

# GROUP BY

## STRUKTUR

```
SELECT nama_kolom COUNT(nama_kolom);
```

## CONTOH

```
SELECT jabatan, COUNT(NIP) AS JumlahPegawai FROM pegawai2 GROUP BY jabatan;
```

## HASIL

```
MariaDB [basis_data]> SELECT * FROM pegawai2;
```

NIP	NDep	NBIk	JK	Alamat	Telp	jabatan	gaji	NoCab
10107	Enya	Salsalina	P	Jl. Suci 78 Bandung	022-555768	Manajer	5250000	C101
10246	Dian	Anggraini	P	Jl. Mawar 5 Semarang	024-555102	Sales	2750000	C103
10324	Martin	Susanto	L	Jl. Bima 51 Jakarta	021-555785	Staf	1750000	C102
10252	Antoni	Irawan	L	Jl. A. Yani 15 Jakarta	021-555888	Manajer	5750000	C102
10176	Diah	Wahyuni	P	Jl. Maluku 56 Bandung	022-555934	Sales	2500000	C101
10314	Ayu	Rahmadani	P	Jl. Malaka 342 Jakarta	021-555098	Sales	1950000	C102
10307	Erik	Andrian	L	Jl. Manggis 5 Semarang	024-555236	Manajer	6250000	C103
10415	Susan	Sumantri	P	Jl. Pahlawan 24 Surabaya	031-555120	NULL	2650000	C104
10407	Rio	Gunawan	L	Jl. Melati 356 Surabaya	031-555231	Staf	1725000	C104

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

```
MariaDB [basis_data]> SELECT jabatan, COUNT(NIP) AS JumlahPegawai FROM pegawai2 GROUP BY jabatan;
```

jabatan	JumlahPegawai
NULL	1
Manajer	3
Sales	3
Staf	2

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

## ANALISIS

- `SELECT jabatan, COUNT(NIP) AS JumlahPegawai` : Ini adalah pernyataan `SELECT` yang memilih dua kolom dari tabel `pegawai2`. Kolom pertama adalah `jabatan`, yang merupakan kolom yang akan dikelompokkan. Kolom kedua adalah `COUNT(NIP) AS JumlahPegawai`, yang menghitung jumlah baris (pegawai) dalam setiap kelompok jabatan dan memberi nama alias `JumlahPegawai` pada hasil perhitungan tersebut.
- `FROM pegawai2` : Ini menunjukkan bahwa data diambil dari tabel `pegawai2`. Tabel ini memiliki kolom `jabatan` (untuk dikelompokkan) dan `NIP` (Nomor Induk Pegawai) yang digunakan untuk perhitungan jumlah pegawai per jabatan.
- `GROUP BY jabatan` : Ini adalah klausa `GROUP BY` yang digunakan untuk mengelompokkan baris-baris dalam tabel berdasarkan nilai unik pada kolom `jabatan`.

Dalam konteks ini, SQL akan menghitung jumlah pegawai untuk setiap nilai yang berbeda dari kolom `jabatan`.

## KESIMPULAN

Kode SQL mengambil data dari tabel `pegawai2`, mengelompokkan mereka berdasarkan jabatan, dan menghitung jumlah pegawai dalam setiap kelompok jabatan tersebut. Hasilnya adalah daftar jabatan beserta jumlah pegawai yang memiliki jabatan tersebut dalam tabel `pegawai2`.

## HAVING

## STRUKTUR

```
SELECT nama_kolom COUNT(nama_kolom) AS nama_ganti FROM nama_tabel GROUP BY
nama_kolom HAVING COUNT(nama_kolom) kondisi;
```

## CONTOH

```
SELECT jabatan, COUNT(NIP) AS JumlahPegawai FROM pegawai2 GROUP BY jabatan
HAVING COUNT(NIP) >= 3;
```

## HASIL

```
MariaDB [basis_data]> SELECT * FROM pegawai2;
```

NIP	NDep	NBIk	JK	Alamat	Telp	jabatan	gaji	NoCab
10107	Enya	Salsalina	P	Jl. Suci 78 Bandung	022-555768	Manajer	5250000	C101
10246	Dian	Anggraini	P	Jl. Mawar 5 Semarang	024-555102	Sales	2750000	C103
10324	Martin	Susanto	L	Jl. Bima 51 Jakarta	021-555785	Staf	1750000	C102
10252	Antoni	Irawan	L	Jl. A. Yani 15 Jakarta	021-555888	Manajer	5750000	C102
10176	Diah	Wahyuni	P	Jl. Maluku 56 Bandung	022-555934	Sales	2500000	C101
10314	Ayu	Rahmadani	P	Jl. Malaka 342 Jakarta	021-555098	Sales	1950000	C102
10307	Erik	Andrian	L	Jl. Manggis 5 Semarang	024-555236	Manajer	6250000	C103
10415	Susan	Sumantri	P	Jl. Pahlawan 24 Surabaya	031-555120	NULL	2650000	C104
10407	Rio	Gunawan	L	Jl. Melati 356 Surabaya	031-555231	Staf	1725000	C104

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

```
MariaDB [basis_data]> SELECT jabatan, COUNT(NIP) AS JumlahPegawai FROM pegawai2 GROUP BY jabatan HAVING COUNT(NIP) >= 3;
```

jabatan	JumlahPegawai
Manajer	3
Sales	3

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

# ANALISIS

- `SELECT jabatan, COUNT(NIP) AS JumlahPegawai`: Ini adalah pernyataan `SELECT` yang memilih dua kolom dari tabel `pegawai2`. Kolom pertama adalah `jabatan`, yang akan dikelompokkan. Kolom kedua adalah `COUNT(NIP) AS JumlahPegawai`, yang menghitung jumlah baris (pegawai) dalam setiap kelompok jabatan dan memberi nama alias `JumlahPegawai` pada hasil perhitungan tersebut.
- `FROM pegawai2`: Ini menunjukkan bahwa data diambil dari tabel `pegawai2`. Tabel ini memiliki kolom `jabatan` (untuk dikelompokkan) dan `NIP` (Nomor Induk Pegawai) yang digunakan untuk perhitungan jumlah pegawai per jabatan.
- `GROUP BY jabatan`: Ini adalah klausa `GROUP BY` yang digunakan untuk mengelompokkan baris-baris dalam tabel berdasarkan nilai unik pada kolom `jabatan`. Dalam konteks ini, SQL akan menghitung jumlah pegawai untuk setiap nilai yang berbeda dari kolom `jabatan`.
- `HAVING COUNT(NIP) >= 3`: Ini adalah klausa `HAVING` yang digunakan untuk memberikan kriteria seleksi pada hasil dari operasi `GROUP BY`. Dalam hal ini, hanya baris-baris yang memiliki nilai `COUNT(NIP)` (jumlah pegawai per jabatan) setidaknya sama dengan 3 yang akan disertakan dalam hasil akhir query.

## KESIMPULAN

Kode SQL di atas mengambil data dari tabel `pegawai2`, mengelompokkannya berdasarkan jabatan, menghitung jumlah pegawai untuk setiap jabatan, dan kemudian memfilter hasilnya dengan hanya menyertakan jabatan-jabatan yang memiliki setidaknya tiga pegawai. Hasilnya adalah daftar jabatan beserta jumlah pegawai untuk jabatan-jabatan yang memenuhi kriteria tersebut.

## SOAL

## TAMPILKAN JUMLAH DATA MOBIL DAN KELOMPOKKAN BERDASARKAN WARNANYA

### STRUKTUR

```
SELECT nama_kolom COUNT(nama_kolom) AS nama_ganti FROM nama_tabel GROUP BY
nama_kolom;
```

## CONTOH

```
SELECT warna, COUNT(id_mobil) AS Jumlah_Warna FROM data_mobil GROUP BY
```

```
warna;
```

## HASIL

```
MariaDB [rental_fare1]> SELECT * FROM data_mobil;
+-----+-----+-----+-----+-----+-----+-----+
| id_mobil | no_plat | no_mesin | warna | pemilik | peminjam | harga_rental |
+-----+-----+-----+-----+-----+-----+-----+
| 1 | DD 2650 XY | ACX3560 | HITAM | FAREL | TAUFIQ | 50000 |
| 2 | DD 2440 AX | BCS1120 | MERAH | FATUR | AHSAN | 100000 |
| 3 | B 1611 QC | LSQ1112 | SILVER | ASEP | CAPO | 50000 |
| 4 | DD 2901 JK | UQL1029 | HITAM | JORDAN | NULL | 150000 |
| 5 | DD 2210 LS | CJH1011 | HITAM | NAFAN | NULL | 100000 |
| 6 | DD 5346 XZ | NSR0077 | MERAH | FACHRI | NULL | 150000 |
+-----+-----+-----+-----+-----+-----+-----+
6 rows in set (0.001 sec)

MariaDB [rental_fare1]> SELECT warna, COUNT(id_mobil) AS Jumlah_Warna FROM data_mobil GROUP BY warna;
+-----+-----+
| warna | Jumlah_Warna |
+-----+-----+
| HITAM | 3 |
| MERAH | 2 |
| SILVER | 1 |
+-----+-----+
3 rows in set (0.002 sec)
```

## ANALISIS

1. **SELECT warna, COUNT(id\_mobil) AS Jumlah\_Warna:**
  - **warna:** Ini adalah kolom dari tabel `data_mobil` yang akan ditampilkan dalam hasil query.
  - **COUNT(id\_mobil) AS Jumlah\_Warna:** Fungsi `COUNT(id_mobil)` digunakan untuk menghitung jumlah baris yang memiliki nilai `id_mobil` tidak null dalam setiap kelompok warna. Hasilnya akan diberi alias `Jumlah_Warna` sehingga lebih mudah dibaca.
2. **FROM data\_mobil:**
  - Ini menentukan tabel sumber data, yaitu `data_mobil`.
3. **GROUP BY warna:**
  - Klausul `GROUP BY` digunakan untuk mengelompokkan data berdasarkan kolom `warna`. Semua baris dengan nilai `warna` yang sama akan dikelompokkan bersama.

## KESIMPULAN

Kode SQL tersebut digunakan untuk menghitung jumlah mobil berdasarkan warna dalam tabel `data_mobil`. Dengan menggunakan klausa `GROUP BY`, kode ini mengelompokkan data berdasarkan kolom `warna` dan menghitung jumlah mobil untuk setiap kelompok warna menggunakan fungsi `COUNT`. Hasil dari query ini akan memberikan distribusi jumlah mobil untuk setiap warna yang ada di dalam tabel.

Berdasarkan query ini tampilkan yg lebih besar dari 3 atau sama dengan 3 pemilik mobilnya

## STRUKTUR

```
SELECT nama_kolom COUNT(nama_kolom) AS nama_ganti FROM nama_tabel GROUP BY
nama_kolom HAVING COUNT(nama_kolom) kondisi;
```

## CONTOH

```
SELECT peminjam, COUNT(id_mobil) AS JumlahMobil FROM data_mobil GROUP BY
peminjam HAVING COUNT(id_mobil) >= 3;
```

## HASIL

```
MariaDB [rental_farel]> SELECT * FROM data_mobil;
+-----+-----+-----+-----+-----+-----+-----+
| id_mobil | no_plat | no_mesin | warna | pemilik | peminjam | harga_rental |
+-----+-----+-----+-----+-----+-----+-----+
| 1 | DD 2650 XY | ACX3560 | HITAM | FAREL | TAUFIQ | 50000 |
| 2 | DD 2440 AX | BCS1120 | MERAH | FATUR | AHSAN | 100000 |
| 3 | B 1611 QC | LSQ1112 | SILVER | ASEP | CAPO | 50000 |
| 4 | DD 2901 JK | UQL1029 | HITAM | JORDAN | NULL | 150000 |
| 5 | DD 2210 LS | CJH1011 | HITAM | NAFAN | NULL | 100000 |
| 6 | DD 5346 XZ | NSR0077 | MERAH | FACHRI | NULL | 150000 |
+-----+-----+-----+-----+-----+-----+-----+
6 rows in set (0.001 sec)

MariaDB [rental_farel]> SELECT peminjam, COUNT(id_mobil) AS JumlahMobil FROM data_mobil GROUP BY peminjam H
AVING COUNT(id_mobil) >= 3;
+-----+-----+
| peminjam | JumlahMobil |
+-----+-----+
| NULL | 3 |
+-----+-----+
1 row in set (0.002 sec)
```

## ANALISIS

1. **SELECT peminjam, COUNT(id\_mobil) AS JumlahMobil:**

- **peminjam:** Kolom dari tabel `data_mobil` yang mewakili identitas atau nama peminjam.
- **COUNT(id\_mobil) AS JumlahMobil:** Fungsi `COUNT(id_mobil)` digunakan untuk menghitung jumlah mobil yang dipinjam oleh setiap peminjam. Hasilnya diberi alias `JumlahMobil` untuk memudahkan pembacaan hasil.

2. **FROM data\_mobil:**

- Ini menentukan tabel sumber data, yaitu `data_mobil`.

3. **GROUP BY peminjam:**

- Klausa `GROUP BY` digunakan untuk mengelompokkan data berdasarkan kolom `peminjam`. Semua baris dengan nilai `peminjam` yang sama akan dikelompokkan bersama.

#### 4. `HAVING COUNT (id_mobil) >= 3`:

- Klausa `HAVING` digunakan untuk memfilter kelompok hasil yang sudah dikelompokkan. Dalam hal ini, hanya kelompok dengan jumlah mobil (`COUNT(id_mobil)`) yang sama dengan atau lebih dari 3 yang akan disertakan dalam hasil akhir.

## KESIMPULAN

Kode SQL ini membantu untuk mengidentifikasi peminjam yang telah meminjam tiga mobil atau lebih. Dengan mengelompokkan data berdasarkan peminjam dan menerapkan filter dengan klausa `HAVING`, kita bisa fokus pada peminjam yang sering meminjam mobil, yang bisa sangat berguna untuk analisis dan pengambilan keputusan terkait peminjaman mobil.

## Tampilkan semua pemilik beserta jumlah mobilnya yang memiliki lebih atau sama dengan 3 mobil

## STRUKTUR

```
SELECT nama_kolom, COUNT(nama_kolom) AS nama_ganti FROM nama_tabel GROUP BY
nama_kolom;
```

## CONTOH

```
SELECT peminjam, COUNT(id_mobil) AS JumlahMobil FROM data_mobil GROUP BY
peminjam;
```

## HASIL



```
MariaDB [rental_farel]> SELECT * FROM data_mobil;
```

id_mobil	no_plat	no_mesin	warna	pemilik	peminjam	harga_rental
1	DD 2650 XY	ACX3560	HITAM	FAREL	TAUFIQ	50000
2	DD 2440 AX	BCS1120	MERAH	FATUR	AHSAN	100000
3	B 1611 QC	LSQ1112	SILVER	ASEP	CAPO	50000
4	DD 2901 JK	UQL1029	HITAM	JORDAN	NULL	150000
5	DD 2210 LS	CJH1011	HITAM	NAFAN	NULL	100000
6	DD 5346 XZ	NSR0077	MERAH	FACHRI	NULL	150000

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

  

```
MariaDB [rental_farel]> SELECT peminjam, COUNT(id_mobil) AS JumlahMobil FROM data_mobil GROUP BY peminjam;
```

peminjam	JumlahMobil
NULL	3
AHSAN	1
CAPO	1
TAUFIQ	1

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

## ANALISIS

### Analisis Kode:

#### 1. SELECT peminjam, COUNT(id\_mobil) AS JumlahMobil:

- **peminjam**: Kolom dari tabel `data_mobil` yang mewakili identitas atau nama peminjam.
- **COUNT(id\_mobil) AS JumlahMobil**: Fungsi `COUNT(id_mobil)` digunakan untuk menghitung jumlah mobil yang dipinjam oleh setiap peminjam. Hasilnya diberi alias `JumlahMobil` untuk memudahkan pembacaan hasil.

#### 2. FROM data\_mobil:

- Ini menentukan tabel sumber data, yaitu `data_mobil`.

#### 3. GROUP BY peminjam:

- Klausula `GROUP BY` digunakan untuk mengelompokkan data berdasarkan kolom `peminjam`. Semua baris dengan nilai `peminjam` yang sama akan dikelompokkan bersama.

## KESIMPULAN

Kode SQL ini membantu untuk menghitung dan menampilkan jumlah mobil yang dipinjam oleh setiap peminjam. Dengan mengelompokkan data berdasarkan kolom `peminjam` dan menghitung jumlah mobil yang dipinjam dalam setiap kelompok, kita bisa mendapatkan gambaran tentang seberapa sering masing-masing peminjam meminjam mobil. Hasil ini bisa digunakan untuk berbagai tujuan analitis, termasuk mengidentifikasi peminjam yang paling aktif.

**Berdasarkan query yang ada pada no 7 praktikum 5 tampilkan data pada tabel mobil dengan mengelompokkan berdasarkan**

**pemiliknya. Hitung menggunakan sum total pendapatan pemilik berdasarkan angka rental**

## **STRUKTUR**

```
SELECT nama_kolom SUM(nama_kolom) AS nama_ganti FROM nama_tabel GROUP BY  
nama_kolom;
```

## **CONTOH**

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

```
SELECT peminjam, SUM(harga_rental) AS total_pendapatan FROM data_mobil GROUP  
BY peminjam;
```

## **HASIL**





```
MariaDB [rental_farel]> SELECT * FROM data_mobil;
```

id_mobil	no_plat	no_mesin	warna	pemilik	peminjam	harga_rental
1	DD 2650 XY	ACX3560	HITAM	FAREL	TAUFIQ	50000
2	DD 2440 AX	BCS1120	MERAH	FATUR	AHSAN	100000
3	B 1611 QC	LSQ1112	SILVER	ASEP	CAPO	50000
4	DD 2901 JK	UQL1029	HITAM	JORDAN	NULL	150000
5	DD 2210 LS	CJH1011	HITAM	NAFAN	NULL	100000
6	DD 5346 XZ	NSR0077	MERAH	FACHRI	NULL	150000

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

  

```
MariaDB [rental_farel]> SELECT pemilik, SUM(harga_rental) AS total_pendapatan FROM data_mobil GROUP BY pemilik;
```

pemilik	total_pendapatan
ASEP	50000
FACHRI	150000
FAREL	50000
FATUR	100000
JORDAN	150000
NAFAN	100000

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

  

```
MariaDB [rental_farel]> SELECT peminjam, SUM(harga_rental) AS total_pendapatan FROM data_mobil GROUP BY peminjam;
```

peminjam	total_pendapatan
NULL	400000
AHSAN	100000
CAPO	50000
TAUFIQ	50000

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

## ANALISIS

- SELECT pemilik, SUM(harga\_rental) AS total\_pendapatan:**
  - pemilik:** Kolom dari tabel `mobil` yang menunjukkan pemilik dari mobil tersebut.
  - SUM(harga\_rental) AS total\_pendapatan:** Fungsi `SUM(harga_rental)` digunakan untuk menjumlahkan seluruh nilai `harga_rental` (pendapatan dari rental) untuk setiap pemilik. Hasilnya diberi alias `total_pendapatan` untuk memudahkan pembacaan hasil.
- FROM mobil:**
  - Menentukan tabel sumber data, yaitu `mobil`.
- GROUP BY pemilik:**
  - Klausula `GROUP BY` digunakan untuk mengelompokkan data berdasarkan kolom `pemilik`. Semua baris dengan nilai `pemilik` yang sama akan dikelompokkan bersama.
- SELECT peminjam, SUM(harga\_rental) AS total\_pendapatan:**
  - peminjam:** Kolom dari tabel `data_mobil` yang menunjukkan peminjam dari mobil tersebut.
  - SUM(harga\_rental) AS total\_pendapatan:** Fungsi `SUM(harga_rental)` digunakan untuk menjumlahkan seluruh nilai `harga_rental` (biaya yang dibayar oleh peminjam) untuk setiap peminjam. Hasilnya diberi alias `total_pendapatan` untuk memudahkan pembacaan hasil.
- FROM data\_mobil:**

- Menentukan tabel sumber data, yaitu `data_mobil`.

#### 6. GROUP BY **peminjam**:

- Klausula `GROUP BY` digunakan untuk mengelompokkan data berdasarkan kolom `peminjam`. Semua baris dengan nilai `peminjam` yang sama akan dikelompokkan bersama.

## KESIMPULAN

1. **Query pertama** menghitung total pendapatan dari rental mobil untuk setiap pemilik mobil dalam tabel `data_mobil`. Ini berguna untuk mengetahui seberapa banyak pendapatan yang dihasilkan oleh setiap pemilik mobil dari menyewakan mobil mereka.
2. **Query kedua** menghitung total biaya rental yang dibayar oleh setiap peminjam mobil dalam tabel `data_mobil`. Ini berguna untuk mengetahui seberapa banyak biaya yang dikeluarkan oleh setiap peminjam untuk menyewa mobil.

Kedua query ini memberikan wawasan yang berbeda: yang satu dari sudut pandang pemilik mobil dan yang lainnya dari sudut pandang peminjam mobil.

**Berdasarkan praktikum 5 no 8 tampilkan jumlah pemasukan pemilik berdasarkan harga rental kelompokkan berdasarkan pemiliknya dan seleksi yang total pemasukannya atau harga rentalnya mencapai lebih besar atau sama dengan 300k**

## STRUKTUR

```
SELECT nama_kolom SUM(nama_kolom) AS nama_ganti FROM nama_tabel GROUP BY
nama_kolom HAVING SUM(nama_kolom) kondisi;
```

## CONTOH

```
SELECT peminjam, SUM(harga_rental) AS TotalPemasukan FROM data_mobil GROUP
BY peminjam HAVING SUM(harga_rental) >= 300000;
```

## HASIL

```
MariaDB [rental_farel]> SELECT * FROM data_mobil;
+-----+-----+-----+-----+-----+-----+
| id_mobil | no_plat | no_mesin | warna | pemilik | peminjam | harga_rental |
+-----+-----+-----+-----+-----+-----+
| 1 | DD 2650 XY | ACX3560 | HITAM | FAREL | TAUFIQ | 50000 |
| 2 | DD 2440 AX | BCS1120 | MERAH | FATUR | AHSAN | 100000 |
| 3 | B 1611 QC | LSQ1112 | SILVER | ASEP | CAPO | 50000 |
| 4 | DD 2901 JK | UQL1029 | HITAM | JORDAN | NULL | 150000 |
| 5 | DD 2210 LS | CJH1011 | HITAM | NAFAN | NULL | 100000 |
| 6 | DD 5346 XZ | NSR0077 | MERAH | FACHRI | NULL | 150000 |
+-----+-----+-----+-----+-----+-----+
6 rows in set (0.001 sec)

MariaDB [rental_farel]> SELECT peminjam, SUM(harga_rental) AS TotalPemasukan FROM data_mobil GROUP BY peminjam HAVING SUM(harga_rental) >= 300000;
+-----+-----+
| peminjam | TotalPemasukan |
+-----+-----+
| NULL | 400000 |
+-----+-----+
1 row in set (0.006 sec)
```

## ANALISIS

- SELECT peminjam, SUM(harga\_rental) AS TotalPemasukan:**
  - peminjam:** Kolom dari tabel `data_mobil` yang menunjukkan peminjam mobil.
  - SUM(harga\_rental) AS TotalPemasukan:** Fungsi `SUM(harga_rental)` digunakan untuk menjumlahkan seluruh nilai `harga_rental` (total biaya rental) untuk setiap peminjam. Hasilnya diberi alias `TotalPemasukan` untuk memudahkan pembacaan hasil.
- FROM data\_mobil:**
  - Menentukan tabel sumber data, yaitu `data_mobil`.
- GROUP BY peminjam:**
  - Klausula `GROUP BY` digunakan untuk mengelompokkan data berdasarkan kolom `peminjam`. Semua baris dengan nilai `peminjam` yang sama akan dikelompokkan bersama.
- HAVING SUM(harga\_rental) >= 300000:**
  - Klausula `HAVING` digunakan untuk memfilter kelompok hasil yang sudah dikelompokkan. Dalam hal ini, hanya kelompok dengan total pendapatan (hasil dari `SUM(harga_rental)`) yang sama dengan atau lebih dari 300000 yang akan disertakan dalam hasil akhir.

## KESIMPULAN

Kode SQL ini membantu untuk menghitung dan menampilkan total biaya rental yang dikeluarkan oleh setiap peminjam yang jumlahnya mencapai atau melebihi 300000. Dengan mengelompokkan data berdasarkan kolom `peminjam` dan menggunakan klausa `HAVING` untuk memfilter hasil, kita bisa fokus pada peminjam dengan pengeluaran tinggi, yang dapat berguna untuk berbagai tujuan analitis dan pengambilan keputusan terkait peminjaman mobil.

## Berdasarkan praktikum 5 no 12 tampilkan rata" pemasukan pemilik mobil kelompokkan

# berdasarkan pemiliknya

## STRUKTUR

```
SELECT nama_kolom AVG(nama_kolom) AS nama_ganti FROM nama_tabel GROUP BY
nama_kolom;
```

## CONTOH

```
SELECT pemilik, AVG(harga_rental) AS total_pemasukan FROM data_mobil GROUP
BY pemilik;
```

```
SELECT peminjam, AVG(harga_rental) AS total_pemasukan FROM data_mobil GROUP
BY peminjam;
```

## HASIL

```
MariaDB [rental_farel]> SELECT * FROM data_mobil;
```

	id_mobil	no_plat	no_mesin	warna	pemilik	peminjam	harga_rental
	1	DD 2650 XY	ACX3560	HITAM	FAREL	TAUFIQ	50000
	2	DD 2440 AX	BCS1120	MERAH	FATUR	AHSAN	100000
	3	B 1611 QC	LSQ1112	SILVER	ASEP	CAPO	50000
	4	DD 2901 JK	UQL1029	HITAM	JORDAN	NULL	150000
	5	DD 2210 LS	CJH1011	HITAM	NAFAN	NULL	100000
	6	DD 5346 XZ	NSR0077	MERAH	FACHRI	NULL	150000

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

```
MariaDB [rental_farel]> SELECT pemilik, AVG(harga_rental) AS total_pemasukan FROM data_mobil GROUP BY pemilik;
```

pemilik	total_pemasukan
ASEP	50000.0000
FACHRI	150000.0000
FAREL	50000.0000
FATUR	100000.0000
JORDAN	150000.0000
NAFAN	100000.0000

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

```
MariaDB [rental_farel]> SELECT peminjam, AVG(harga_rental) AS total_pemasukan FROM data_mobil GROUP BY peminjam;
```

peminjam	total_pemasukan
NULL	133333.3333
AHSAN	100000.0000
CAPO	50000.0000
TAUFIQ	50000.0000

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

# ANALISIS

## 1. **SELECT** **pemilik**, **AVG (harga\_rental)** **AS** **total\_pemasukan**:

- **pemilik**: Kolom dari tabel `data_mobil` yang menunjukkan pemilik dari mobil tersebut.
- **AVG (harga\_rental) AS total\_pemasukan**: Fungsi `AVG(harga_rental)` digunakan untuk menghitung nilai rata-rata dari `harga_rental` untuk setiap pemilik. Hasilnya diberi alias `total_pemasukan` untuk memudahkan pembacaan hasil.

## 2. **FROM** **data\_mobil**:

- Menentukan tabel sumber data, yaitu `data_mobil`.

## 3. **GROUP BY** **pemilik**:

- Klausa `GROUP BY` digunakan untuk mengelompokkan data berdasarkan kolom `pemilik`. Semua baris dengan nilai `pemilik` yang sama akan dikelompokkan bersama.

## 4. **SELECT** **peminjam**, **AVG (harga\_rental)** **AS** **total\_pemasukan**:

- **peminjam**: Kolom dari tabel `data_mobil` yang menunjukkan peminjam mobil.
- **AVG (harga\_rental) AS total\_pemasukan**: Fungsi `AVG(harga_rental)` digunakan untuk menghitung nilai rata-rata dari `harga_rental` yang dibayar oleh setiap peminjam. Hasilnya diberi alias `total_pemasukan` untuk memudahkan pembacaan hasil.

## 5. **FROM** **data\_mobil**:

- Menentukan tabel sumber data, yaitu `data_mobil`.

## 6. **GROUP BY** **peminjam**:

- Klausa `GROUP BY` digunakan untuk mengelompokkan data berdasarkan kolom `peminjam`. Semua baris dengan nilai `peminjam` yang sama akan dikelompokkan bersama.

# KESIMPULAN

1. **Query pertama** menghitung rata-rata pendapatan dari rental mobil untuk setiap pemilik mobil dalam tabel `data_mobil`. Ini berguna untuk memahami seberapa besar pendapatan rata-rata yang diperoleh setiap pemilik dari menyewakan mobil mereka.
2. **Query kedua** menghitung rata-rata biaya rental yang dibayar oleh setiap peminjam mobil dalam tabel `data_mobil`. Ini berguna untuk memahami seberapa besar rata-rata pengeluaran setiap peminjam untuk menyewa mobil.

Kedua query ini memberikan wawasan yang berbeda: satu dari sudut pandang pemilik mobil dan yang lainnya dari sudut pandang peminjam mobil, dengan fokus pada rata-rata pendapatan atau biaya rental.

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

## **STRUKTUR**

```
SELECT nama_kolom MAX(nama_kolom) AS nama_ganti, MIN(nama_kolom) AS
nama_ganti FROM nama_tabel GROUP BY nama_kolom HAVING COUNT(nama_kolom)
kondisi;
```

## **CONTOH**

```
SELECT pemilik, MAX(harga_rental) AS PemasukanTerbesar, MIN(harga_rental) AS
PemasukanTerkecil FROM data_mobil GROUP BY pemilik HAVING COUNT(id_mobil) >=
1;
```

```
SELECT peminjam, MAX(harga_rental) AS PemasukanTerbesar, MIN(harga_rental)
AS PemasukanTerkecil FROM data_mobil GROUP BY peminjam HAVING
COUNT(id_mobil) > 1;
```

## **HASIL**

```

MariaDB [rental_farel]> SELECT * FROM data_mobil;
+-----+-----+-----+-----+-----+-----+-----+
| id_mobil | no_plat | no_mesin | warna | pemilik | peminjam | harga_rental |
+-----+-----+-----+-----+-----+-----+-----+
| 1 | DD 2650 XY | ACX3560 | HITAM | FAREL | TAUFIQ | 50000 |
| 2 | DD 2440 AX | BCS1120 | MERAH | FATUR | AHSAN | 100000 |
| 3 | B 1611 QC | LSQ1112 | SILVER | ASEP | CAPO | 50000 |
| 4 | DD 2901 JK | UQL1029 | HITAM | JORDAN | NULL | 150000 |
| 5 | DD 2210 LS | CJH1011 | HITAM | NAFAN | NULL | 100000 |
| 6 | DD 5346 XZ | NSR0077 | MERAH | FACHRI | NULL | 150000 |
+-----+-----+-----+-----+-----+-----+-----+
6 rows in set (0.001 sec)

MariaDB [rental_farel]> SELECT pemilik, MAX(harga_rental) AS PemasukanTerbesar, MIN(harga_rental) AS PemasukanTerkecil FROM data_mobil GROUP BY pemilik HAVING COUNT(id_mobil) > 1;
Empty set (0.005 sec)

MariaDB [rental_farel]> SELECT pemilik, MAX(harga_rental) AS PemasukanTerbesar, MIN(harga_rental) AS PemasukanTerkecil FROM data_mobil GROUP BY pemilik HAVING COUNT(id_mobil) >= 1;
+-----+-----+-----+
| pemilik | PemasukanTerbesar | PemasukanTerkecil |
+-----+-----+-----+
| ASEP | 50000 | 50000 |
| FACHRI | 150000 | 150000 |
| FAREL | 50000 | 50000 |
| FATUR | 100000 | 100000 |
| JORDAN | 150000 | 150000 |
| NAFAN | 100000 | 100000 |
+-----+-----+-----+
6 rows in set (0.001 sec)

MariaDB [rental_farel]> SELECT peminjam, MAX(harga_rental) AS PemasukanTerbesar, MIN(harga_rental) AS PemasukanTerkecil FROM data_mobil GROUP BY peminjam HAVING COUNT(id_mobil) > 1;
+-----+-----+-----+
| peminjam | PemasukanTerbesar | PemasukanTerkecil |
+-----+-----+-----+
| NULL | 150000 | 100000 |
+-----+-----+-----+
1 row in set (0.002 sec)

```

## ANALISIS

- SELECT pemilik, MAX(harga\_rental) AS PemasukanTerbesar, MIN(harga\_rental) AS PemasukanTerkecil:**
  - pemilik:** Kolom dari tabel `data_mobil` yang menunjukkan pemilik dari mobil tersebut.
  - MAX(harga\_rental) AS PemasukanTerbesar:** Fungsi `MAX(harga_rental)` digunakan untuk menemukan nilai terbesar dari `harga_rental` untuk setiap pemilik. Hasilnya diberi alias `PemasukanTerbesar`.
  - MIN(harga\_rental) AS PemasukanTerkecil:** Fungsi `MIN(harga_rental)` digunakan untuk menemukan nilai terkecil dari `harga_rental` untuk setiap pemilik. Hasilnya diberi alias `PemasukanTerkecil`.
- FROM data\_mobil:**
  - Menentukan tabel sumber data, yaitu `data_mobil`.
- GROUP BY pemilik:**
  - Klausula `GROUP BY` digunakan untuk mengelompokkan data berdasarkan kolom `pemilik`. Semua baris dengan nilai `pemilik` yang sama akan dikelompokkan bersama.
- HAVING COUNT(id\_mobil) >= 1:**
  - Klausula `HAVING` digunakan untuk memfilter kelompok hasil yang sudah dikelompokkan. Dalam hal ini, hanya kelompok dengan jumlah mobil (hasil dari

`COUNT(id_mobil)` ) yang lebih dari atau sama dengan 1 yang akan disertakan dalam hasil akhir.

5. **SELECT** `peminjam, MAX(harga_rental) AS PemasukanTerbesar, MIN(harga_rental) AS PemasukanTerkecil`:

- **peminjam**: Kolom dari tabel `data_mobil` yang menunjukkan peminjam mobil.
- **MAX(harga\_rental) AS PemasukanTerbesar**: Fungsi `MAX(harga_rental)` digunakan untuk menemukan nilai terbesar dari `harga_rental` yang dibayar oleh setiap peminjam. Hasilnya diberi alias `PemasukanTerbesar`.
- **MIN(harga\_rental) AS PemasukanTerkecil**: Fungsi `MIN(harga_rental)` digunakan untuk menemukan nilai terkecil dari `harga_rental` yang dibayar oleh setiap peminjam. Hasilnya diberi alias `PemasukanTerkecil`.

6. **FROM** `data_mobil`:

- Menentukan tabel sumber data, yaitu `data_mobil`.

7. **GROUP BY** `peminjam`:

- Klausula `GROUP BY` digunakan untuk mengelompokkan data berdasarkan kolom `peminjam`. Semua baris dengan nilai `peminjam` yang sama akan dikelompokkan bersama.

8. **HAVING** `COUNT(id_mobil) > 1`:

- Klausula `HAVING` digunakan untuk memfilter kelompok hasil yang sudah dikelompokkan. Dalam hal ini, hanya kelompok dengan jumlah mobil (hasil dari `COUNT(id_mobil)` ) yang lebih dari 1 yang akan disertakan dalam hasil akhir.

## KESIMPULAN

1. **Query pertama** menghitung nilai terbesar dan terkecil dari pendapatan rental mobil untuk setiap pemilik mobil dalam tabel `data_mobil`, dengan syarat pemilik tersebut memiliki setidaknya satu mobil yang terdaftar.
2. **Query kedua** menghitung nilai terbesar dan terkecil dari biaya rental yang dibayar oleh setiap peminjam mobil dalam tabel `data_mobil`, dengan syarat peminjam tersebut telah meminjam lebih dari satu mobil.

Kedua query ini memberikan wawasan yang berbeda: satu dari sudut pandang pemilik mobil dan yang lainnya dari sudut pandang peminjam mobil, dengan fokus pada nilai maksimal dan minimal dari pendapatan atau biaya rental.