LIKE dan '%', yang merupakan wildcard, mengindikasikan bahwa pencarian dilakukan terhadap string yang tidak dimulai dengan 'A'.

Kesimpulan

Kesimpulan: Query ini akan mengembalikan semua baris dari tabel "desc_mobil" di mana nama peminjam mobil tidak dimulai dengan huruf 'A'. Dengan kata lain, query ini akan memberikan daftar mobil yang belum dipinjam oleh orang yang namanya dimulai dengan 'A'.

Null & Not Null

Mencari data kosong

Struktur

```
select * from nama_tabel where nama_kolom is null;
```

Contoh

```
select * from desc_mobil where peminjam is null;
```

Hasil

```
MariaDB [rental_mobil]> select * from desc_mobil where peminjam is null;
```

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

```
2 rows in set (0.005 sec)
```

(ASET%20SELECT/mencaridatakosong.png)

Analisis

1. `SELECT *`: Perintah ini digunakan untuk memilih semua kolom dari tabel.
2. `FROM desc_mobil`: Ini menunjukkan bahwa data akan diambil dari tabel bernama `desc_mobil`.
3. `WHERE peminjam IS NULL`: Ini adalah klausal WHERE yang membatasi hasil hanya pada baris di mana nilai kolom `peminjam` adalah NULL.

Kesimpulan

Kesimpulan: Perintah SQL ini akan mengembalikan semua baris dari tabel `desc_mobil` di mana kolom `peminjam` memiliki nilai NULL. Ini bermanfaat untuk menemukan mobil yang tidak sedang dipinjamkan kepada siapa pun pada saat tersebut

Mencari data yang tidak kosong

Struktur

```
select * from nama_tabel where nama_kolom is not null;
```

Contoh

```
select * from desc_mobil where peminjam is not null;
```

Hasil

```
MariaDB [rental_mobil]> select * from desc_mobil where peminjam is not null;
```

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

```
3 rows in set (0.011 sec)
```

(ASET%20SELECT/mencaridatayangtidakkosong.png)

Analisis

1. `SELECT *`: Perintah ini digunakan untuk memilih semua kolom dari tabel.
2. `FROM desc_mobil`: Ini menunjukkan bahwa data akan diambil dari tabel bernama `desc_mobil`.
3. `WHERE peminjam IS NOT NULL`: Ini adalah klausul WHERE yang membatasi hasil hanya pada baris di mana nilai kolom `peminjam` tidak NULL.

Kesimpulan

Kesimpulan: Perintah SQL ini akan mengembalikan semua baris dari tabel `desc_mobil` di mana kolom `peminjam` memiliki nilai yang tidak NULL. Ini bermanfaat untuk menemukan mobil yang sedang dipinjamkan kepada seseorang pada saat tersebut.

Order by

Mengurutkan data dari data terkecil

Struktur

```
select * from nama_tabel order by nama_kolom asc;
```

Contoh

```
select * from desc_mobil order by peminjam asc;
```

Hasil

```
MariaDB [rental_mobil]> select * from desc_mobil order by peminjam asc;
```

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
1	DD 2650 XY	ACX3568	Hitam	Ibrahim	Afdal	50000
3	B 1611 QC	LSQ1112	Silver	Baim	Anty	50000
2	DD 2440 AX	BCS1120	Merah	Ibrahim	Elia	100000

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

(ASET%20SELECT/mengurutkandataridataterkecil.png)

Analisis

1. `SELECT *`: Perintah ini digunakan untuk memilih semua kolom dari tabel.
2. `FROM desc_mobil`: Ini menunjukkan bahwa data akan diambil dari tabel bernama `desc_mobil`.
3. `ORDER BY peminjam ASC`: Ini adalah klausal `ORDER BY` yang digunakan untuk mengurutkan baris-baris hasil query berdasarkan kolom `peminjam` secara menaik (dari nilai yang paling kecil ke nilai yang paling besar).

Kesimpulan

Kesimpulan: Perintah SQL ini akan mengembalikan semua baris dari tabel `desc_mobil` yang diurutkan berdasarkan nilai kolom `peminjam` secara menaik. Dengan kata lain, baris-baris dengan nilai `peminjam` yang lebih kecil akan muncul lebih dulu dalam hasil query.

Mengurutkan data dari data terbesar

Struktur

```
select * from nama_tabel order by nama_kolom desc;
```

Contoh

```
select * from desc_mobil order by peminjam desc;
```

Hasil

"mengurutkan data dari data terbesar.png" could not be found.

(ASET%20SELECT/mengurutkandataridataterbesar.png)

Analisis

1. `SELECT *`: Perintah ini digunakan untuk memilih semua kolom dari tabel.
2. `FROM desc_mobil`: Ini menunjukkan bahwa data akan diambil dari tabel bernama `desc_mobil`.
3. `ORDER BY peminjam DESC`: Ini adalah klausal ORDER BY yang digunakan untuk mengurutkan baris-baris hasil query berdasarkan kolom `peminjam` secara menurun (dari nilai yang paling besar ke nilai yang paling kecil).

Kesimpulan

Kesimpulan: Perintah SQL ini akan mengembalikan semua baris dari tabel `desc_mobil` yang diurutkan berdasarkan nilai kolom `peminjam` secara menurun. Dengan kata lain, baris-baris dengan nilai `peminjam` yang lebih besar akan muncul lebih dulu dalam hasil query.

Distinct

Struktur

```
select distinct (nama_kolom) from nama_tabel;
```

```
select distinct (nama_kolom) from nama_tabel order by nama_kolom desc;
```

Contoh

```
select distinct (pemilik) from desc_mobil;
```

```
select distinct (harga_rental) from desc_mobil order by harga_rental desc;
```

Hasil

```
MariaDB [rental_mobil]> select distinct (pemilik) from desc_mobil;
+-----+
| pemilik |
+-----+
| Ibrahim |
| Ibrahim |
| Baim    |
| Ibe     |
+-----+
4 rows in set (0.003 sec)
```

(ASET%20SELECT/distinc.png)

"distinc 1.png" could not be found.

(ASET%20SELECT/distinc1.png)

Analisis

1. `SELECT DISTINCT pemilik` : Perintah ini digunakan untuk memilih nilai unik dari kolom `pemilik`.
2. `FROM desc_mobil` : Ini menunjukkan bahwa data akan diambil dari tabel bernama `desc_mobil`.
3. `SELECT DISTINCT harga_rental` : Perintah ini digunakan untuk memilih nilai unik dari kolom `harga_rental`.
4. `FROM desc_mobil` : Ini menunjukkan bahwa data akan diambil dari tabel bernama `desc_mobil`.
5. `ORDER BY harga_rental DESC` : Ini adalah klausul ORDER BY yang digunakan untuk mengurutkan nilai-nilai `harga_rental` secara menurun.

Kesimpulan

Kesimpulan: Perintah SQL ini akan mengembalikan daftar nilai unik dari kolom `pemilik` dalam tabel `desc_mobil`. Dengan kata lain, jika ada beberapa baris dengan nilai `pemilik` yang sama, perintah ini hanya akan mengembalikan satu nilai untuk setiap nilai `pemilik` yang berbeda. Hal ini berguna untuk mengetahui siapa saja yang merupakan pemilik mobil yang terdaftar dalam tabel tersebut.

Kesimpulan: Perintah SQL ini akan mengembalikan daftar nilai unik dari kolom `harga_rental` dalam tabel `desc_mobil`, diurutkan dari nilai tertinggi ke terendah. Ini memungkinkan pengguna untuk melihat daftar harga sewa yang unik yang tersedia untuk mobil-mobil yang terdaftar dalam tabel tersebut.

Concat,Concat_WS,AS

Menggabungkan Kolom Tanpa Pemisah

Struktur

```
select concat (nama_kolom,nama_kolom) from nama_tabel;
```

Contoh

```
select concat (pemilik,warna) from desc_mobil;
```

Hasil

```
MariaDB [rental_mobil]> select concat (pemilik,warna) from desc_mobil;
+-----+
| concat (pemilik,warna) |
+-----+
| IbrahimHitam           |
| IbrahimMerah           |
| BaimSilver             |
| IbeHitam               |
| IbeHitam               |
+-----+
5 rows in set (0.026 sec)
```

(ASET%20SELECT/kolomtanpapemisah.png)

Analisis

Perintah SQL di atas menggunakan fungsi CONCAT untuk menggabungkan nilai dari dua kolom, yaitu "pemilik" dan "warna", dari tabel "desc_mobil". Ini akan menghasilkan sebuah kolom baru yang berisi hasil penggabungan nilai dari kedua kolom tersebut.

Kesimpulan

Kesimpulan dari perintah ini adalah untuk menampilkan informasi yang mengkombinasikan nama pemilik mobil dan warna mobil dari tabel "desc_mobil". Dengan demikian, output dari perintah ini akan memberikan nilai gabungan dari kolom "pemilik" dan "warna" untuk setiap baris dalam tabel tersebut.

Menggabungkan Kolom Dengan Pemisah

Struktur

```
select concat_ws("-",nama_kolom,nama_kolom,nama_kolom) from nama_tabel;
```

Contoh

```
select concat_ws("-",no_plat,no_mesin,id_mobil) from desc_mobil;
```

Hasil

"kolom dengan pemisah.png" could not be found.

(ASET%20SELECT/kolomdenganpemisah.png)

Analisis

Perintah SQL di atas menggunakan fungsi CONCAT_WS untuk menggabungkan nilai dari tiga kolom, yaitu "no_plat", "no_mesin", dan "id_mobil", dari tabel "desc_mobil". Fungsi CONCAT_WS memungkinkan kita untuk menentukan pemisah yang akan digunakan di antara nilai-nilai yang digabungkan.

Dalam hal ini, tanda "-" digunakan sebagai pemisah di antara nilai-nilai yang digabungkan. Oleh karena itu, output dari perintah ini akan memberikan nilai gabungan dari ketiga kolom tersebut, dengan tanda "-" di antara setiap nilai.

Kesimpulan

Kesimpulan dari perintah ini adalah untuk menampilkan informasi yang mengkombinasikan nomor plat, nomor mesin, dan ID mobil dari tabel "desc_mobil", dengan menggunakan tanda "-" sebagai pemisah. Dengan demikian, output dari perintah ini akan memberikan nilai gabungan yang unik untuk setiap baris dalam tabel tersebut, mencakup ketiga kolom yang dimaksud.

Memberikan Nama Kolom Alias

Struktur

```
select concat_ws("+",nama_kolom,nama_kolom) as nama_kolom from nama_tabel;
```

Contoh

```
select concat_ws("+",pemilik,peminjam) as collab from desc_mobil;
```

Hasil

```
MariaDB [rental_mobil]> select concat_ws("+",pemilik,peminjam) as collab from desc_mobil;
+-----+
| collab |
+-----+
| Ibrahim+Afdal |
| Ibrahim+Elia |
| Baिम+Anty |
| Ibe |
| Ibe |
+-----+
5 rows in set (0.031 sec)
```

(ASET%20SELECT/namakolomalias.png)

Analisis

Perintah SQL di atas menggunakan fungsi CONCAT_WS untuk menggabungkan nilai dari dua kolom, yaitu "pemilik" dan "peminjam", dari tabel "desc_mobil". Fungsi CONCAT_WS memungkinkan untuk menggabungkan nilai-nilai tersebut dengan menggunakan tanda "+" sebagai pemisah di antara mereka.

Kolom hasil akan diberi alias "collab" dengan menggunakan klausa AS. Jadi, output dari perintah ini akan menghasilkan satu kolom baru yang berisi nilai gabungan dari kolom "pemilik" dan "peminjam", dipisahkan oleh tanda "+", untuk setiap baris dalam tabel "desc_mobil".

Kesimpulan

Kesimpulan dari perintah ini adalah untuk membuat kolom baru yang menampilkan kolaborasi antara pemilik dan peminjam mobil, dengan nilai-nilai tersebut digabungkan dan dipisahkan oleh tanda "+". Hal ini mungkin berguna untuk melacak hubungan antara pemilik dan peminjam mobil dalam suatu konteks tertentu.

View

Membuat Tabel Virtual

Struktur

```
create view nama_kolom as
  select nama_kolom, nama_kolom, nama_kolom, nama_kolom
  from nama_tabel
  where nama_kolom = "nilai_kolom";
```

Contoh

```
create view info_no_plat as
  select id_mobil, no_plat, pemilik, peminjam
  from desc_mobil
  where pemilik = "ibrahim";
```

Hasil

```
MariaDB [rental_mobil]> create view info_no_plat as
-> select id_mobil, no_plat, pemilik, peminjam
-> from desc_mobil
-> where pemilik = "ibrahim";
Query OK, 0 rows affected (0.253 sec)
```

(ASET%20SELECT/membuattabelvirtual.png)

Analisis

Perintah SQL di atas membuat sebuah view dengan nama "info_no_plat". View ini akan menampilkan kolom "id_mobil", "no_plat", "pemilik", dan "peminjam" dari tabel "desc_mobil" hanya untuk baris-baris di mana nilai kolom "pemilik" adalah "ibrahim".

Dengan membuat view seperti ini, pengguna dapat dengan mudah mengakses informasi spesifik tentang mobil yang dimiliki oleh "ibrahim" tanpa perlu menulis kueri yang sama berulang-ulang.

Kesimpulan

Kesimpulan dari perintah ini adalah untuk membuat sebuah view yang menyediakan informasi tentang mobil-mobil yang dimiliki oleh "ibrahim" berdasarkan data yang tersedia dalam tabel "desc_mobil". Ini memungkinkan pengguna untuk dengan cepat dan mudah melihat detail mobil milik "ibrahim" dengan menggunakan view yang telah dibuat.

Menampilkan Tabel Virtual

Struktur

```
select * from nama_tabel;
```

Contoh

```
select * from info_no_plat;
```

Hasil

"menampilkan tabel virtual.png" could not be found.

(ASET%20SELECT/menampilkantabelvirtual.png)

Analisis

Perintah SQL di atas menjalankan kueri untuk menampilkan semua data dari view yang disebut "info_no_plat".

Kesimpulan

Kesimpulan dari perintah ini adalah bahwa kita sedang mencoba untuk melihat semua informasi tentang mobil yang dimiliki oleh "ibrahim" yang telah disaring dan disajikan dalam

bentuk view.

Menghapus Tabel Virtual

Struktur

```
drop view nama_tabel;
```

Contoh

```
drop view info_no_plat;
```

Hasil

"hapus tabel virtual.png" could not be found.

(ASET%20SELECT/hapustabelvirtual.png)

Analisis

Perintah SQL di atas adalah perintah untuk menghapus view yang disebut "info_no_plat" dari basis data.

- DROP VE : ???
- nfo_plat : ???

Kesimpulan

Kesimpulan dari perintah ini adalah bahwa kita sedang mencoba untuk menghapus view yang telah kita buat sebelumnya untuk menyediakan informasi tentang mobil-mobil yang dimiliki oleh "ibrahim". Dengan menjalankan perintah ini, view "info_no_plat" akan dihapus dari basis data, dan tidak akan lagi tersedia untuk digunakan.

Tantangan View

Nomor 1

Struktur

```
CREATE VIEW mobil_Tanpa_peminjam AS  
Select nama_kolom,nama_kolom
```

```
From nama_tabel  
WHERE nama_kolom IS NULL;
```

Contoh

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

Hasil

```
MariaDB [rental_mobil]> CREATE VIEW mobil_Tanpa_peminjam AS  
-> Select no_plat,peminjam  
-> From desc_mobil  
-> WHERE peminjam IS NULL;  
Query OK, 0 rows affected (0.009 sec)  
  
MariaDB [rental_mobil]> select * From mobil_Tanpa_peminjam ;  
+-----+-----+  
| no_plat | peminjam |  
+-----+-----+  
| DD 2901 JK | NULL |  
| DD 2210 LS | NULL |  
+-----+-----+  
2 rows in set (0.005 sec)
```

(ASET%20SELECT/TANTANGAN2.png)

Analisis

`CREATE VIEW mobil_tanpa_peminjam AS` : adalah perintah untuk membuat sebuah view baru atau seperti tabel baru dalam basis data dengan nama mobil_tanpa_peminjam.

- `SELECT no_plat, peminjam` : adalah perintah untuk memilih dua kolom, yaitu no_plat dan peminjam, dari tabel mobil.
- `FROM mobil` : Menunjukkan bahwa data diambil dari tabel bernama mobil.
- `WHERE peminjam IS NULL` : adalah klausa WHERE yang mencari baris-baris dari tabel mobil dimana nilai kolom peminjam adalah NULL.
- `SELECT *`: adalah perintah untuk memilih semua kolom dari view atau tabel.
- `FROM mobil_Tanpa_peminjam`: Menunjukkan bahwa data diambil dari view yang disebut mobil_Tanpa_peminjam, yang telah dibuat sebelumnya.

Kesimpulan

CREATE VIEW mobil_tanpa_peminjam AS Select no_plat, peminjaman FROM mobil WHERE peminjaman IS NULL; digunakan untuk membuat sebuah view baru bernama mobil_Tanpa_peminjam. Viewnya berisi dua kolom, yaitu no_plat dan peminjaman, yang diambil dari tabel mobil hanya baris-baris yang memiliki nilai NULL pada kolom peminjam yang dimasukkan ke dalam view.

SELECT * FROM mobil_tanpa_peminjam; digunakan untuk menampilkan semua data dari view mobil_Tanpa_peminjam, yang telah dibuat sebelumnya dengan kriteria yang bernilai NULL.

Nomor 2

Penjelasan

- `UPDATE` adalah perintah SQL yang digunakan untuk mengubah data dalam tabel.
- `desc_mobil` adalah nama tabel yang akan diubah.
- `SET peminjam=NULL` adalah klausa yang menentukan kolom `peminjam` yang akan diubah nilainya menjadi `NULL`.
- `WHERE id_mobil=2` adalah klausa yang menentukan kondisi di mana perubahan akan diterapkan, dalam hal ini, hanya pada baris dengan nilai `id_mobil` sama dengan 2.

Hasil


```

MariaDB [rental_mobil]> select * from desc_mobil;
+-----+-----+-----+-----+-----+-----+-----+-----+
| id_mobil | no_plat | no_mesin | warna | pemilik | peminjam | deadline | harga_rental |
+-----+-----+-----+-----+-----+-----+-----+-----+
| 1 | DD 2650 XY | ACX3568 | Hitam | Ibrahim | Afdal | 2024-04-24 | 50000 |
| 2 | DD 2440 AX | BCS1120 | Merah | Ibrahim | Elia | 2024-04-24 | 100000 |
| 3 | B 1611 QC | LSQ1112 | Silver | Baim | Anty | 2024-04-24 | 50000 |
| 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_mobil]> Update desc_mobil SET peminjam=NULL WHERE id_mobil=2;
Query OK, 1 row affected (0.003 sec)
Rows matched: 1 Changed: 1 Warnings: 0

MariaDB [rental_mobil]> show tables;
+-----+
| Tables_in_rental_mobil |
+-----+
| desc_mobil |
| frhan |
| mobil_tanpa_peminjam |
| tantangan_raihan |
+-----+
4 rows in set (0.001 sec)

MariaDB [rental_mobil]> select * from mobil_tanpa_peminjam;
+-----+-----+
| no_plat | peminjam |
+-----+-----+
| DD 2440 AX | NULL |
| DD 2901 JK | NULL |
| DD 2210 LS | NULL |
+-----+-----+
3 rows in set (0.004 sec)

```

(ASET%20SELECT/t2.png)

Nomor 3

Berikan Kesimpulan mengapa tabel Virtual 1,2 ini dibuat?

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

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

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

Agregasi

Sum

Struktur

```
select sum(nama_kolom) as total_harga from nama_kolom;
```

Contoh

```
select sum(harga_rental) as total_harga from desc_mobil;
```

Hasil

```
MariaDB [rental_mobil]> select sum(harga_rental) as total_harga from desc_mobil;
+-----+
| total_harga |
+-----+
|      450000 |
+-----+
1 row in set (0.000 sec)
```

(ASET%20SELECT/agregassm1.png)

Analisis

1. **SELECT** : Ini adalah klausa dalam SQL yang digunakan untuk memilih kolom atau ekspresi dari sebuah tabel.
2. **SUM(harga_rental)** : Ini adalah fungsi agregat yang digunakan untuk menghitung total dari nilai-nilai dalam kolom **harga_rental** dari tabel **desc_mobil**. Fungsi SUM() mengambil satu argumen, yaitu kolom yang akan dijumlahkan.
3. **AS total_harga** : Ini adalah alias yang digunakan untuk memberi nama pada hasil dari operasi SUM(harga_rental). Dalam hal ini, hasil sum akan diberi nama **total_harga**.
4. **FROM desc_mobil** : Ini adalah klausa yang menentukan dari mana data diambil. Dalam hal ini, data diambil dari tabel **desc_mobil**.

Kesimpulan

Jadi, secara keseluruhan, program tersebut mengambil nilai-nilai dari kolom **harga_rental** dalam tabel **desc_mobil**, menjumlahkannya, dan memberi nama hasilnya sebagai **total_harga**. Ini berguna untuk mendapatkan total harga rental dari semua mobil yang ada dalam tabel **desc_mobil**.

Count

Struktur

Pemilik

```
select count(nama_kolom) from nama_tabel;
```

Peminjam

```
select count(nama_kolom) from nama_tabel;
```

Contoh

Pemilik

```
select count(pemilik) from desc_mobil;
```

Peminjam

```
select count(peminjam) from desc_mobil;
```

Hasil

Pemilik

```
MariaDB [rental_mobil]> select count(pemilik) from desc_mobil;
+-----+
| count(pemilik) |
+-----+
|                5 |
+-----+
1 row in set (0.029 sec)
```

(ASET%20SELECT/q.png)

Peminjam

```
MariaDB [rental_mobil]> select count(peminjam) from desc_mobil;
+-----+
| count(peminjam) |
+-----+
|                 3 |
+-----+
1 row in set (0.001 sec)
```

(ASET%20SELECT/pm.png)

Analisis

Pemilik

1. **SELECT**: Klausa SQL yang digunakan untuk memilih data dari database.

2. `COUNT(pemilik)`: Ini adalah fungsi agregat yang menghitung jumlah baris di mana nilai kolom `pemilik` tidak null. Fungsi `COUNT()` mengambil satu argumen, yaitu kolom yang ingin dihitung. Dalam hal ini, kolom yang dipilih adalah `pemilik`.
3. `FROM desc_mobil`: Klausa SQL yang menunjukkan tabel mana yang akan digunakan untuk mengambil data. Dalam hal ini, data diambil dari tabel `desc_mobil`.

Peminjam

1. `SELECT`: Klausa SQL yang digunakan untuk memilih data dari database.
2. `COUNT(peminjam)`: Ini adalah fungsi agregat yang menghitung jumlah baris di mana nilai kolom `peminjam` tidak null. Fungsi `COUNT()` mengambil satu argumen, yaitu kolom yang ingin dihitung. Dalam hal ini, kolom yang dipilih adalah `peminjam`.
3. `FROM desc_mobil`: Klausa SQL yang menunjukkan tabel mana yang akan digunakan untuk mengambil data. Dalam hal ini, data diambil dari tabel `desc_mobil`.

Kesimpulan

Jadi, secara keseluruhan, program tersebut akan mengembalikan jumlah baris dalam tabel `desc_mobil` di mana kolom `pemilik` tidak null. Ini berguna untuk menghitung jumlah entitas dalam tabel yang memiliki informasi pemilik mobil.

Jadi, secara keseluruhan, program tersebut akan mengembalikan jumlah baris dalam tabel `desc_mobil` di mana kolom `peminjam` tidak null. Ini berguna untuk menghitung jumlah entitas dalam tabel yang memiliki informasi peminjam mobil.

Min

Struktur

```
SELECT MIN(nama_kolom) AS nilai_minimum FROM nama_tabel;
```

Contoh

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

Hasil

```
MariaDB [rental_mobil]> SELECT MIN(harga_rental) AS MINIMAL FROM desc_mobil;
+-----+
| MINIMAL |
+-----+
| 50000 |
+-----+
1 row in set (0.014 sec)
```

(ASET%20SELECT/MIN.png)

Analisis

1. `SELECT MIN(harga_rental) AS MINIMAL FROM desc_mobil`: Ini adalah klausa SELECT dalam SQL yang digunakan untuk mengambil nilai minimum dari kolom "harga_rental" dari tabel "desc_mobil". Fungsi agregat MIN() digunakan untuk menghitung nilai minimum dari semua nilai dalam kolom "harga_rental".
2. `AS MINIMAL`: Ini adalah klausa yang digunakan untuk memberi alias pada hasil perhitungan nilai minimum. Dalam hal ini, hasil nilai minimum akan diberi nama "MINIMAL".

Kesimpulan

Kesimpulan: Perintah SQL `SELECT MIN(harga_rental) AS MINIMAL FROM desc_mobil;` akan menghasilkan nilai minimal dari kolom "harga_rental" di tabel "desc_mobil". Dengan kata lain, kita akan mendapatkan harga rental terendah yang tercatat dalam tabel tersebut. Hal ini berguna untuk mengetahui harga rental minimum yang tersedia dalam kumpulan data mobil yang ada.

Max

Struktur

```
SELECT MAX(nama_kolom) AS nilai_minimum FROM nama_tabel;
```

Contoh

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

Hasil

```
MariaDB [rental_mobil]> SELECT MAX(harga_rental) AS MAXIMAL FROM desc_mobil;
+-----+
| MAXIMAL |
+-----+
| 150000 |
+-----+
1 row in set (0.002 sec)
```

(ASET%20SELECT/MAX.png)

Analisis

1. `SELECT MAX(harga_rental) AS MAXIMAL FROM desc_mobil`: Ini adalah klausa SELECT dalam SQL yang digunakan untuk mengambil nilai maksimum dari kolom "harga_rental" dari tabel "desc_mobil". Fungsi agregat MAX() digunakan untuk menghitung nilai maksimum dari semua nilai dalam kolom "harga_rental".
2. `AS MAXIMAL`: Ini adalah klausa yang digunakan untuk memberi alias pada hasil perhitungan nilai maksimum. Dalam hal ini, hasil nilai maksimum akan diberi nama "MAXIMAL".

Kesimpulan

Perintah SQL tersebut bertujuan untuk mengambil nilai maksimum dari kolom "harga_rental" dari tabel "desc_mobil" dan memberikan alias "MAXIMAL" pada hasilnya. Ini berarti program tersebut akan mengembalikan nilai maksimum dari harga rental mobil yang ada dalam tabel "desc_mobil". Dengan kata lain, jika ada mobil dengan harga rental tertinggi di tabel tersebut, program ini akan mengembalikan nilai tersebut sebagai "MAXIMAL".

Avg

Struktur

```
SELECT AVG(nama_kolom) AS rata-rata FROM nama_tabel;
```

Contoh

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

Hasil

```
MariaDB [rental_mobil]> SELECT AVG(harga_rental) AS RATA_RATA FROM desc_mobil;
+-----+
| RATA_RATA |
+-----+
| 90000.0000 |
+-----+
1 row in set (0.004 sec)
```

(ASET%20SELECT/AVG.png)

Analisis

1. `SELECT AVG(harga_rental) AS RATA_RATA FROM desc_mobil`: Ini adalah klausa `SELECT` dalam SQL yang digunakan untuk menghitung rata-rata (mean) dari nilai kolom "harga_rental" dari tabel "desc_mobil". Fungsi agregat `AVG()` digunakan untuk menghitung rata-rata dari semua nilai dalam kolom "harga_rental".
2. `AS RATA_RATA`: Ini adalah klausa yang digunakan untuk memberi alias pada hasil perhitungan nilai rata-rata. Dalam hal ini, hasil nilai rata-rata akan diberi nama "RATA_RATA".

Kesimpulan

Perintah SQL tersebut bertujuan untuk menghitung nilai rata-rata (average) dari kolom "harga_rental" dalam tabel "desc_mobil" dan memberikan alias "RATA_RATA" pada hasilnya. Ini berarti program tersebut akan mengembalikan nilai rata-rata dari harga rental mobil yang ada dalam tabel "desc_mobil". Dengan kata lain, program ini akan menghitung total harga rental semua mobil dalam tabel, kemudian membaginya dengan jumlah total mobil, dan mengembalikan nilai rata-rata tersebut sebagai "RATA_RATA".