

AND

Struktur Query

```
SELECT kolom1,kolom2 FROM [nama_tabel] WHERE kolom1="nilai" AND kolom2="nilai2";
```

Contoh Query

```
SELECT warna,pemilik FROM mobil WHERE warna="Biru" AND pemilik="HASIZA";
```

Hasil

```
MariaDB [rental_hasiza]> SELECT warna,pemilik FROM mobil WHERE warna="Biru" AND pemilik="HASIZA";
+-----+-----+
| warna | pemilik |
+-----+-----+
| Biru  | HASIZA  |
+-----+-----+
1 row in set (0.001 sec)
```

Analisis

- SELECT *: Bagian kueri ini menentukan bahwa Anda ingin mengambil semua kolom dari tabel.
- FROM mobil: Ini menunjukkan bahwa Anda menanyakan tabel "mobil".
- WHERE warna="Biru" AND pemilik="HASIZA": Ini adalah kondisi untuk memfilter baris. Ini menetapkan bahwa Anda hanya menginginkan baris yang kolom "warna" sama dengan "Biru" dan kolom "pemilik" sama dengan "HASIZA".

Kesimpulan

"SELECT * FROM mobil WHERE warna='Biru' AND pemilik='HASIZA'" adalah bahwa perintah tersebut akan mengambil semua data dari tabel "mobil" yang memiliki nilai kolom "warna" sama dengan "Biru" dan nilai kolom "pemilik" sama dengan "HASIZA", perintah ini akan mengembalikan semua baris dari tabel "mobil" yang memenuhi kedua kriteria tersebut.

OR

Struktur Query

```
SELECT kolom1,kolom2 FROM [nama_tabel] WHERE kolom1="nilai" AND kolom2="nilai2";
```

Contoh Query

```
SELECT warna,pemilik FROM mobil WHERE warna="Biru" OR pemilik="HASIZA";
```

Hasil

```
MariaDB [rental_hasiza]> SELECT warna,pemilik FROM mobil WHERE warna="BIRU" OR pemilik="HASIZA";
+-----+-----+
| warna | pemilik |
+-----+-----+
| Biru  | HASIZA  |
+-----+-----+
1 row in set (0.001 sec)
```

Analisis

- Kolom "warna" memiliki nilai "Biru".
- Kolom "pemilik" memiliki nilai "HASIZA".

Dalam hal ini, perintah `SELECT *` digunakan untuk mengambil semua kolom (semua atribut) dari tabel "mobil". `FROM mobil` menunjukkan bahwa tabel yang dimaksud adalah "mobil". Kondisi `WHERE` digunakan untuk memfilter baris-baris dalam tabel "mobil".

Operator `OR` menunjukkan bahwa setidaknya salah satu kondisi harus dipenuhi agar baris tersebut diambil. Jadi, baris akan diambil jika warna mobil adalah "Biru" atau jika pemilik mobil adalah "HASIZA".

Kesimpulan

Kesimpulan "`SELECT * FROM mobil WHERE warna='BIRU' OR pemilik='HASIZA';`" adalah bahwa Anda sedang mencari semua data dari tabel "mobil" di mana nilai kolom "warna" sama dengan "BIRU" atau nilai kolom "pemilik" sama dengan "HASIZA".

BETWEEN

Struktur Query

```
SELECT * FROM [nama_tabel] WHERE kolom1 BETWEEN nilai1 AND nilai2;
```

Contoh Query

```
SELECT * FROM mobil WHERE harga_rental BETWEEN 1000000000 AND 300000000000;
```

Hasil

```
MariaDB [rental_hasiza]> SELECT * FROM mobil WHERE harga_rental BETWEEN 1000000000 AND 300000000000;
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| id_mobil | no_plat | no_mesin | warna | pemilik | peminjam | batas_peminjaman | harga_rental | Type |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| 2 | DD1310JH | F1106 | Hitam | AL | CHICA | 1 Tahun | 1000000000 | SUV |
| 4 | DD5853PT | D2J4KK | Putih | DILLA | ALWI | 2 Bulan | 300000000000 | Hotchback |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
2 rows in set (0.001 sec)
```

Analisis

- **SELECT :** Ini merupakan bagian dari perintah SELECT yang digunakan untuk menentukan kolom mana yang ingin Anda ambil dari tabel. Dalam hal ini, tanda "" digunakan untuk mengambil semua kolom yang ada di tabel "mobil".
- **FROM mobil :** Ini menentukan bahwa data akan diambil dari tabel bernama "mobil".
- **WHERE harga_rental BETWEEN 1000000000 AND 300000000000 :** Ini adalah klausal WHERE yang digunakan untuk memfilter baris yang akan diambil berdasarkan kondisi tertentu. Dalam hal ini, kondisi yang digunakan adalah "harga_rental BETWEEN 1000000000 AND 300000000000".
- **Operator BETWEEN** digunakan untuk memeriksa apakah nilai kolom "harga_rental" berada di antara dua angka yang diberikan, yaitu 100.000.000.0 dan 300.000.000.000.

Kesimpulan

Kesimpulan "SELECT * FROM mobil WHERE harga_rental BETWEEN 100.000.000.0 AND 300.000.000.000;" adalah bahwa perintah tersebut digunakan untuk mengambil semua data dari tabel "mobil" yang memenuhi kondisi harga_rental berada di antara 100.000.000.0 dan 300.000.000.000.

NOT BETWEEN

Struktur Query

```
SELECT * FROM [nama_tabel] WHERE kolom1 BETWEEN nilai1 AND nilai2;
```

Contoh Query

```
SELECT * FROM mobil WHERE harga_rental NOT BETWEEN 1000000000 AND 300000000000;
```

Hasil

```
MariaDB [rental_hasiza]> SELECT * FROM mobil WHERE harga_rental NOT BETWEEN 1000000000 AND 300000000000;
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| id_mobil | no_plat | no_mesin | warna | pemilik | peminjam | batas_peminjaman | harga_rental | Type |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| 1 | DD2812HF | J3356 | Biru | HASIZA | FATIH | 3 Bulan | 21340000000000 | Sport |
| 3 | DD4455JK | H4845 | SILVER | JIMIN | PUTRA | 1 Bulan | 23500000000000 | MPV |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
2 rows in set (0.001 sec)

MariaDB [rental_hasiza]>
```

Analisis

- **SELECT** : Bagian ini menentukan bahwa Anda ingin mengambil semua kolom dari tabel yang ditentukan.
- **FROM mobil**: Ini menunjukkan bahwa Anda menanyakan tabel bernama mobil.
- **WHERE harga_rental NOT BETWEEN 1000000000 AND 3000000000000**: Ini adalah kondisi yang memfilter baris. Ini hanya memilih baris yang nilai kolomnya `harga_rental` tidak berada dalam kisaran 100.000.000.0 dan 300.000.000.000.

Kesimpulan

Kesimpulan "SELECT * FROM mobil WHERE harga_rental NOT BETWEEN 1000000000 AND 3000000000000;" adalah bahwa perintah tersebut digunakan untuk mengambil semua data dari tabel "mobil" di mana harga_rental tidak berada di antara 100.000.000.0 dan 300.000.000.000

<=

Struktur

```
SELECT * FROM mobil WHERE harga_rental <= 50000;
```

Contoh

```
SELECT * from mobil WHERE harga_rental <= 3000000000000;
```

Hasil

```
MariaDB [rental_hasiza]> SELECT * from mobil WHERE harga_rental <= 3000000000000;
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| id_mobil | no_plat | no_mesin | warna | pemilik | peminjam | batas_peminjaman | harga_rental | Type |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| 2 | DD1310JH | F1106 | Hitam | AL | CHICA | 1 Tahun | 1000000000 | SUV |
| 4 | DD5853PT | D2J4KK | Putih | DILLA | ALWI | 2 Bulan | 300000000000 | Hotchback |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
2 rows in set (0.001 sec)
```

Analisis

- **SELECT** : Bagian kueri ini digunakan untuk menentukan kolom yang ingin Anda ambil. Tanda bintang `(*)` adalah karakter wildcard yang mewakili semua kolom dalam tabel "mobil".
- **FROM mobil**: Menentukan tabel tempat Anda ingin mengambil data, dalam hal ini, tabel "mobil".
- **WHERE harga_rental <= 3000000000000**: Ini adalah kondisi yang memfilter baris berdasarkan kriteria tertentu. Dalam hal ini, ia hanya memilih baris yang nilai di kolom

"harga_rental" kurang dari atau sama dengan 3000000000000.

Kesimpulan

Kesimpulan "SELECT * FROM mobil WHERE harga_rental <= 3000000000000;" adalah bahwa Anda mencari semua data dari tabel "mobil" di mana nilai kolom "harga_rental" kurang dari atau sama dengan 3000000000000.

>=

Struktur

```
SELECT * FROM mobil WHERE harga_rental >= 50000;
```

Contoh

```
SELECT * from mobil WHERE harga_rental >= 23500000000000;
```

Hasil

```
MariaDB [rental_hasiza]> SELECT * from mobil WHERE harga_rental >= 23500000000000;
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| id_mobil | no_plat | no_mesin | warna | pemilik | peminjam | batas_peminjaman | harga_rental | Type |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| 3 | DD4455JK | H4845 | SILVER | JIMIN | PUTRA | 1 Bulan | 23500000000000 | MPV |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
1 row in set (0.001 sec)
```

Analisis

- SELECT digunakan untuk memilih kolom atau data yang ingin ditampilkan dalam hasil query.
- Tanda bintang (*) setelah kata kunci SELECT menunjukkan bahwa semua kolom dalam tabel "mobil" akan ditampilkan dalam hasil query.
- FROM digunakan untuk menentukan tabel yang akan digunakan dalam query. Dalam kasus ini, tabel yang digunakan adalah "mobil".
- WHERE digunakan untuk melakukan filter atau seleksi pada baris-baris data yang memenuhi kondisi tertentu.
- Kondisi harga_rental >= 23500000000000 menunjukkan bahwa hanya baris-baris data yang memiliki nilai harga_rental yang lebih besar atau sama dengan 23500000000000 yang akan ditampilkan.

Kesimpulan

Perintah SQL SELECT * FROM mobil WHERE harga_rental >= 23500000000000; digunakan untuk mengambil semua data (semua kolom) dari tabel "mobil" di mana nilai

pada kolom "harga_rental" lebih besar atau sama dengan 23500000000000.

<> ATAU !=

Struktur Query

```
SELECT * FROM [nama_tabel] where kolom1 <> nilai1;
```

```
SELECT * FROM [nama_tabel] where kolom1 != nilai1;
```

Contoh Query

```
SELECT * FROM mobil where harga_rental <> 1000000000;
```

```
SELECT * FROM mobil where harga_rental != 300000000000;
```

Hasil

```
MariaDB [rental_hasiza]> SELECT * FROM mobil where harga_rental <> 1000000000;
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| id_mobil | no_plat | no_mesin | warna | pemilik | peminjam | batas_peminjaman | harga_rental | Type |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| 1 | DD2812HF | J3356 | Biru | HASIZA | FATIH | 3 Bulan | 21340000000000 | Sport |
| 3 | DD4455JK | H4845 | SILVER | JIMIN | PUTRA | 1 Bulan | 23500000000000 | MPV |
| 4 | DD5853PT | D2J4KK | Putih | DILLA | ALWI | 2 Bulan | 300000000000 | Hotchback |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
3 rows in set (0.014 sec)

MariaDB [rental_hasiza]>
MariaDB [rental_hasiza]> SELECT * FROM mobil where harga_rental != 300000000000;
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| id_mobil | no_plat | no_mesin | warna | pemilik | peminjam | batas_peminjaman | harga_rental | Type |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| 1 | DD2812HF | J3356 | Biru | HASIZA | FATIH | 3 Bulan | 21340000000000 | Sport |
| 2 | DD1310JH | F1106 | Hitam | AL | CHICA | 1 Tahun | 1000000000 | SUV |
| 3 | DD4455JK | H4845 | SILVER | JIMIN | PUTRA | 1 Bulan | 23500000000000 | MPV |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
3 rows in set (0.001 sec)

MariaDB [rental_hasiza]> |
```

Analisis

- SELECT : Ini menentukan bahwa Anda ingin memilih semua kolom dari tabel.
- FROM mobil : Ini menentukan nama tabel "mobil" dari mana Anda ingin mengambil datanya.
- WHERE harga_rental <> 300000000000 : Ini adalah kondisi yang memfilter baris. Ini hanya memilih baris dimana nilai di kolom "harga_rental" tidak sama dengan 300000000000.

Kesimpulan

Kesimpulan dari "SELECT * FROM mobil WHERE harga_rental <> 300000000000;" adalah bahwa pernyataan tersebut akan mengembalikan semua baris dari tabel "mobil" di mana nilai kolom "harga_rental" tidak sama dengan 300000000000.

TANTANGAN LOGIN

Hasil

```
MariaDB [rental_chica]> SELECT nama
-> FROM akun
-> WHERE username = 'admin'
-> AND password = '12345';
+-----+
| nama  |
+-----+
| Githa |
+-----+
1 row in set (0.001 sec)
```

Analisis

1. **SELECT nama:**

- Bagian ini menunjukkan kolom yang akan diambil dari tabel. Dalam hal ini, kolom "nama" akan diambil dari tabel "Akun".

2. **FROM Akun:**

- Bagian ini menunjukkan tabel dari mana data akan diambil. Dalam hal ini, tabel "Akun" adalah sumber data.

3. **WHERE username = 'Hasiza' AND password = '12345':**

- Bagian ini adalah kondisi yang harus dipenuhi agar data diambil. Di sini, dua kondisi harus terpenuhi:
 - `username` harus sama dengan 'Hasiza'
 - `password` harus sama dengan '12345'
- Kedua kondisi tersebut digabungkan dengan operator logika AND, yang berarti kedua kondisi tersebut harus benar untuk mengambil data.

Kesimpulan

1. **Keamanan:**

- Menyimpan kata sandi dalam bentuk teks biasa (plain text) adalah praktek yang tidak aman. Kata sandi seharusnya disimpan dalam bentuk hash untuk meningkatkan keamanan data.

2. **Fungsi Query:**

- Query ini akan mengembalikan nilai kolom "nama" dari tabel "Akun" untuk baris yang memiliki `username = 'Hasiza'` dan `password = '12345'`.
- Jika tidak ada baris yang memenuhi kedua kondisi tersebut, query tidak akan mengembalikan data apapun.

3. Penggunaan dalam Otentikasi:

- Query ini kemungkinan besar digunakan untuk memverifikasi pengguna berdasarkan kombinasi username dan password. Ini adalah metode yang umum dalam otentikasi pengguna, namun perlu diperhatikan aspek keamanannya.

IN

Struktur Query

```
SELECT * FROM [nama_tabel] WHERE [nama_kolom1] IN("nilai1","nilai2");
```

Contoh Query

```
SELECT * FROM mobil WHERE warna IN("SILVER","Hitam");
```

Hasil

```
MariaDB [rental_hasiza]> SELECT * FROM mobil WHERE warna IN("SILVER","Hitam");
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| id_mobil | no_plat | no_mesin | warna | pemilik | peminjam | batas_peminjaman | harga_rental | Type |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| 2 | DD1310JH | F1106 | Hitam | AL | CHICA | 1 Tahun | 10000000000 | SUV |
| 3 | DD4455JK | H4845 | SILVER | JIMIN | PUTRA | 1 Bulan | 23500000000000 | MPV |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
2 rows in set (0.001 sec)
```

Analisis

1. SELECT * FROM mobil : Pernyataan ini memilih semua kolom dari tabel "mobil".
2. WHERE warna IN ('SILVER', 'Hitam') : WHERE digunakan untuk menerapkan kondisi pada hasil query. kondisi yang diterapkan adalah "warna IN ('SILVER', 'Hitam')", yang berarti hanya baris dengan nilai kolom "warna" yang sama dengan 'SILVER' atau 'Hitam' yang akan diambil.

Kesimpulan

Query ini akan mengambil semua baris dari tabel "mobil" di mana nilai kolom "warna" adalah 'SILVER' atau 'Hitam'. query ini akan mengembalikan semua informasi tentang mobil-mobil yang memiliki warna 'SILVER' atau 'Hitam'. Hasilnya akan berupa kumpulan baris yang mewakili mobil-mobil dengan warna yang sesuai dengan kriteria tersebut.

IN + AND

Struktur Query


```
SELECT * FROM [nama_tabel]
-> WHERE [nama_kolom1] IN ("nilai1","nilai2")
-> AND [nama_kolom2] = nilai;
```

Contoh Query

```
SELECT * FROM mobil
-> WHERE warna IN ("SILVER","Hitam")
-> AND harga_rental = 10000000000;
```

Hasil

```
MariaDB [rental_hasiza]> SELECT * FROM mobil
-> WHERE warna IN ("SILVER","Hitam")
-> AND harga_rental = 10000000000;
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| id_mobil | no_plat | no_mesin | warna | pemilik | peminjam | batas_peminjaman | harga_rental | Type |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| 2 | DD1310JH | F1106 | Hitam | AL | CHICA | 1 Tahun | 10000000000 | SUV |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
1 row in set (0.001 sec)
```

Analisis

1. *SELECT FROM mobil* : memilih semua kolom () dari tabel "mobil".
2. *WHERE warna in('Hitam','SILVER') AND harga_rental = 10000000000* : Pernyataan WHERE digunakan untuk menerapkan kondisi pada hasil query. terdapat dua kondisi yang diterapkan :
 - Kondisi pertama adalah "warna IN ('Hitam', 'SILVER')", yang berarti hanya baris dengan nilai kolom "warna" yang sama dengan 'Hitam' atau 'SILVER' yang akan diambil.
 - Kondisi kedua adalah "harga_rental = 10000000000" , yang berarti hanya baris dengan nilai kolom "harga_rental" yang sama dengan 10000000000 yang akan diambil.

Kesimpulan

Kesimpulan dari SQL tersebut adalah mencari semua entri (baris) dalam tabel "mobil" di mana nilai kolom "warna" adalah 'Hitam' atau 'SILVER', dan nilai kolom "harga_rental" adalah 10000000000. query tersebut akan mengembalikan semua data mobil yang memiliki warna 'Hitam' atau 'SILVER' dan memiliki harga rental sebesar 10000000000.

IN + OR

Struktur Query

```
SELECT * FROM [nama_tabel]
-> WHERE [nama_kolom1] IN ("nilai1","nilai2")
-> OR [nama_kolom2] = nilai;
```

Contoh Query

```
SELECT * FROM mobil
-> WHERE warna IN ("Putih","Biru")
-> OR harga_rental = 10000000000;
```

Hasil

```
gnc syntax to use near -> WHERE warna IN ( 'Putih' , 'Biru' ) at line 2
MariaDB [rental_hasiza]> SELECT * FROM mobil
-> WHERE warna IN ("Putih","Biru")
-> OR harga_rental = 10000000000;
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| id_mobil | no_plat | no_mesin | warna | pemilik | peminjam | batas_peminjaman | harga_rental | Type |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| 1 | DD2812HF | J3356 | Biru | HASIZA | FATIH | 3 Bulan | 21340000000000 | Sport |
| 2 | DD1310JH | F1106 | Hitam | AL | CHICA | 1 Tahun | 10000000000 | SUV |
| 4 | DD5853PT | D2J4KK | Putih | DILLA | ALWI | 2 Bulan | 3000000000000 | Hotchback |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
3 rows in set (0.018 sec)

MariaDB [rental_hasiza]>
```

Analisis

- **SELECT *** : Ini memilih semua kolom dari tabel.
- **FROM mobil** : Ini menentukan nama tabel sebagai "mobil" dari mana data akan diambil.
- **WHERE warna IN ('Putih', 'Biru')** : Kondisi ini memfilter baris berdasarkan nilai pada kolom "warna". Ini memilih baris di mana kolom "warna" memiliki nilai 'Putih' atau 'Biru'.
- **OR harga_rental = 10000000000** : Kondisi ini memfilter lebih lanjut baris-baris tersebut dengan memilih baris-baris yang kolom "harga_rental"-nya bernilai 10000000000.

Kesimpulan

Kesimpulan SQL adalah untuk mengambil semua data dari tabel "mobil" dimana warna mobil adalah 'Putih' atau 'Biru', atau harga rental mobil adalah 10000000000.

menggabungkan kondisi OR antara warna mobil dan harga rental, hasilnya akan mencakup semua mobil yang memiliki warna Putih' atau 'Biru', dan juga mobil dengan harga rental sebesar 10000000000, termasuk mobil yang memenuhi salah satu atau kedua kondisi tersebut.

IN + AND + OPERATOR

Struktur

```
select * from nama_tabel
-> where nama_kolom in('nilai1','nilai2')
-> AND nama_kolom > nilai3
```

```
select * from nama_tabel
-> where nama_kolom in('nilai1','nilai2')
-> AND nama_kolom > nilai3
```

Contoh

```
SELECT * FROM mobil
-> WHERE warna in('Hitam','SILVER')
-> AND harga_rental > 50000;
```

```
SELECT * FROM mobil
-> WHERE warna in('Hitam','SILVER')
-> AND harga_rental > 50000;
```

Hasil

```
MariaDB [rental_hasiza]> SELECT * FROM mobil
-> WHERE warna in('Hitam','SILVER')
-> AND harga_rental > 50000;
```

id_mobil	no_plat	no_mesin	warna	pemilik	peminjam	batas_peminjaman	harga_rental	Type
2	DD1310JH	F1106	Hitam	AL	CHICA	1 Tahun	1000000000	SUV
3	DD4455JK	H4845	SILVER	JIMIN	PUTRA	1 Bulan	23500000000000	MPV

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

Analisis

- Tabel yang diambil tidak disebutkan secara eksplisit dalam pernyataan, hanya disebutkan sebagai `nama_tabel`.
- Kondisi pertama adalah `nama_kolom` harus sama dengan 'nilai1' atau 'nilai2', yang didefinisikan dengan klausa `IN ('nilai1', 'nilai2')`.
- Kondisi kedua adalah `nama_kolom` harus lebih besar dari `nilai3`.
- Perlu diperhatikan bahwa penggunaan `nama_tabel` dan `nama_kolom` tidak spesifik dalam contoh ini. Dalam implementasi yang sebenarnya, Anda akan mengganti `nama_tabel` dengan nama tabel yang sebenarnya dan `nama_kolom` dengan nama kolom yang relevan dalam tabel tersebut.
- • Tabel yang diambil adalah `mobil`.
- Kondisi pertama adalah `warna` harus sama dengan 'Hitam' atau 'SILVER', yang didefinisikan dengan klausa `IN ('Hitam', 'SILVER')`.
- Kondisi kedua adalah `harga_rental` harus lebih besar dari 50000.

- Hasil dari pernyataan ini akan mengembalikan semua kolom dari baris-baris yang memenuhi kedua kondisi tersebut.

Kesimpulan

kedua pernyataan SQL ini digunakan untuk mengambil data dari tabel tertentu dengan beberapa kondisi, yang didefinisikan dengan klausa WHERE.

LIKE

Mencari Awalan

Struktur

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

Contoh

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

Hasil

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

id_mobil	no_plat	no_mesin	warna	pemilik	peminjam	batas_peminjaman	harga_rental	Type
1	DD2812HF	J3356	Biru	ASIZA	FATIH	3 Bulan	21340000000000	Sport
2	DD1310JH	F1106	Hitam	AL	CHICA	1 Tahun	1000000000	SUV
3	DD4455JK	H4845	SILVER	AIMIN	PUTRA	1 Bulan	23500000000000	MPV

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

Analisis

1. SELECT * FROM mobil : Ini akan mengambil semua kolom dari tabel mobil.
2. WHERE pemilik LIKE '%m' : Ini akan memfilter data hanya untuk mobil yang dimiliki oleh pemilik yang nama terakhirnya adalah 'm' atau berisi huruf 'm'. Tanda persen '%' digunakan sebagai wildcard untuk mencari nama pemilik yang mengakhiri dengan 'm' atau berisi huruf 'm' di mana saja.

Kesimpulan

Query ini akan mengambil data dari tabel mobil yang dimiliki oleh pemilik dengan nama terakhir 'm' atau berisi huruf 'm'. Hasil dari query ini akan menampilkan semua kolom dari tabel mobil yang memenuhi

Mencari awalan & akhiran

Struktur

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

Contoh

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

Hasil

```
WHERE pemilik LIKE 'A%' at line 1  
MariaDB [rental_hasiza]> SELECT * FROM mobil  
-> WHERE pemilik LIKE 'A%N';  
+-----+-----+-----+-----+-----+-----+-----+-----+-----+  
| id_mobil | no_plat | no_mesin | warna | pemilik | peminjam | batas_peminjaman | harga_rental | Type |  
+-----+-----+-----+-----+-----+-----+-----+-----+-----+  
| 3 | DD4455JK | H4845 | SILVER | AIMIN | PUTRA | 1 Bulan | 23500000000000 | MPV |  
+-----+-----+-----+-----+-----+-----+-----+-----+-----+  
1 row in set (0.017 sec)
```

Analisis

1. SELECT * FROM mobil : Ini akan mengambil semua kolom dari tabel mobil.
2. WHERE pemilik LIKE 'b%m' : Ini akan memfilter data hanya untuk mobil yang dimiliki oleh pemilik yang nama awalnya dimulai dengan huruf 'b' dan nama terakhirnya berakhir dengan huruf 'm'. Tanda persen '%' digunakan sebagai wildcard untuk mencari nama pemilik yang dimulai dengan huruf 'b' dan mengikuti dengan karakter apapun, lalu diakhiri dengan huruf 'm'.

Kesimpulan

tabel mobil yang dimiliki oleh pemilik dengan nama awal dimulai dengan huruf 'b' dan nama terakhir berakhir dengan huruf 'm'. Hasil dari query ini akan menampilkan semua kolom dari tabel mobil

Mencari total karakter

Struktur

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

```
SELECT * FROM mobil
```

```
-> WHERE pemilik LIKE '___';
```

Contoh

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

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

Hasil

```
MariaDB [rental_hasiza]> SELECT * FROM mobil  
-> WHERE pemilik LIKE 'A____';  
+-----+-----+-----+-----+-----+-----+-----+-----+-----+  
| id_mobil | no_plat | no_mesin | warna | pemilik | peminjam | batas_peminjaman | harga_rental | Type |  
+-----+-----+-----+-----+-----+-----+-----+-----+-----+  
| 1 | DD2812HF | J3356 | Biru | ASIZA | FATIH | 3 Bulan | 21340000000000 | Sport |  
| 3 | DD4455JK | H4845 | SILVER | AIMIN | PUTRA | 1 Bulan | 23500000000000 | MPV |  
+-----+-----+-----+-----+-----+-----+-----+-----+-----+  
2 rows in set (0.002 sec)  
MariaDB [rental_hasiza]>
```

```
MariaDB [rental_hasiza]> SELECT * FROM mobil  
-> WHERE pemilik LIKE '____';  
+-----+-----+-----+-----+-----+-----+-----+-----+-----+  
| id_mobil | no_plat | no_mesin | warna | pemilik | peminjam | batas_peminjaman | harga_rental | Type |  
+-----+-----+-----+-----+-----+-----+-----+-----+-----+  
| 1 | DD2812HF | J3356 | Biru | ASIZA | FATIH | 3 Bulan | 21340000000000 | Sport |  
| 3 | DD4455JK | H4845 | SILVER | AIMIN | PUTRA | 1 Bulan | 23500000000000 | MPV |  
| 4 | DD5853PT | D2J4KK | Putih | DILLA | ALWI | 2 Bulan | 30000000000000 | Hotchback |  
+-----+-----+-----+-----+-----+-----+-----+-----+-----+  
3 rows in set (0.001 sec)
```

Analisis

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

- *pemilik*: Ini adalah nama kolom yang akan difilter. Biasanya ini mewakili nama pemilik kendaraan.
- *MENYUKAI ' _ '*:
- *LIKE*: Operator yang digunakan untuk mencari pola tertentu dalam string.
- *' _ '*: Adalah pola pencarian yang digunakan.
- *___* - Mencocoki *tiga karakter apapun* (wildcard).

Kesimpulan

- (LIKE 'A__'): Mencari pemilik yang ****diawali dengan huruf "A", diikuti oleh dua karakter apapun**. Ini akan mengembalikan pemilik dengan nama minimal 3 karakter.
- (LIKE '_ _'): Mencari pemilik yang memiliki nama dengan ****panjang tepat 3 karakter apapun**.

Kombinasi

Struktur

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

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

Contoh

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

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

Hasil

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

id_mobil	no_plat	no_mesin	warna	pemilik	peminjam	batas_peminjaman	harga_rental	Type
4	DD5853PT	D2J4KK	Putih	DILLA	ALWI	2 Bulan	300000000000	Hotchback

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

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

id_mobil	no_plat	no_mesin	warna	pemilik	peminjam	batas_peminjaman	harga_rental	Type
1	DD2812HF	J3356	Biru	ASIZA	FATIH	3 Bulan	21340000000000	Sport

1 row in set (0.001 sec)

Analisis

- Mencari semua data dari tabel mobil (SELECT *).
- Di mana nilai pada kolom pemilik (WHERE pemilik LIKE '__l%'):
- *LIKE*: Operator untuk mencari pola dalam string.
- *'__l%'*: Pola pencarian yang digunakan.
- *__* : Mencocoki dua karakter apapun (wildcard).
- *r* : Mencari huruf "l" secara literal.
- *%* : Mencocoki nol karakter atau lebih karakter apapun.
- Mencari semua data dari tabel mobil (SELECT *).
- Di mana nilai pada kolom pemilik (WHERE pemilik LIKE '_S%'):
- *LIKE*: Operator untuk mencari pola dalam string.
- **'_S%'*: Pola pencarian yang digunakan.
- *_* : Mencocoki satu karakter apapun.
- *S* : Mencari huruf "S" secara literal.
- *%* : Mencocoki nol karakter atau lebih karakter apapun.

Kesimpulan

- (*LIKE '__l%'*): Mencari pemilik yang namanya ****berakhiran dengan huruf "l"**. Ini akan mengembalikan pemilik dengan nama minimal 3 karakter.
- (*LIKE '_S%'*): Mencari pemilik yang namanya memiliki huruf "S" di ****posisi kedua**. Ini akan mengembalikan pemilik dengan nama minimal 3 karakter.

NOT LIKE

Struktur

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

Contoh

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

Hasil


```
MariaDB [rental_hasiza]> SELECT * FROM mobil WHERE peminjam NOT LIKE 'A%';
```

id_mobil	no_plat	no_mesin	warna	pemilik	peminjam	batas_peminjaman	harga_rental	Type
1	DD2812HF	J3356	Biru	ASIZA	FATIH	3 Bulan	21340000000000	Sport
2	DD1310JH	F1106	Hitam	AL	CHICA	1 Tahun	1000000000	SUV
3	DD4455JK	H4845	SILVER	AIMIN	PUTRA	1 Bulan	23500000000000	MPV

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

Analisis

1. `SELECT * FROM mobil` : Ini akan mengambil semua kolom dari tabel mobil.
2. `WHERE peminjam NOT LIKE 'A%'` : Ini akan memfilter data hanya untuk mobil yang saat ini tidak dipinjam oleh peminjam yang nama awalnya dimulai dengan huruf 'A'. Tanda persen '%' digunakan sebagai wildcard untuk mencari nama peminjam yang dimulai dengan huruf 'A' dan mengikuti dengan karakter apapun. Tanda negasi 'NOT' digunakan untuk mengecualikan peminjam dengan nama awal yang dimulai dengan huruf 'A'.

Kesimpulan

Kesimpulan "`SELECT * FROM mobil WHERE peminjam NOT LIKE 'A%'`" adalah bahwa query tersebut akan mengembalikan semua data dari tabel "mobil" dimana nilai kolom "peminjam" tidak dimulai dengan huruf 'A'.

NULL & NOT NULL

Mencari data kosong

Struktur

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

Contoh

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

Hasil

```
MariaDB [rental_hasiza]> SELECT * FROM mobil WHERE peminjam IS NULL;
Empty set (0.002 sec)
```

Analisis

- `*SELECT` : Menginstruksikan untuk memilih semua kolom (atribut) dari tabel mobil.
- `FROM mobil`: Mendefinisikan tabel tempat data akan diambil, yaitu tabel mobil.

- *WHERE peminjam IS NULL*: Ini adalah klausa WHERE yang memfilter data berdasarkan kondisi tertentu.
- *peminjam*: Merujuk pada kolom pada tabel mobil yang kemungkinan berisi nama peminjam kendaraan.
- *IS NULL*: Operator perbandingan yang mengecek apakah nilai pada kolom peminjam adalah NULL.

Kesimpulan

- Perintah ini hanya mengembalikan mobil dengan kolom peminjam bernilai *NULL* (tidak ada data peminjam).
- Perintah ini membantu dalam mengidentifikasi mobil yang tersedia untuk dipinjam.
- Hasilnya bergantung pada data yang ada di kolom peminjam pada tabel mobil.

Mencari data yang tidak kosong

Struktur

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

Contoh

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

Hasil

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

id_mobil	no_plat	no_mesin	warna	pemilik	peminjam	batas_peminjaman	harga_rental	Type
1	DD2812HF	J3356	Biru	ASIZA	FATIH	3 Bulan	21340000000000	Sport
2	DD1310JH	F1106	Hitam	AL	CHICA	1 Tahun	1000000000	SUV
3	DD4455JK	H4845	SILVER	AIMIN	PUTRA	1 Bulan	23500000000000	MPV
4	DD5853PT	D2J4KK	Putih	DILLA	ALWI	2 Bulan	300000000000	Hotchback
5	DD3712HF	E1326	ABU ABU	SPECKY		20 Bulan	200	Hotchback

5 rows in set (0.001 sec)

Analisis

1. *SELECT* : Ini adalah klausa *SELECT* yang digunakan untuk memilih semua kolom dari tabel "mobil". Dengan menggunakan tanda bintang (*), kita memilih semua kolom yang ada dalam tabel.
2. *FROM mobil*: Ini adalah klausa *FROM* yang menentukan tabel yang akan digunakan dalam kueri. Dalam hal ini, tabel yang digunakan adalah "mobil".
3. *WHERE peminjam IS NOT NULL*: Ini adalah klausa *WHERE* yang digunakan untuk menerapkan kondisi pada kueri. Kondisi yang diterapkan di sini adalah "peminjam IS

NOT NULL", yang berarti hanya baris-baris di mana kolom "peminjam" tidak kosong atau memiliki nilai yang tidak NULL akan dipilih.

Kesimpulan

Digunakan untuk mengambil semua baris dari tabel "mobil" di mana kolom "peminjam" memiliki nilai yang tidak NULL. Hasilnya akan berisi semua kolom dari baris-baris ini.

ORDER BY

Mengurutkan data dari terkecil

Struktur

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

Contoh

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

Hasil

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

id_mobil	no_plat	no_mesin	warna	pemilik	peminjam	batas_peminjaman	harga_rental	Type
3	DD4455JK	H4845	SILVER	AIMIN	PUTRA	1 Bulan	23500000000000	MPV
2	DD1310JH	F1106	Hitam	AL	CHICA	1 Tahun	1000000000	SUV
1	DD2812HF	J3356	Biru	ASIZA	FATIH	3 Bulan	21340000000000	Sport
4	DD5853PT	D2J4KK	Putih	DILLA	ALWI	2 Bulan	300000000000	Hotchback
5	DD3712HF	E1326	ABU ABU	SPECKY		20 Bulan	200	Hotchback

5 rows in set (0.001 sec)

Analisis

- `SELECT * FROM mobil` : Memilih semua kolom dari tabel `mobil` .
- `ORDER BY pemilik ASC` : Mengurutkan hasil berdasarkan kolom `pemilik` secara naik (ascending), artinya hasilnya akan disusun dari nilai `pemilik` yang terkecil ke terbesar.

Kesimpulan

Pernyataan SQL tersebut akan menghasilkan semua baris dari tabel `mobil` , diurutkan berdasarkan nilai kolom `pemilik` dari yang terkecil ke yang terbesar. Ini berarti baris dengan nilai `pemilik` terkecil akan muncul terlebih dahulu dalam hasil, sedangkan baris dengan nilai `pemilik` terbesar akan muncul terakhir.

Mengurutkan data dari terbesar

Struktur

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

Contoh

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

Hasil

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

id_mobil	no_plat	no_mesin	warna	pemilik	peminjam	batas_peminjaman	harga_rental	Type
3	DD4455JK	H4845	SILVER	AIMIN	PUTRA	1 Bulan	23500000000000	MPV
1	DD2812HF	J3356	Biru	ASIZA	FATIH	3 Bulan	21340000000000	Sport
2	DD1310JH	F1106	Hitam	AL	CHICA	1 Tahun	1000000000	SUV
4	DD5853PT	D2J4KK	Putih	DILLA	ALWI	2 Bulan	300000000000	Hotchback
5	DD3712HF	E1326	ABU ABU	SPECKY		20 Bulan	200	Hotchback

5 rows in set (0.004 sec)

Analisis

- `SELECT * FROM mobil` : Memilih semua kolom dari tabel `mobil`.
- `ORDER BY peminjam DESC` : Mengurutkan hasil berdasarkan kolom `peminjam` secara menurun (descending), artinya nilai terbesar `peminjam` akan muncul terlebih dahulu di hasil.

Kesimpulan

ernyataan SQL tersebut akan menghasilkan semua baris dari tabel `mobil`, diurutkan berdasarkan nilai kolom `peminjam` dari yang terbesar ke yang terkecil. Ini berarti baris dengan nilai `peminjam` terbesar akan muncul terlebih dahulu dalam hasil, sedangkan baris dengan nilai `peminjam` terkecil akan muncul terakhir.

Membatasi Data Yang Tampil

Struktur Query

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

Contoh Query

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

Hasil

```
MariaDB [rental_fina]> SELECT * FROM mobil WHERE warna = "Hitam" ORDER BY harga_rental LIMIT 2;
```

id_mobil	no_plat	no_mesin	warna	pemilik	peminjam	deadline	harga_rental
1	DD 2650 XY	ACX3560	Hitam	Ibrahim	Afdal	2024-04-24	50000
5	DD 2210 LS	CJH1011	Hitam	Ibe	NULL	NULL	100000

2 rows in set (0.004 sec)

Analisis

- ***SELECT** :__
 - Menunjukkan bahwa semua kolom dari tabel `mobil` akan dipilih.
- **FROM mobil**:
 - Menyatakan tabel `mobil` sebagai sumber data.
- **WHERE warna = 'Hitam'**:
 - Merupakan kondisi filter yang hanya memilih baris di mana nilai kolom `warna` adalah 'Hitam'.
 - Pastikan penggunaan tanda kutip tunggal (') untuk string, bukan tanda kutip ganda (").
- **ORDER BY harga ASC**:
 - Mengurutkan hasil berdasarkan kolom `harga` dalam urutan naik (dari yang termurah ke yang termahal).
- **LIMIT 2**:
 - Membatasi hasil yang ditampilkan hanya pada dua baris pertama setelah pengurutan dilakukan.

Kesimpulan

query `SELECT * FROM daftar_mobil WHERE warna = "hitam" ORDER BY harga_rental ASC LIMIT 2;` akan menampilkan informasi 2 mobil berwarna hitam dengan harga rental terendah dari tabel "daftar_mobil".

DISTINCT

Struktur

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

Contoh

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

Hasil

```
MariaDB [rental_hasiza]> SELECT DISTINCT (harga_rental) FROM mobil ORDER BY harga_rental DESC;
+-----+
| harga_rental |
+-----+
| 300000000000 |
| 2350000000000 |
| 2134000000000 |
| 200          |
| 1000000000   |
+-----+
5 rows in set (0.001 sec)
```

Analisis

- Perintah ini akan mengambil nilai unik dari kolom `pemilik` dari tabel `mobil`.
- Klausa `DISTINCT` digunakan untuk memastikan bahwa setiap nilai yang dikembalikan hanya muncul satu kali.
- Hasil dari perintah ini akan berisi daftar pemilik mobil yang unik, tanpa duplikasi.
- Perintah ini akan mengambil nilai unik dari kolom `harga_rental` dari tabel `mobil`.
- Klausa `DISTINCT` digunakan untuk memastikan bahwa setiap nilai yang dikembalikan hanya muncul satu kali.
- Klausa `ORDER BY harga_rental DESC` mengurutkan hasil berdasarkan `harga_rental` secara menurun (descending), sehingga nilai terbesar akan muncul terlebih dahulu.
- Hasil dari perintah ini akan berisi daftar harga rental yang unik, tanpa duplikasi, yang diurutkan dari harga rental tertinggi ke terendah.

Kesimpulan

kedua pernyataan SQL ini digunakan untuk mengambil nilai unik dari kolom tertentu dari tabel `mobil`, dengan perintah pertama mengambil pemilik mobil yang unik, dan perintah kedua mengambil harga rental yang unik dan diurutkan dari harga tertinggi ke terendah.

CONCAT, CONCAT_WS, AS

Menggabungkan kolom tanpa pemisah

Struktur

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

Contoh

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

Hasil

```
MariaDB [rental_hasiza]> SELECT CONCAT (pemilik,warna) FROM mobil;
+-----+
| CONCAT (pemilik,warna) |
+-----+
| ASIZABiru              |
| ALHitam                |
| AIMINSILVER            |
| DILLAPutih             |
| SPECKYABU ABU          |
+-----+
5 rows in set (0.001 sec)
```

Analisis

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

Kesimpulan

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

Menggabungkan kolom dengan pemisah

Struktur

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

Contoh

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

Hasil

```
MariaDB [rental_hasiza]> SELECT CONCAT_WS("-",no_plat,no_mesin,id_mobil) FROM mobil;
+-----+
| CONCAT_WS("-",no_plat,no_mesin,id_mobil) |
+-----+
| DD2812HF-J3356-1 |
| DD1310JH-F1106-2 |
| DD4455JK-H4845-3 |
| DD5853PT-D2J4KK-4 |
| DD3712HF-E1326-5 |
+-----+
5 rows in set (0.003 sec)
```

Analisis

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

Kesimpulan

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

Memberikan nama kolom alias

Struktur

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

Contoh

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

Hasil


```

MariaDB [rental_hasiza]> SELECT CONCAT_WS ("+",pemilik,peminjam) AS COLLAB FROM mobil;
+-----+
| COLLAB |
+-----+
| ASIZA+FATIH |
| AL+CHICA |
| AIMIN+PUTRA |
| DILLA+ALWI |
| SPECKY+ |
+-----+
5 rows in set (0.001 sec)

```

Analisis

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

Kesimpulan

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

VIEW

Struktur Query

Membuat Tabel Virtual

```
Create View Info_no_plat AS Select id_mobil, no_plat, pemilik, peminjam from mobil Where pemilik = "Ibrahim";
```

```
Create View Info_no_plat AS
-> Select id_mobil, no_plat, pemilik, peminjam
-> from mobil
-> Where pemilik = "ASIZA";
```

```
ERROR 1146 (42S02): Table 'rental_hasiza.rental_hasiza' doesn't exist
MariaDB [rental_hasiza]> Create View Info_no_plat AS
-> Select id_mobil, no_plat, pemilik, peminjam
-> from mobil
-> Where pemilik = "ASIZA";
Query OK, 0 rows affected (0.021 sec)
```

Menampilkan Tabel Virtual

```
Select From Info_no_plat;
```

```
Select From Info_no_plat;
```

Menghapus Tabel Virtual

```
Drop View info_no_plat:
```

```
Drop View info_no_plat:
```

Analisis

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

Kesimpulan

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

TANTANGAN VIEW

Nomor 1

Penjelasan

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, peminjaman : 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.

Query

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

Hasil

```
MariaDB [rental_hasiza]> CREATE VIEW
-> mobil_tanpa_peminjaman AS
-> SELECT no_plat,peminjam
-> FROM daftar_mobil;
Query OK, 0 rows affected (0.019 sec)
```

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 mobil: adalah perintah untuk memperbarui data dalam tabel yang disebut mobil.

- SET peminjaman = NULL : menetapkan nilai kolom peminjam menjadi NULL.
- WHERE peminjam = 'afdal' : adalah klausa WHERE yang membatasi update hanya pada baris-baris dimana nilai kolom peminjam adalah 'afdal'. Maksudnya perubahan hanya akan berlaku untuk baris-baris yang memiliki peminjam dengan nama 'afdal'.
- 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.

Query

```
UPDATE mobil
-> SET peminjaman = NULL
-> WHERE peminjaman = 'FATIH';
```

Hasil

```
MariaDB [rental_hasiza]> UPDATE daftar_mobil
-> SET peminjam = NULL
-> WHERE peminjam = 'FATIH';
Query OK, 1 row affected, 1 warning (0.017 sec)
Rows matched: 1  Changed: 1  Warnings: 1

MariaDB [rental_hasiza]> SELECT * FROM daftar_mobil;
```

	id_mobil	no_plat	no_mesin	warna	pemilik	peminjam	batas_peminjam	harga_rental	Type
1	DD2812HF	J3356	Biru	HASIZA		NULL		213400000000000	Sport
2	DD1310JH	F1106	Hitam	AL	CHICA	NULL		1000000000	SUV
3	DD4455JK	H4845	Silver	AIMIN	PUTRA	NULL		235000000000000	MPV
4	DD5853PT	D2J4KK	Putih	DILLA	ALWI	NULL		300000000000000	Hotchback
5	DD3712HF	E1326	Abu Abu	SPECKY		NULL		250000000000000	Hotchback
6	DD2224LC	H4856CH	Merah	AI	ECA	NULL		230000000000000	

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

Kesimpulan

UPDATE mobil SET peminjaman = NULL WHERE peminjaman = 'afdal'; nilai pada kolom peminjaman pada tabel `mobil` yang memiliki nilai 'afdal' akan diubah menjadi NULL.

Kesimpulannya, perintah digunakan untuk menghapus atau mengubah nilai peminjaman menjadi NULL untuk semua data di tabel mobil yang berada di kolom `peminjaman` memiliki nilai 'afdal'.

SELECT * FROM mobil_tanpa_peminjam; digunakan untuk menampilkan semua data dari view mobil_tanpa_peminjam, yang telah dibuat sebelumnya dengan mengubah atau menghapus nilai peminjam menjadi NULL untuk tabel mobil dimana peminjam memiliki nilai afdal.

Nomor 3

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

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

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

AGREGASI

Sum

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

Struktur

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

Contoh

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

Hasil

```
MariaDB [rental_hasiza]> SELECT SUM(harga_rental) FROM mobil;  
+-----+  
| SUM(harga_rental) |  
+-----+  
|      45141000000200 |  
+-----+  
1 row in set (0.013 sec)
```

Analisis

- `SELECT` : Digunakan untuk mengambil data dari database.
- `SUM()` : Fungsi agregat dalam SQL yang digunakan untuk menghitung jumlah nilai dalam kolom.
- `harga_rental` : Ini adalah nama kolom yang ingin Anda hitung jumlahnya.
- `FROM mobil` : Menentukan tabel dari mana Anda ingin mengambil datanya, dalam hal ini, tabel bernama `mobil`.

Kesimpulan

Kesimpulan dari pernyataan SQL `SELECT SUM(harga_rental) FROM mobil;` adalah bahwa Anda ingin menghitung jumlah total harga rental dari semua mobil yang ada dalam tabel "mobil".

Count

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

Struktur

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

Contoh

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

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

Hasil

```
MariaDB [rental_hasiza]> SELECT COUNT(pemilik) FROM mobil;
+-----+
| COUNT(pemilik) |
+-----+
|                5 |
+-----+
1 row in set (0.018 sec)
```

Analisis

pemilik

- `SELECT` : Kata kunci ini digunakan untuk mengambil data dari database.
- `COUNT()` : Ini adalah fungsi agregat yang menghitung jumlah baris yang dikembalikan oleh kueri. Dalam hal ini, ia akan menghitung jumlah nilai bukan nol pada kolom "pemilik".
- `pemilik` : Ini nama kolomnya. Fungsi ini `COUNT()` akan menghitung jumlah nilai bukan nol di kolom ini.
- `FROM mobil` : Ini menentukan tabel dari mana data akan diambil. Dalam hal ini, itu adalah tabel "mobil".

peminjam

- `SELECT` : Kata kunci ini digunakan untuk mengambil data dari database.
- `COUNT(peminjam)` : Fungsi ini menghitung jumlah nilai bukan nol pada kolom yang ditentukan, dalam hal ini, "peminjam".
- `FROM mobil` : Ini menentukan tabel untuk mengambil data, dalam hal ini, tabel bernama "mobil".

Kesimpulan

pemilik

Kesimpulan dari `SELECT COUNT(pemilik) FROM mobil;` adalah bahwa Anda ingin menghitung jumlah entri unik dalam kolom "pemilik" dari tabel "mobil".

peminjam

Kesimpulan dari `SELECT COUNT(peminjam) FROM mobil;` adalah bahwa Anda ingin menghitung jumlah entri unik dalam kolom "peminjam" dari tabel "mobil". Hasilnya akan memberikan jumlah peminjam mobil yang terdaftar dalam tabel.

Min

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

Struktur

```
SELECT MIN(nama_colom) AS nilai_minimum

FROM nama_tabel

WHERE kondisi_opsional;
```

Contoh

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

Hasil

```
MariaDB [rental_hasiza]> SELECT MIN(harga_rental) AS MINIMAL FROM mobil;
+-----+
| MINIMAL |
+-----+
| 1000000000 |
+-----+
1 row in set (0.001 sec)
```

Analisis

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

Kesimpulan

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

Max

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

Struktur

```
SELECT MAX(nama_colom) AS nilai_maksimum

FROM nama_tabel

WHERE kondisi_opsional;
```

Contoh

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

Hasil

```
MariaDB [rental_hasiza]> SELECT MAX(harga_rental) AS MAXIMAL FROM mobil;
+-----+
| MAXIMAL |
+-----+
| 300000000000 |
+-----+
1 row in set (0.001 sec)
```

Analisis

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

Kesimpulan

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

AVG

Fungsi agregasi yang digunakan untuk menghitung nilai rata-rata

Struktur

```
SELECT AVG(nama_colom) AS rata_rata

FROM nama_tabel

WHERE kondisi_opsional;
```

Contoh

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

Hasil

```
MariaDB [rental_hasiza]> SELECT AVG(harga_rental) AS RATA_RATA FROM mobil;
+-----+
| RATA_RATA |
+-----+
| 9028200000040 |
+-----+
1 row in set (0.015 sec)
```

Analisis

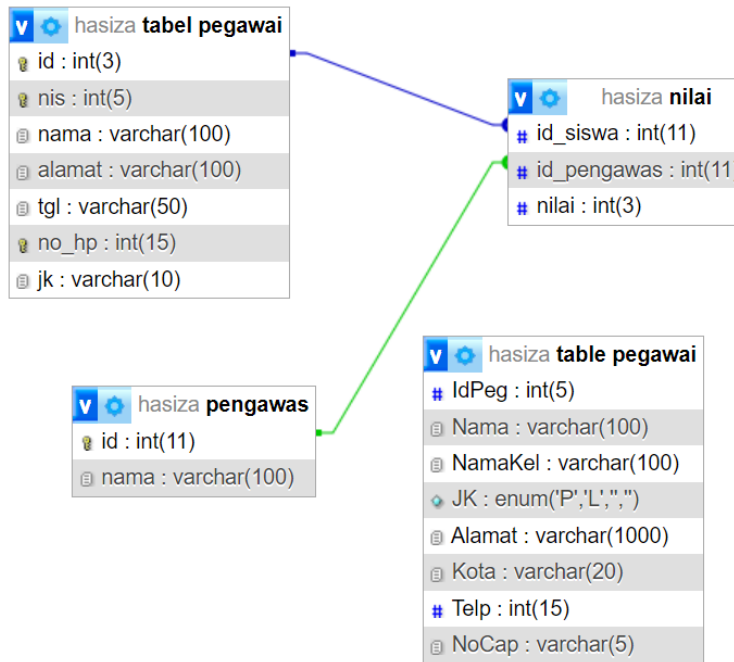
- `SELECT AVG(harga_rental)` : Ini adalah perintah SQL yang digunakan untuk mengambil nilai rata-rata dari kolom `harga_rental`. `AVG()` adalah fungsi agregat yang digunakan untuk menghitung rata-rata dari sekumpulan nilai.
- `AS RATA_RATA` : Menggunakan kata kunci `AS` untuk memberi nama pada hasil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

Kesimpulan

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

Tantangan PHP MyAdmin

Hasil Relasi



Konsol

Struktur Pegawai

Server: 127.0.0.1 » Database: hasiza » Tabel: table pegawai									
Jelajahi Struktur SQL Cari Tambahkan Ekspor Impor Hak Akses Operasi Trigger									
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(1000)	utf8mb4_general_ci		Tidak	Tidak ada			Ubah Hapus Lainnya
<input type="checkbox"/>	6 Kota	varchar(20)	utf8mb4_general_ci		Tidak	Tidak ada			Ubah Hapus Lainnya
<input type="checkbox"/>	7 Telp	int(15)			Tidak	Tidak ada			Ubah Hapus Lainnya
<input type="checkbox"/>	8 NoCap	varchar(5)	utf8mb4_general_ci		Tidak	Tidak ada			Ubah Hapus Lainnya

Data Pegawai

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

Extra options

				idpeg	nama	namakel	jk	alamat	kota	telp	nocab
<input type="checkbox"/>	Ubah	Salin	Hapus	10107	Emya	Salsalina	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

Before

[Edit dikotak] [Ubah] [Buat kode PHP]

Struktur tabel | Tampilan hubungan

#	Nama	Jenis	Penyortiran	Atribut	Tak Ternilai	Bawaan	Komentar	Ekstra	Tindakan
<input type="checkbox"/>	1	IdPeg	int(5)		Tidak	Tidak ada			Lainnya
<input type="checkbox"/>	2	Nama	varchar(100)	utf8mb4_general_ci	Tidak	Tidak ada			Lainnya
<input type="checkbox"/>	3	NamaKel	varchar(100)	utf8mb4_general_ci	Tidak	Tidak ada			Lainnya
<input type="checkbox"/>	4	JK	varchar(11)	utf8mb4_general_ci	Tidak	Tidak ada			Lainnya
<input type="checkbox"/>	5	Alamat	varchar(1000)	utf8mb4_general_ci	Tidak	Tidak ada			Lainnya
<input type="checkbox"/>	6	Kota	varchar(20)	utf8mb4_general_ci	Tidak	Tidak ada			Lainnya
<input type="checkbox"/>	7	Telp	int(15)		Tidak	Tidak ada			Lainnya
<input type="checkbox"/>	8	NoCap	varchar(5)	utf8mb4_general_ci	Tidak	Tidak ada			Lainnya

After

Server: 127.0.0.1 » Database: hasiza » Tabel: table pegawai

Jelajahi | Struktur | SQL | Cari | Tambahkan | Ekspor | Impor | Hak Akses | Operasi | Trigger

Struktur tabel | Tampilan hubungan

#	Nama	Jenis	Penyortiran	Atribut	Tak Ternilai	Bawaan	Komentar	Ekstra	Tindakan
<input type="checkbox"/>	1	IdPeg	int(5)		Tidak	Tidak ada			Lainnya
<input type="checkbox"/>	2	Nama	varchar(100)	utf8mb4_general_ci	Tidak	Tidak ada			Lainnya
<input type="checkbox"/>	3	NamaKel	varchar(100)	utf8mb4_general_ci	Tidak	Tidak ada			Lainnya
<input type="checkbox"/>	4	JK	enum('P', 'L', '', '')	utf8mb4_general_ci	Tidak	Tidak ada			Lainnya
<input type="checkbox"/>	5	Alamat	varchar(1000)	utf8mb4_general_ci	Tidak	Tidak ada			Lainnya
<input type="checkbox"/>	6	Kota	varchar(20)	utf8mb4_general_ci	Tidak	Tidak ada			Lainnya
<input type="checkbox"/>	7	Telp	int(15)		Tidak	Tidak ada			Lainnya
<input type="checkbox"/>	8	NoCap	varchar(5)	utf8mb4_general_ci	Tidak	Tidak ada			Lainnya

Cabang

Before

Struktur tabel | Tampilan hubungan

#	Nama	Jenis	Penyortiran	Atribut	Tak Ternilai	Bawaan	Komentar	Ekstra	Tindakan
<input type="checkbox"/>	1	NoCab	varchar(5)	utf8mb4_general_ci	Tidak	Tidak ada			Lainnya
<input type="checkbox"/>	2	Nama	varchar(30)	utf8mb4_general_ci	Tidak	Tidak ada			Lainnya
<input type="checkbox"/>	3	Alamat	varchar(50)	utf8mb4_general_ci	Tidak	Tidak ada			Lainnya
<input type="checkbox"/>	4	Kabupaten	varchar(20)	utf8mb4_general_ci	Tidak	Tidak ada			Lainnya
<input type="checkbox"/>	5	No Telp	int(15)		Tidak	Tidak ada			Lainnya
<input type="checkbox"/>	6	Kab Kota	varchar(20)	utf8mb4_general_ci	Tidak	Tidak ada			Lainnya
<input type="checkbox"/>	7	Email	varchar(50)	utf8mb4_general_ci	Tidak	Tidak ada			Lainnya

After

✓ Tabel cabang berhasil diubah.

```
ALTER TABLE `cabang` CHANGE `Email` `Email` TEXT CHARACTER SET utf8mb4 COLLATE utf8mb4_general_ci NOT NULL;
```

[[Edit dikotak](#)] [[Ubah](#)] [[Buat kode PHP](#)]



Struktur tabel



Tampilan hubungan

	#	Nama	Jenis	Penyortiran	Atribut	Tak Ternilai	Bawaan	Komentar	Ekstra	Tindakan
<input type="checkbox"/>	1	NoCab	varchar(5)	utf8mb4_general_ci		Tidak	Tidak ada			Ubah Hapus Lainnya
<input type="checkbox"/>	2	Nama	varchar(30)	utf8mb4_general_ci		Tidak	Tidak ada			Ubah Hapus Lainnya
<input type="checkbox"/>	3	Alamat	varchar(50)	utf8mb4_general_ci		Tidak	Tidak ada			Ubah Hapus Lainnya
<input type="checkbox"/>	4	Kabupaten	varchar(20)	utf8mb4_general_ci		Tidak	Tidak ada			Ubah Hapus Lainnya
<input type="checkbox"/>	5	No Telp	int(15)			Tidak	Tidak ada			Ubah Hapus Lainnya
<input type="checkbox"/>	6	Kab Kota	varchar(20)	utf8mb4_general_ci		Tidak	Tidak ada			Ubah Hapus Lainnya
<input type="checkbox"/>	7	Email	text	utf8mb4_general_ci		Tidak	Tidak ada			Ubah Hapus Lainnya