

Tabel Keaktifan

No	Nama	Peran	Nilai
1	Fadhil Amir	Membuat soal + membuat praktikum 5	3
2	Rayhan Alfazari	Membuat soal + membuat praktikum 5	3
3	M Nafan Nabil	Membuat soal + Mencari ket. opsi jawaban	3
4	Fachri Ramadhan	Membuat soal + membuat Ppt	3

Keterangan :

0 : Tidak Aktif

1 : Kurang Aktif

2 : Cukup Aktif

3 : Sangat Aktif

Tabel Pegawai

```
MariaDB [company_fadhil]> desc pegawai;
```

Field	Type	Null	Key	Default	Extra
NIP	int(11)	NO	PRI	NULL	
NDep	varchar(50)	YES		NULL	
NBlk	varchar(50)	YES		NULL	
JK	enum('P','L')	YES		NULL	
Alamat	varchar(100)	YES		NULL	
Telp	varchar(15)	YES		NULL	
Jabatan	varchar(20)	YES		NULL	
Gaji	bigint(10)	YES		NULL	
NoCab	varchar(10)	YES		NULL	

```
9 rows in set (1.285 sec)
```

```
MariaDB [company_fadhil]> select * from pegawai;
```

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

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

Query

```
SELECT COUNT(NIP) AS JumlahPegawai, COUNT(jabatan) AS JumlahJabatan FROM pegawai;
```

Hasil

```
MariaDB [company_fadhil]> select count(nip) as jumlahpegawai, count(jabatan) as jumlahjabatan from pegawai
+-----+-----+
| jumlahpegawai | jumlahjabatan |
+-----+-----+
|          9   |          8   |
+-----+-----+
1 row in set (0.001 sec)

MariaDB [company_fadhil]>
```

Analisis

- **SELECT** : Untuk memilih kolom apa saja yang ingin dipilih (untuk dihitung)
 - **COUNT(NIP)** : Untuk menghitung jumlah barisan data yang mempunyai isi data dari kolom yang dipilih. **NIP** adalah nama kolom yang dipilih untuk dihitung
 - **AS** : Untuk mengubah nama dari suatu kolom untuk sementara
 - **"JumlahPegawai"** : merupakan nama ubahan dari perintah **AS** yang digunakan merupakan nama sementara dari perintah **COUNT(NIP)**
 - **COUNT(jabatan)** : untuk menghitung jumlah barisan data yang mempunyai isi data dari kolom yang dipilih **jabatan** adalah nama kolom yang dipilih untuk dihitung
 - **AS** : Untuk mengubah nama dari suatu kolom untuk sementara
 - **"JumlahJabatan"** : merupakan nama sementara dari perintah **COUNT(jabatan)**
 - **FROM "pegawai2"** : merupakan dari tabel mana datanya yang digunakan **"pegawai2"** adalah nama tabel yang datanya ingin digunakan
 - **Hasilnya** : karena ada 9 barisan data, yang ingin dihitung adalah kolom **NIP**, jumlah dari kolom **NIP** (isi datanya) ada 9, ditampilkan sebagai **"JumlahPegawai"**. Kolom **"jabatan"** juga dihitung, akan tetapi ada satu data yang berisi **NULL** (kosong) oleh karena itu hanya ada 8 data ditampilkan sebagai **"JumlahJabatan"**
-

2

Contoh query

```
select COUNT(NIP) AS jumlahpegawai  
from pegawai  
where NoCab = 'C102';
```

Hasil

```
MariaDB [company_fadhil]> select count(nip) as jumlahpegawai  
-> from pegawai  
-> where NoCab = 'C102';  
+-----+  
| jumlahpegawai |  
+-----+  
|          3 |  
+-----+  
1 row in set (0.199 sec)
```

Analisis

- **Select** = untuk memilih kolom mana saja yang ingin dipilih untuk dihitung.
- **COUNT(NIP)** = untuk menghitung jumlah barisan data yang mempunyai data dari kolom yang dipilih.
Nip adalah nama kolom yang dipilih untuk dihitung.
- **AS** = untuk mengubah nama dari suatu kolom sementara.
- **JumlahPegawai** = nama sementara yang dipilih untuk kolom COUNT(NIP)
- **FROM Pegawai** = dari tabel mana datanya akan digunakan.
Pegawai adalah nama tabel yang dipilih untuk digunakan.
- **WHERE** = merupakan kondisi yang harus dipenuhi agar datanya dapat dihitung dengan query COUNT(NIP).
- **(Nocab = 'C102')** = adalah kondisi dari where yang harus dipenuhi, jadi hanya barisan data yang memiliki 'C102' di kolom "Nocab" yang bisa dihitung.
- **Hasilnya** = di 9 barisan data yang ada pada tabel pegawai, kita ingin menghitung jumlah barisan data yang memiliki nilai 'C102' pada kolom "Nocab"nya dengan menggunakan COUNT. Jadi yang muncul adalah 3 barisan data. Kita juga ingin mengubah nama dari kolom hasil perintah COUNT secara sementara dengan perintah AS namanya adalah JumlahPegawai.

3

Contoh query

```
SELECT Nocab, COUNT(NIP) AS Jumlah_pegawai
  from pegawai
 GROUP BY Nocab;
```

Hasil

```
MariaDB [company_fadhil]> select Nocab, count(nip) as jumlah_pegawai
-> from pegawai
-> group by Nocab;
+-----+-----+
| Nocab | jumlah_pegawai |
+-----+-----+
| C101  |                2 |
| C102  |                3 |
| C103  |                2 |
| C104  |                2 |
+-----+-----+
4 rows in set (0.126 sec)
```

Analisis

- **SELECT** = untuk memilih kolom mana saja yang ingin dihitung atau ditampilkan.
- **Nocab** = merupakan nama kolom yang ingin ditampilkan.
- **‘COUNT(NIP)’** = untuk menghitung jumlah barisan data yang mempunyai isi data dari kolom yang dipilih.
Nip adalah nama kolom yang ingin dipilih untuk dihitung.
- **AS** = untuk mengubah nama dari suatu kolom untuk sementara.
- **Jumlah Pegawai** = merupakan nama sementara dari kolom hasil COUNT(NIP).
- **From Pegawai** = dari tabel mana yang data kodomnya ingin digunakan.
Pegawai adalah nama tabel yang dipilih untuk digunakan.
- **GROUP BY** = untuk mengelompokkan data berdasarkan nilai data yang telah ditentukan Pada kolom yang dipilih.
- **Nocab** = nama kolom Yang dipilih untuk datanya dikelompokkan.
- **Hasilnya** = Berdasarkan 9 barisan data, masing-masing nilai dalam kolom Nocab dikelompokkan berdasarkan nilainya sendiri. Jadi Nocab clol bersama Nocab yang nilai

nya sama Yaitu clol. Jadi Nocab Yang memiliki C101 ada 2, C102 ada 3, C103 ada 2, C104 ada 2. Total semuanya 9, sesuai densan Jumlah barisan data Yand ada. Adapun nama dari kolom hasil Yaitu Jumlah-Pegawai dari Perintah AS.

4

Contoh query

```
SELECT Nocab, COUNT(NIP) AS Jumlah_pegawai  
-> from pegawai  
-> GROUP BY NoCab HAVING COUNT(NIP) >= 3;
```

Hasil

```
MariaDB [company_fadhil]> select NoCab, count(nip) as jumlah_pegawai  
-> from pegawai  
-> group by NoCab having count(nip) >= 3;  
+-----+-----+  
| NoCab | jumlah_pegawai |  
+-----+-----+  
| C102 | 3 |  
+-----+-----+  
1 row in set (0.127 sec)
```

Analisis

- **SELECT** = untuk memilih kolom mana saja yang ingin dihitung atau ditampilkan.
- **Nocab** = merupakan nama kolom yang ingin ditampilkan.
- **COUNT (NIP)** = untuk menghitung Jumlah barisan data yang mempunyai isi data dari kolom Yang dipilih.
NIP adalah nama kolom Yang dipilih untuk dihitung.
- **AS** = untuk mengubah nama dari suatu kolom untuk sementara.
- **Jumlah_Pegawai** = nama sementara dari kolom hasil COUNT (NIP).
- **From Pegawai** = untuk memilih dari tabel mana Yand data kolomnya ingin digunakan.
Pegawai adalah nama tabel Xang dipilih untuk digunakan.
- **GROUP BY** = untuk menjelompokkan data berdasarkan nilai data Yang telah ditentukan
Pada kolom Yang dipilih.

- Nocab - nama kolom Yang dipilih untuk dikelompokkan datanya.
- HAVING = untuk menentukan kondisi (Yand hans dipenuhi) oleh suatu kelompok data agar bisa ditampilkan.
- (COUNT (NIP) >= 3) = merupakan kondisi Yand harus dipenuhi oleh suatu kelompok data. Jadi hanya kelompok data Yang hasil hitungannya lebih atau Sama dengan 3.
- Hasilnya seperti sebelumnya, ada 9 barisan data dibagi sesuai Nocab nya masing- masing. Namun Yand ingin ditampilkan adalah hasil hitungan yang lebih dari atau sama dengan 3. Yaitu Nocab C102 Yang ada 3. Yand Jain c101 ada 2, c103 ada 2, C104 ada 2.

5

Query

```
Select SUM(Gaji) AS Total_Gaji
->FROM pegawai;
```

##Hasil

```
MariaDB [company_fadhil]> select sum(gaji) as total_gaji
-> from pegawai;
+-----+
| total_gaji |
+-----+
| 30575000 |
+-----+
1 row in set (0.113 sec)
```

Analisis

- SELECT = Untuk memilih kolom mana saja yang dipilih untuk dijumlahkan.
- SUM(Gaji) = Untuk menghitung jumlah data (khusus angka) pada kolom yang harus dipilih. Gaji merupakan nama kolom yang dipilih untuk dihitung jumlah isi datanya.
- AS = Untuk mengganti nama dari kolom hasil SUM(Gaji) untuk sementara.
- Total_Gaji = Merupakan nama sementara dari perintah AS.
- FROM pegawai = Untuk memilih dari table mana yang kolom datanya akan digunakan. Pegawai adalah nama dari table yang dipilih.
- Hasilnya = kolom Gaji yang isi datanya berupa angka-angka, semuanya dijumlahkan menjadi satu seperti ditotalkan (sama seperti matematika pada umumnya). Dan hasilnya

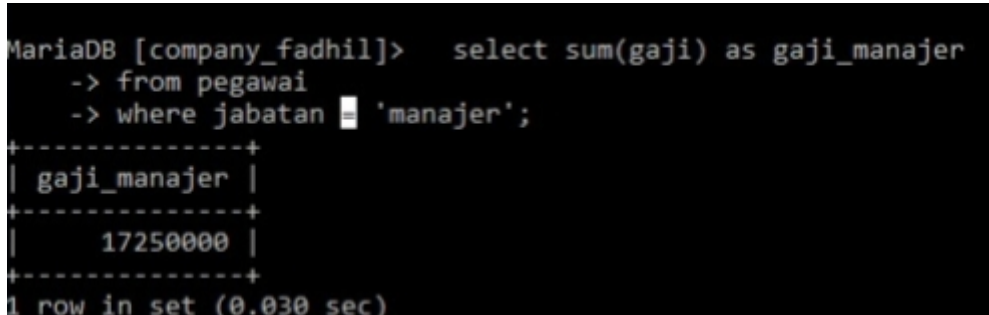
adalah 30575000. Adapun nama kolom dari hasil jumlah tersebut diubah dari SUM(Gaji) menjadi Total_Gaji.

6

Query

```
Select SUM(Gaji) AS Gaji_Manager  
->FROM pegawai  
->WHERE Jabatan = 'Manager';
```

Hasil



```
MariaDB [company_fadhil]> select sum(gaji) as gaji_manajer  
-> from pegawai  
-> where jabatan = 'manajer';  
+-----+  
| gaji_manajer |  
+-----+  
|      17250000 |  
+-----+  
1 row in set (0.030 sec)
```

Analisis

- **Select** = untuk memiliki kolom mana saja yang dipilih untuk digunakan.
 - **SUM** = untuk menghitung jumlah isi data (khusus angka) pada kolom yang dipilih. gaji yang dipilih untuk dijumlahkan isi datanya.
 - **AS** = mengganti nama dari kolom hasil SUM(Gaji) secara sementara.
 - **Gaji_Manager** = mengubah nama sementara dari perintah AS.
 - **FROM pegawai** = untuk memilih dari tabel mana yang kolom datanya akan digunakan pegawai adalah nama dari tabel yang dipilih.
 - **WHERE** = kondisi yang harus dipenuhi oleh suatu kolom agar datanya bisa dijumlah
 - **(Jabatan="Manajer")** = merupakan kondisi dari WHERE. hanya barisan data yang kolom Gaji-nya bisa dijumlahkan.
 - **Hasilnya** = barisan data yang kolom jumlahnya berisi manager akan dijumlah kolom Gaji-nya menjadi 17.250.000. jadi hanya beberapa kolom saja yang dijumlah.
-

7

Query

```
Select NoCab, SUM(Gaji) TotalGaji  
->FROM pegawai  
->GROUP BY NoCab;
```

Hasil

```
MariaDB [company_fadhil]> select NoCab, sum(gaji) as total_gaji  
-> from pegawai  
-> group by NoCab;
```

NoCab	total_gaji
C101	7750000
C102	9450000
C103	9000000
C104	4375000

rows in set (0.080 sec)

Analisis

- **SELECT** = untuk memilih kolom mana saja yang dipilih untuk ditampilkan atau dijumlahkan.
- **SUM(Gaji)** = untuk menghitung jumlah data (khusus angka) pada kolom yang dipilih. Gaji adalah nama kolom yang dipilih untuk dijumlahkan isi datanya.
- **AS** = untuk mengganti nama dari kolom hasil SUM(Gaji) untuk sementara.
- **TOTALGaji** = merupakan nama sementara dari perintah AS.
- **FROM pegawai** = untuk memilih dari tabel mana yang data kolomnya akan digunakan. pegawai adalah nama tabel yang dipilih.
- **GROUP BY** = untuk mengelompokkan data berdasarkan nilai data yang telah ditentukan pada kolo yang dipilih.
- **NoCab** = nama kolom yang datanya dipilih untuk dikelompokkan.
- **Hasilnya** = jadi berdasarkan kolom NoCab, barisan data yang kolom NoCab-nya berisi C102 maka kolom gaji dari barisan data itu digunakan bersama barisan data yang meiliki NoCab C101 juga. maka kolom gaji dijumlahkan sesuai dengan kolom NOCab masing-masing, mulai dari C101 memiliki 2 kolom gaji yang bisa dijumlahkan. sama denfan C103 dand C104. adapun C102 memiliki 3 kolom Gaji yanf dapat dijumlahkan. TotalGaji merupakan hasil perintah dari AS untuk mengubah nama kolom gasil dari SUM(Gaji).

8

Query

```
SELECT noCab, SUM(Gaji) AS Total_Gaji
from pegawai
GROUP BY noCab HAVING SUM(Gaji) >= 8000000;
```

Hasil

```
MariaDB [company_fadhil]> select NoCab, sum(gaji) as total_gaji
-> from pegawai
-> group by NoCab having sum(gaji) >= 8000000;
+-----+-----+
| NoCab | total_gaji |
+-----+-----+
| C102  | 9450000    |
| C103  | 9000000    |
+-----+-----+
2 rows in set (0.057 sec)
```

Analisis

- `select` untuk memilih kolom mana saja yang dipilih untuk di tampilkan atau di jumlah.
- `noCab` nama kolom yang dipilih untuk ditampilkan.
- `SUM(Gaji)` untuk menghitung jumlah data (Khusus Angka) pada kolom yang dipilih. `Gaji` adalah nama kolom yang dipilih untuk dijumlahkan isi datanya.
- `AS` untuk mengganti nama dari `SUM(Gaji)` untuk sementara.
- `Gaji` adalah nama sementara dari perintah `AS`.
- `From Pegawai` untuk memilih dari tabel mana yang data kolomnya akan di gunakan, `Pegawai` Adalah Nama Tabel Yang akan di pilih.
- `GROUP BY` untuk mengelompokkan data berdasarkan nilai data Yang telah ditentukan Pada kolom yang dipilih.
- `Nocab` nama kolom Yang dipilih untuk datanya dikelompokkan.
- `Having` kondisi yang harus dipenuhi oleh suatu kelompok data agar bisa ditampilkan.
- `(Sum (Gaji) >= 8000000)` kondisi dari `HAVING`, Hasil dari Penjumlahan Gai yang hanya bisa ditampilkan adalah Hasil yang lebih dari atau sama dengan 8000000.
- `Hasilnya` Sama seperti sebelumnya, tetapi `Nocab` Yang memenuhi kondisi tersebut hanyalah C102 dan C103 karena hasil Jumlah kolom Gaji nya lebih dari atau sama

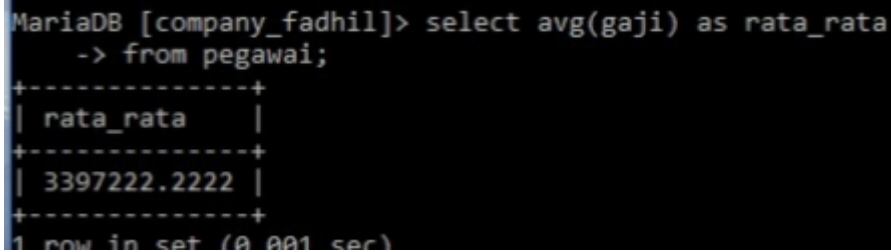
dengan 8000000. Adapun hasil kolom Sum(Gaji) diganti Jadi Total-Gaji.

9

Query

```
SELECT AVG(Gaji) AS rata_rata from pegawai;
```

Hasil



```
MariaDB [company_fadhil]> select avg(gaji) as rata_rata
-> from pegawai;
+-----+
| rata_rata |
+-----+
| 3397222.2222 |
+-----+
1 row in set (0.001 sec)
```

Analisis

- `Select` untuk memilih kolom mana data Yang dipilih untuk ditampilkan.
 - `AVG (Gaji)` untuk menghitung rata-rata dari data yang ada Pada kolom Yang dipilih. Gaji adalah nama kolom Yang dipilih untuk dihitung rata-ratanya.
 - `As` untuk mengganti nama dari kolom hasil `AVG (Gaji)` untuk sementara.
 - `Rata-rata` nama sementara dari Perintah `AS`.
 - `From Pegawai` untuk memilih dari tabel mana Yang data kolomnya ingin digunakan
 - `Pegawai` adalah nama dari tabel yang dipilih.
 - Hasilnya `3397222.2222` merupakan hasil rata-rata dari semua 9 barisan data Pada kolom Gaji. Adapun nama kolom hasil dari `AVG (Gaji)` Yaitu Rata-rata
-

10

Query

```
SELECT AVG(Gaji) AS GajiRataManager FROM pegawai WHERE Jabatan = "Manager";
```

Hasil

```
MariaDB [company_fadhil]> select avg(gaji) as gajiratamgr  
-> from pegawai  
-> where jabatan = 'manajer';  
+-----+  
| gajiratamgr |  
+-----+  
| 5750000.0000 |  
+-----+  
1 row in set (0.001 sec)
```

Analisis

- `SELECT` untuk memilih kolom mana saja yang dipilih untuk ditampilkan.
- `AVG (Gaji)` untuk menghitung rata-rata dari data yang ada pada kolom Yang dipilih. Gaji adalah nama kolom Yang dipilih untuk dihitung rata-ratanya.
- `AS` untuk mengganti nama dari kolom hasil `AVG (Gaji)` untuk sementara.
- `GajiRataManager` nama sementara dari Perintah `AS`.
- `FROM pegawai` untuk memilih dari tabel mana Yang data kolomnya ingin Pegawai adalah nama dari tabel Yang diinginkan digunakan.
- `WHERE` Kondisi yang harus dipenuhi oleh suatu kolom agar datanya bisa dihitung rata-ratanya
- `(Jabatan = 'Manajer')` kondisi dari `WHERE`. Barisan data yang kolom Jabatannya Manajer akan dihitung rata-rata kolom Gaji nya.
- Hasilnya `5750000.0000` merupakan hasil hitung rata-rata dari barisan data yang memiliki manajer di kolom Jabatan nya,dari situ kolom Gajinya di hitung.

11

Query

```
SELECT nocab, AVG(gaji) AS ratagaji FROM pegawai GROUP BY nocab;
```

Hasil

```
MariaDB [company_fadhil]> select NoCab, avg(gaji) as ratagaji
-> from pegawai
-> group by NoCab;
+-----+-----+
| NoCab | ratagaji |
+-----+-----+
| C101  | 3875000.0000 |
| C102  | 3150000.0000 |
| C103  | 4500000.0000 |
| C104  | 2187500.0000 |
+-----+-----+
4 rows in set (0.023 sec)
```

Analisis

- `SELECT` untuk memilih kolom mana saja yang dipilih untuk ditampilkan/dihitung.
- `nocab` kolom yang dipilih untuk ditampilkan.
- `AVG(gaji)` untuk menghitung rata-rata dari data yang ada pada kolom yang dipilih. `gaji` adalah nama kolom yang dipilih untuk dihitung rata-ratanya.
- `AS` untuk mengganti nama dari kolom hasil `AVG(gaji)` untuk sementara.
- `RataGaji` adalah nama sementara dari perintah `AS`.
- `FROM pegawai` untuk memilih dari tabel mana yang data kolomnya ingin digunakan. `Pegawai` adalah nama dari tabel yang dipilih.
- `GROUP BY` untuk mengelompokkan data berdasarkan nilai data yang telah ditentukan pada kolom yang dipilih.
- `nocab` nama kolom yang dipilih untuk datanya dikelompokkan.
- hasilnya hampir sama seperti no.7 masing-masing kolom `nocab` dihitung rata-ratanya. Sesuai dengan isi `nocab`, jadi yang `C101` dihitung dengan `C101` yang lainnya juga (yang sama) adapun `ratagaji` merupakan nama sementara dari kolom hasil `AVG(gaji)`.

12

Query

```
SELECT nocab, AVG(gaji) AS Ratagaji FROM pegawai GROUP BY nocab HAVING nocab =
'C101' OR nocab = 'C102';
```

Hasil

```
MariaDB [company_fadhil]> select NoCab, avg(gaji) as ratagaji
-> from pegawai
-> group by NoCab having NoCab= 'C101' OR NoCab = 'C102';
```

NoCab	ratagaji
C101	3875000.0000
C102	3150000.0000

2 rows in set (0.088 sec)

Analisis

- `SELECT` untuk memilih kolom mana saja yang dipilih untuk ditampilkan/dihitung.
- `nocab` kolom yang dipilih untuk ditampilkan.
- `AVG(gaji)` untuk menghitung rata-rata dari data yang ada pada kolom yang dipilih. `gaji` adalah nama kolom yang dipilih untuk dihitung rata-ratanya.
- `AS` untuk mengganti nama dari kolom hasil `AVG(gaji)` untuk sementara.
- `RataGaji` adalah nama sementara dari perintah `AS`.
- `FROM pegawai` untuk memilih dari tabel mana yang data kolomnya ingin digunakan. Pegawai adalah nama dari tabel yang dipilih.
- `GROUP BY` untuk mengelompokkan data berdasarkan nilai data yang telah ditentukan pada kolom yang dipilih.
- `nocab` nama kolom yang dipilih untuk datanya dikelompokkan.
- `HAVING` kondisi yang harus dipenuhi oleh suatu kelompok data.
- `nocab = 'C101' OR nocab = 'C102'` merupakan kondisi dari `HAVING`. Jadi kolom `nocab` yang memiliki C101 atau C102 yang hanya akan ditampilkan OR adalah kondisi yang hanya salah satu datanya yang harus dipenuhi.

13

Query

```
SELECT MAX(gaji) AS gajiterbesar, MIN(gaji) AS gajiterkecil FROM pegawai;
```

Hasil

```

MariaDB [company_fadhil]> \
MariaDB [company_fadhil]> select max(gaji) as gajiterbesar, min(gaji) as gajiterkecil
-> from pegawai;
+-----+-----+
| gajiterbesar | gajiterkecil |
+-----+-----+
|      6250000 |      1725000 |
+-----+-----+
1 row in set (0.018 sec)

```

Analisis

- **SELECT** untuk memilih kolom mana saja yang dipilih untuk ditampilkan/dihitung.
- **MAX(gaji)** untuk menampilkan nilai maksimum atau terbesar/tertinggi dari suatu data dalam kolom yang dipilih. Gaji adalah nama kolom yang dipilih.
- **AS gajiterbesar** untuk mengganti nama dari kolom hasil **MAX(gaji)** menjadi nama sementara yaitu gajiterbesar.
- **MIN(gaji)** untuk menampilkan nilai minimum atau terkecil/terendah dari suatu data dalam kolom yang dipilih. gaji adalah nama kolom yang dipilih.
- **AS gajiterkecil** untuk mengganti nama dari kolom hasil **MIN(gaji)** menjadi gajiterkecil untuk sementara.
- **FROM pegawai** untuk memilih dari tabel mana yang data kolomnya ingin digunakan. Pegawai adalah nama dari tabel yang dipilih.
- hasilnya jadi dari 9 nilai yang ada di kolom gaji, gaji maksimumnya adalah 6250000 dan namanya diubah menjadi gajiterbesar, bagi minimumnya adalah 1725000 dan namanya diubah menjadi gajiterkecil.

14

Query

```

SELECT MAX(Gaji) AS GajiTerbesar, MIN(Gaji) AS GajiTerkecil
FROM pegawai
WHERE Jabatan = "Manajer";

```

Hasil

```

MariaDB [company_fadhil]> select max(gaji) as gajiterbesar, min(gaji) as gajiterkecil
-> from pegawai
-> where jabatan = 'manajer';
+-----+-----+
| gajiterbesar | gajiterkecil |
+-----+-----+
|      6250000 |      5250000 |
+-----+-----+
1 row in set (0.005 sec)

```

Analisis

- **SELECT** = Untuk memilih kolom nama mana saja yang dipilih untuk ditampilkan I.
- **MAX(Gaji)** = Untuk menampilkan nilai terbesar dari suatu data dalam kolom yang dipilih. Gaji adalah nama kolom yang dipilih.
- **AS GajiTerbesar** = Untuk Menganti nama dari kolom hasil **MAX(Gaji)** menjadi **GajiTerbesar** untuk sementara.
- **MIN(Gaji)** = Untuk menampilkan nilai terkecil dari suatu data dalam kolom yang dipilih. Gaji adalah nama kolom yang dipilih.
- **AS GajiTerkecil** = Untuk Menganti nama dari kolom hasil **MIN(Gaji)** menjadi **GajiTerkecil** untuk sementara.
- **FROM pegawai** = Untuk memilih dari tabel mana yang data kolomnya ingin ditampilkan.
- **WHERE** = Kondisi yang harus dipenuhi oleh suatu kolom data agar bisa ditampilkan.
- **(Jabatan = "Manajer")** = Kondisi dari **WHERE** yang harus dipenuhi. Barisan data yang kolom jabatannya berisi manajer akan ditampilkan kolom Gajinya.
- **Hasilnya** = Jabatan Manajer yang memiliki nilai maksimum adalah 6250000 kolom hasil **MAX** nya diubah jadi **GajiTerbesar**.
Sedangkan nilai minimumnya adalah 5250000 kolom hasil **MIN** nya diubah jadi **GajiTerkecil**

15

Query

```

SELECT NoCab, MAX(Gaji) AS GajiTerbesar, MIN(Gaji) AS GajiTerkecil
FROM pegawai
GROUP BY NoCab;

```

Hasil

```
MariaDB [company_fadhil]> select NoCab, max(gaji) as gajiterbesar, min(gaji) as gajiterkecil
-> from pegawai
-> group by NoCab;
```

NoCab	gajiterbesar	gajiterkecil
C101	5250000	2500000
C102	5750000	1750000
C103	6250000	2750000
C104	2650000	1725000

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

Analisis

- **SELECT** = Untuk memilih kolom nama mana saja yang dipilih untuk ditampilkan.
- **NoCab** = Nama kolom yang ingin ditampilkan.
- **MAX(Gaji)** = Untuk menampilkan nilai terbesar dari suatu data dalam kolom yang dipilih. Gaji adalah nama kolom yang dipilih.
- **AS GajiTerbesar** = Untuk Menganti nama dari kolom hasil **MAX(Gaji)** menjadi **GajiTerbesar** untuk sementara.
- **MIN(Gaji)** = Untuk menampilkan nilai terkecil dari suatu data dalam kolom yang dipilih. Gaji adalah nama kolom yang dipilih.
- **AS GajiTerkecil** = Untuk Menganti nama dari kolom hasil **MIN(Gaji)** menjadi **GajiTerkecil** untuk sementara.
- **FROM pegawai** = Untuk memilih dari tabel mana yang data kolomnya ingin ditampilkan.
- **GROUP BY** = Untuk mengelompokkan data berdasarkan nilai data yang telah ditentukan pada kolom yang dipilih.
- **NoCab** = Nama kolom yang ingin dikelompokkan.
- **Hasilnya** = Masing-masing **NoCab** dicari nilai maksimum dan minimumnya. Mulai dari C101,C102,,C103,C104 dan nama hasil kolomnya diubah menjadi **GajiTerbesar** dan **GajiTerkecil**.

16

Query

```
SELECT NoCab, MAX(Gaji) AS GajiTerbesar, MIN(Gaji) AS GajiTerkecil
FROM pegawai
```



```
GROUP BY NoCab HAVING COUNT(NIP) >= 3;
```

Hasil

```
MariaDB [company_fadhil]> select NoCab, max(gaji) as gajiterbesar, min(gaji) as gajiterkecil
-> from pegawai
-> group by NoCab having count(nip) >= 3;
+-----+-----+-----+
| NoCab | gajiterbesar | gajiterkecil |
+-----+-----+-----+
| C102  |      5750000 |      1750000 |
+-----+-----+-----+
1 row in set (0.026 sec)
```

Analisis

- **SELECT** = Untuk memilih kolom nama mana saja yang dipilih untuk ditampilkan.
- **NoCab** = Nama kolom yang ingin ditampilkan.
- **MAX(Gaji)** = Untuk menampilkan nilai terbesar dari suatu data dalam kolom yang dipilih. Gaji adalah nama kolom yang dipilih.
- **AS GajiTerbesar** = Untuk Menganti nama dari kolom hasil **MAX(Gaji)** menjadi **GajiTerbesar** untuk sementara.
- **MIN(Gaji)** = Untuk menampilkan nilai terkecil dari suatu data dalam kolom yang dipilih. Gaji adalah nama kolom yang dipilih.
- **AS GajiTerkecil** = Untuk Menganti nama dari kolom hasil **MIN(Gaji)** menjadi **GajiTerkecil** untuk sementara.
- **FROM pegawai** = Untuk memilih dari tabel mana yang data kolomnya ingin ditampilkan.
- **GROUP BY** = Untuk mengelompokkan data berdasarkan nilai data yang telah ditentukan pada kolom yang dipilih.
- **NoCab** = Nama kolom yang ingin dikelompokkan.
- **HAVING** = Kondisi yang harus dipenuhi oleh suatu kelompok data.
- **(COUNT(NIP) >= 3)** = Kondisi dari **HAVING** hanya hasil hitung kolom **NIP** yang lebih dari atau sama dengan 3 yang muncul.
- **Hasilnya** = Seperti no.4, yang mempunyai hasil hitung lebih dari atau sama dengan 3 adalah **NoCab C102** . Jadi hanya itu yang dicari nilai maksimum dan minimumnya pada kolom **Gaji** .

Program

```
Select COUNT(NIP) AS JumlahPegawai,SUM(Gaji) AS totalGaji  
-> AVG (Gaji) AS RataGaji, Max(Gaji), AS GajiMaks, MIN(Gaji), AS GajiMin  
-> From Pegawai
```

Penjelasan

- `Select` = untuk memilih kolom mana saja yang dipilih untuk di tampilkan.
- `COUNT(NIP)` = untuk menghitung jumlah barisan data yang ada pada kolom yang dipilih.
- `AS JumlahPegawai` = untuk mengganti nama kolom hasil `COUNT(NIP)` menjadi Jumlah Pegawai .
- `SUM(Gaji)` = untuk Menjumlah data yang ada pada kolom yang dipilih. `Gaji` adalah kolom yang dipilih.
- `AS TotalGaji` = untuk mengganti nama kolom hasil `sum(gaji)` menjadi total `Gaji` .
- `AVG(Gaji)` = untuk menghitung rata-ratanya suatu data dalam kolom yang dipilih. `Gaji` adalah nama kolom yang dipilih untuk dihitung.
- `AS RataGaji` = untuk mengganti nama kolom hasil `AVG(Gaji)` menjadi `RataGaji` .
- `MAX(Gaji)` = untuk menampilkan nilai terbesar dari suatu data dalam kolom yang dipilih `Gaji` adalah nama kolom yang dipilih.
- `AS Gajimaks` = untuk menampilkan nama dari kolom hasil `MAX(Gaji)` menjadi `Gajimaks` untuk sementara.
- `MIN(Gaji)` = untuk menampilkan nilai terkecil dari suatu kolom yang dipilih `Gaji` nama kolom yang dipilih.
- `AS Gaji min` = untuk mengganti nama dari kolom hasil `min(gaji)` menjadi `Gajimin` untuk sementara.
- `From Pegawai` = untuk memilih tabel mana yang dipilih untuk ditampilkan pegawai adalah nama tabel yang dipilih.
- Hasil Dihitung berapa `NIP` , dijumlahkan semua data pada kolom `Gaji` , Dihitung Rata-Rata dari kolom `Gaji` , Ditampilkan Nilai terbesar pada kolom `Gaji` , dan nilai terkecil dalam kolom `Gaji` .

Hasil

```

MariaDB [company_fadhil]> select count(nip) as jumlahpegawai, sum(gaji) as totalgaji,
-> avg(gaji) as ratagaji, max(gaji) as gajimaks, min(gaji) as gajimin
-> from pegawai;
+-----+-----+-----+-----+-----+
| jumlahpegawai | totalgaji | ratagaji      | gajimaks | gajimin |
+-----+-----+-----+-----+-----+
|              | 30575000 | 3397222.2222  | 6250000  | 1725000  |
+-----+-----+-----+-----+-----+
1 row in set (0.001 sec)

```

18

Program

```

Select COUNT(NIP) AS JumlahPegawai,SUM(Gaji) AS totalGaji
-> ,AVG (Gaji) AS RataGaji, Max(Gaji) AS GajiMaks, MIN(Gaji) AS GajiMin
-> From pegawai
-> WHERE Jabatan = 'Staf' OR Jabatan = 'Sales'
-> GROUP BY NoCab HAVING SUM(Gaji) <= 26000000;

```

Penjelasan

- **Select** = untuk memilih kolom mana saja yang ingin digunakan.
- **COUNT(NIP)** = untuk menghitung barisan data yang ada pada kolom yang dipilih.
- **AS JumlahPegawai** = untuk mengganti nama dari kolom hasil **COUNT(NIP)** menjadi jumlah pegawai untuk sementara
- **SUM(Gaji)** = untuk Menjumlah data yang ada pada kolom yang dipilih. Gaji adalah nama kolom yang dipilih
- **AS TotalGaji** = untuk mengganti nama dari kolom hasil **Sum(Gaji)** menjadi **totalGaji** untuk sementara.
- **AVG(Gaji)** = untuk menghitung Rata-Rata dari kolom yang di pilih. Gaji adalah nama kolom yang di pilih.
- **AS Rata_gaji** = untuk mengganti nama dari kolom **AVG(Gaji)** menjadi **ratagaji** untuk sementara.
- **MAX(Gaji)** = untuk menampilkan nilai terbesar dari suatu data dalam kolom yang dipilih. Gaji adalah nama kolom yang dipilih.
- **AS Gajimaks** = untuk mengganti nama dari kolom **Max(Gaji)** menjadi **Gajimaks** sementara.
- **MIN(Gaji)** = untuk menampilkan nilai minimum dari suatu data dalam kolom yang dipilih. Gaji adalah nama kolom yang dipilih.

- `AS Gajimin` = untuk mengganti nama dari kolom `Min(Gaji)` menjadi `Gajimin` untuk sementara
- `From pegawai` = untuk memilih dari tabel mana yang datanya ingin digunakan Pegawai adalah nama tabel yang dipilih.
- `Where` = kondisi yang harus dipenuhi oleh suatu kolom
- `(jabatan = staf OR jabatan = sales)` = Kondisi dari `where`.
- jadi hanya jabatan staf atau jabatan sales, hanya salah satunya saja yang harus dipenuhi agar bisa tampil. OR (hanya salah satu kondisi yang harus dipenuhi)
- `GRUP BY` = untuk mengelompokkan data sesuai dengan kolom yang ingin di pilih
- `Nocab` = nama kolom yang di pilih untuk dikelompokkan
- `HAVING` = kondisi yang harus dipenuhi oleh suatu kelompok data agar bisa di tampilkan
- `(Sum(Gaji)) <= 2.600.000` = kondisi dari `Having`, hanya data yang hasil jumlah gajinya kurang dari atau sama dengan `2600000` yang bisa tampil
- `Hasilnya` = ada 2 barisan data yang memenuhi kondisi `<= 2600000` Adapun kondisi `where` yang juga dipenuhi oleh barisan data tersebut. Masing-masing namanya diubah sesuai Perintah `AS`.

Hasil

```
MariaDB [company_fadhil]> select count(nip) as jumlahpegawai, sum(gaji) as totalgaji,
-> avg(gaji) as ratagaji, max(gaji) as gajimaks, min(gaji) as gajimin
-> from pegawai
-> where jabatan = 'staf' or jabatan = 'sales'
-> group by NoCab having sum(gaji) <= 2600000;
```

jumlahpegawai	totalgaji	ratagaji	gajimaks	gajimin
1	2500000	2500000.0000	2500000	2500000
1	1725000	1725000.0000	1725000	1725000

2 rows in set (0.002 sec)

Hasil Tantangan

Data Tabel

Table structure

Relation view

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	IdPeg	int(5)			No	None			Change Drop More
2	Nama	varchar(100)	utf8mb4_general_ci		No	None			Change Drop More
3	NamaKel	varchar(100)	utf8mb4_general_ci		No	None			Change Drop More
4	JK	varchar(100)	utf8mb4_general_ci		No	None			Change Drop More
5	Alamat	varchar(100)	utf8mb4_general_ci		No	None			Change Drop More
6	Kota	varchar(100)	utf8mb4_general_ci		No	None			Change Drop More
7	Telp	varchar(100)	utf8mb4_general_ci		No	None			Change Drop More
8	NoCab	varchar(4)	utf8mb4_general_ci		No	None			Change Drop More

Check all

With selected:

Browse

Change

Drop

Primary

Unique

Index

Spatial

Fulltext

Add to central columns

Remove from central columns

Perubahan Data Tabel Pegawai dan Cabang

Pegawai

Before

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

After

			idpeg	nama	namakel	jk	alamat	kota	telp	nocab
<input type="checkbox"/>	Edit	Copy	Delete	10107	Emya	Salsalina	P	Jl.suci 78	Bandung	022-555768 C101
<input type="checkbox"/>	Edit	Copy	Delete	10176	Diah	Wahyuni	P	Jl.Maluku 56	Bandung	022-555934 C101
<input type="checkbox"/>	Edit	Copy	Delete	10246	Dian	Anggraini	P	Jl.Mawar 5	Semarang	024-555102 C103
<input type="checkbox"/>	Edit	Copy	Delete	10252	Cristiano	Messi	L	Jl.A.Yani 15	Makassar	021-555888 C102
<input type="checkbox"/>	Edit	Copy	Delete	10307	Erik	Andrian	L	Jl.Manggis 155	Semarang	024-555236 C103
<input type="checkbox"/>	Edit	Copy	Delete	10314	Ayu	Rahmadani	P	Jl.Malaka 342	Jakarta	021-555098 C102
<input type="checkbox"/>	Edit	Copy	Delete	10324	Martin	Susanto	L	Jl.Bima 51	Jakarta	021-555785 C102
<input type="checkbox"/>	Edit	Copy	Delete	10407	Rio	Gunawan	L	Jl.Melati 356	Surabaya	031-555231 C104
<input type="checkbox"/>	Edit	Copy	Delete	10415	Susan	Sumantri	P	Jl.Pahlawan 24	Surabaya	031-555120 C104

☐ Check all With selected: Edit Copy Delete Export

Cabang

Before

			noCab	nama	alamat	kabupaten	no_telp	kab_kota	email
<input type="checkbox"/>	Edit	Copy	Delete	C101	Fadhil	Jl.Maccini Kidul	Makassar	988727273	faddhhill@gmail.com
<input type="checkbox"/>	Edit	Copy	Delete	C102	Radja	Jl.Maccini Kidul	Makassar	89778667	raddjaa898@gmail.com
<input type="checkbox"/>	Edit	Copy	Delete	C103	Angga	Jl.Malengkeri	Makassar	819183763	Angga@gmail.com
<input type="checkbox"/>	Edit	Copy	Delete	C104	ardi	Jl.Maccini Tengah	Makassar	977837373	ardi@gmail.com

☐ Check all With selected: Edit Copy Delete Export

After

			noCab	nama	alamat	kabupaten	no_telp	kab_kota	email
<input type="checkbox"/>	Edit	Copy	Delete	C101	Fadhil	Jl.Maccini Kidul	Makassar	988727273	faddhhill@gmail.com
<input type="checkbox"/>	Edit	Copy	Delete	C102	Radja	Jl.Maccini Kidul	Makassar	89778667	raddjaa898@gmail.com
<input type="checkbox"/>	Edit	Copy	Delete	C103	Pele	Jl.Cendrawasih	Makassar	819183763	Pele878@gmail.com
<input type="checkbox"/>	Edit	Copy	Delete	C104	ardi	Jl.Maccini Tengah	Makassar	977837373	ardi@gmail.com

☐ Check all With selected: Edit Copy Delete Export

Query Relasi

```

SELECT s.nama, n.nilai
FROM nilai AS n
INNER JOIN siswa AS s ON s.nis = n.nis
WHERE n.nilai > 75;

```

Hasil

Showing rows 0 - 1 (2 total, Query took 0.0008 seconds.)

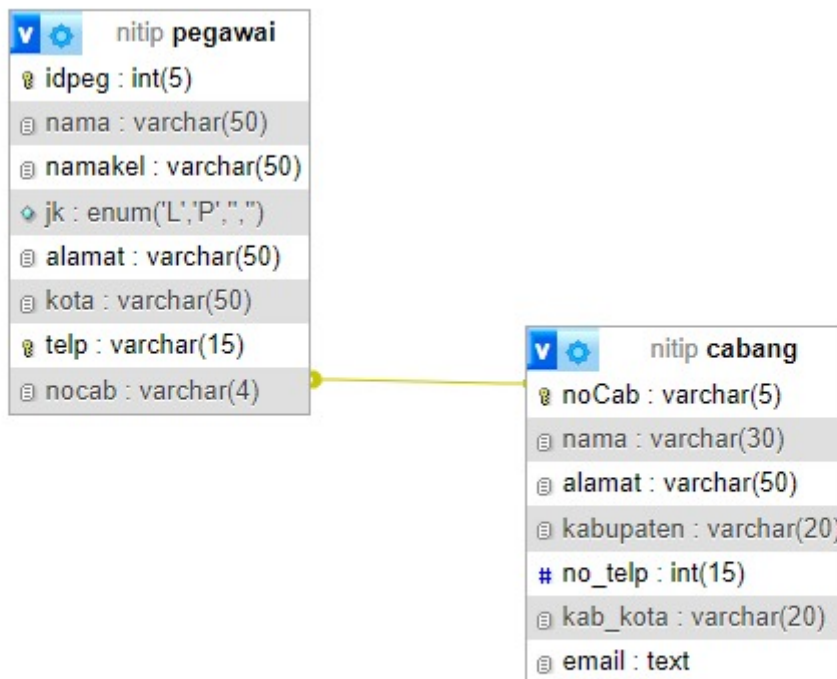
```
SELECT s.nama, n.nilai FROM nilai AS n INNER JOIN siswa AS s ON s.id = n.id_siswa WHERE n.nilai > 75;
```

☐ Profiling [[Edit inline](#)] [[Edit](#)] [[Explain SQL](#)] [[Create PHP code](#)] [[Refresh](#)]

☐ Show all | Number of rows: 25 | Filter rows: Search this table | Sort by key: None

Extra options

nama	nilai
Alkawsar	90
Angga	80



Tugas Basis Data

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

Query

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



```
GROUP BY warna;
```

Analisis

- `SELECT warna, COUNT(*) AS jumlah_mobil` : Memilih kolom warna dan menghitung jumlah mobil (`COUNT(*)`) untuk setiap warna. Hasil perhitungan ini diberi alias `jumlah_mobil`.
- `FROM mobil` : Mengambil data dari tabel mobil.
- `GROUP BY warna` : Mengelompokkan hasil berdasarkan kolom warna.

Kesimpulan

`SELECT warna, COUNT(*) AS jumlah_mobil FROM mobil GROUP BY warna;` digunakan untuk menghitung dan menampilkan jumlah mobil untuk setiap warna, dengan hasil dikelompokkan berdasarkan warna.

Hasil

✓ Showing rows 0 - 2 (3 total, Query took 0.0008 seconds.)

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

☐ Profiling [[Edit inline](#)] [[Edit](#)] [[Explain SQL](#)] [[Create PHP code](#)] [[Refresh](#)]

☐ Show all | Number of rows: 25 ▾ | Filter rows:

Extra options

warna	jumlah_mobil
Hitam	3
Merah	1
Silver	1

berdasarkan query ini tampilkan yang lebih Kecil dari 3 atau sama dengan 3 pemilik mobil nya

Query

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

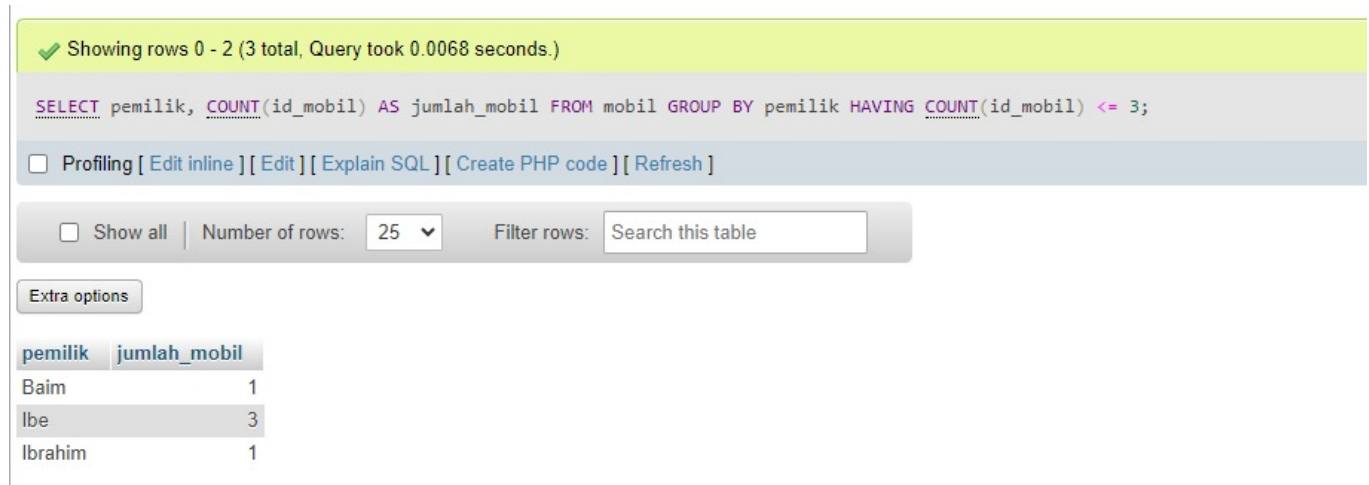
Analisis

- `SELECT pemilik, COUNT(id_mobil) AS jumlah_mobil:`
- `pemilik:` Menampilkan nama pemilik mobil.
- `COUNT(id_mobil) AS jumlah_mobil:` Menghitung jumlah mobil yang dimiliki oleh setiap pemilik, dan menampilkan hasilnya dengan alias `jumlah_mobil`.
- `FROM mobil:` Mengambil data dari tabel mobil.
- `GROUP BY pemilik:` Mengelompokkan data berdasarkan kolom pemilik, sehingga setiap pemilik dikelompokkan bersama dengan jumlah mobil yang mereka miliki.
- `HAVING COUNT(id_mobil) <= 3:` Menyaring hasil kelompok untuk hanya menyertakan pemilik yang memiliki jumlah mobil kurang dari atau sama dengan 3.

Kesimpulan

`SELECT pemilik, COUNT(id_mobil) AS jumlah_mobil FROM mobil GROUP BY pemilik HAVING COUNT(id_mobil) <= 3;` digunakan untuk menghitung dan menampilkan jumlah mobil untuk setiap pemilik yang memiliki maksimal 3 mobil, dengan hasil dikelompokkan berdasarkan pemilik.

Hasil

 Showing rows 0 - 2 (3 total, Query took 0.0068 seconds.)

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

☐ Profiling [\[Edit inline \]](#) [\[Edit \]](#) [\[Explain SQL \]](#) [\[Create PHP code \]](#) [\[Refresh \]](#)

☐ Show all | Number of rows: Filter rows:

Extra options

pemilik	jumlah_mobil
Baim	1
Ibe	3
Ibrahim	1

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

Query

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

Analisis

- `SELECT pemilik, COUNT(*) AS jumlah_mobil`: Memilih kolom `pemilik` dan menghitung jumlah mobil (`COUNT(*)`) untuk setiap pemilik. Hasil perhitungan ini diberi alias `jumlah_mobil`.
- `FROM mobil`: Mengambil data dari tabel `mobil`.
- `GROUP BY pemilik`: Mengelompokkan hasil berdasarkan kolom `pemilik`.
- `HAVING COUNT(*) >= 3`: Menyaring hasil kelompok yang jumlah mobilnya lebih besar atau sama dengan 3.

Kesimpulan

`SELECT pemilik, COUNT(id_mobil) AS jumlah_mobil FROM mobil GROUP BY pemilik HAVING COUNT(id_mobil) >= 3;` digunakan untuk menghitung dan menampilkan jumlah mobil untuk setiap pemilik yang memiliki minimal 3 mobil, dengan hasil dikelompokkan berdasarkan pemilik.

Hasil

Your SQL query has been executed successfully.

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

☐ Profiling [[Edit inline](#)] [[Edit](#)] [[Explain SQL](#)] [[Create PHP code](#)] [[Refresh](#)]

Extra options

pemilik	jumlah_mobil
lbe	3

berdasarkan query yang ada pada praktikum 5 bagian 7 tampilkan data pada table mobil dengan mengelompokkan berdasarkan pemiliknya.hitung menggunakan sum total pendapatan pemilik berdasarkan harga rental

Query

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

Analisis

- `SELECT pemilik, SUM(harga_rental) AS total_pendapatan` : Memilih kolom pemilik dan menghitung total pendapatan (`SUM(harga_rental)`) untuk setiap pemilik. Hasil perhitungan ini diberi alias `total_pendapatan`.
- `FROM mobil` : Mengambil data dari tabel mobil.
- `GROUP BY pemilik` : Mengelompokkan hasil berdasarkan kolom pemilik.

Kesimpulan

`SELECT pemilik, SUM(harga_rental) AS total_pendapatan FROM mobil GROUP BY pemilik;` digunakan untuk menghitung dan menampilkan total pendapatan dari harga rental untuk setiap pemilik, dengan hasil dikelompokkan berdasarkan pemilik.

Hasil

✓ Showing rows 0 - 2 (3 total, Query took 0.0354 seconds.)

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

☐ Profiling [[Edit inline](#)] [[Edit](#)] [[Explain SQL](#)] [[Create PHP code](#)] [[Refresh](#)]

☐ Show all

Number of rows:

25

Filter rows:

Extra options

					pemilik	total_pendapatan		
<input type="checkbox"/>		Edit		Copy		Delete	Baim	50000
<input type="checkbox"/>		Edit		Copy		Delete	Ibe	350000
<input type="checkbox"/>		Edit		Copy		Delete	Ibrahim	50000

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

Query

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

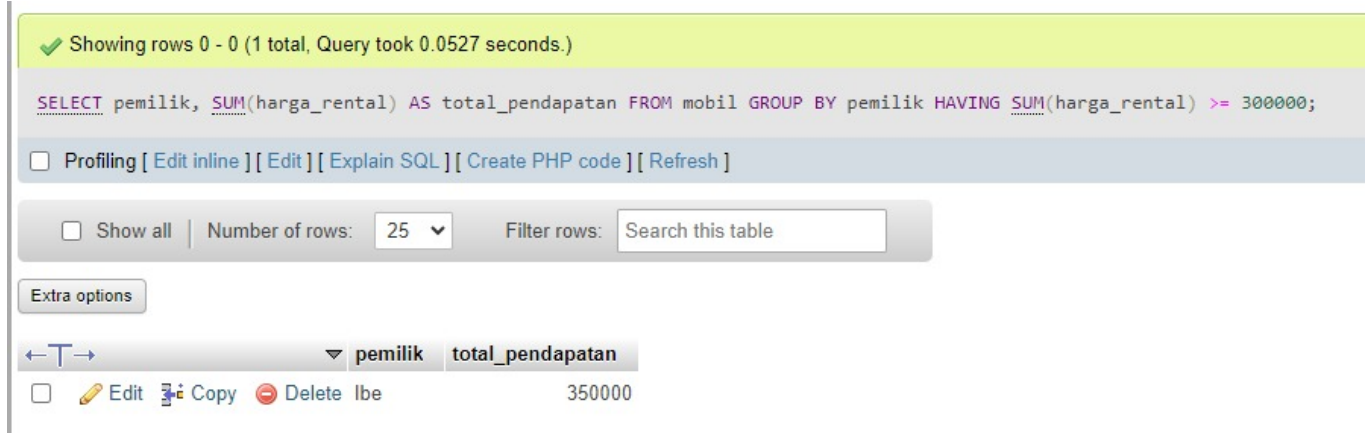
Analisis

- `SELECT pemilik, SUM(harga_rental) AS total_pendapatan` : Memilih kolom pemilik dan menghitung total pendapatan (`SUM(harga_rental)`) untuk setiap pemilik. Hasil perhitungan ini diberi alias `total_pendapatan`.
- `FROM mobil` : Mengambil data dari tabel mobil.
- `GROUP BY pemilik` : Mengelompokkan hasil berdasarkan kolom pemilik.
- `HAVING SUM(harga_rental) >= 300000` : Menyaring hasil kelompok yang total pendapatannya lebih besar atau sama dengan 300.000.

Kesimpulan

`SELECT pemilik, SUM(harga_rental) AS total_pendapatan FROM mobil GROUP BY pemilik HAVING SUM(harga_rental) >= 300000;` digunakan untuk menghitung dan menampilkan total pendapatan dari harga rental untuk setiap pemilik yang memiliki total pendapatan minimal 300000, dengan hasil dikelompokkan berdasarkan pemilik.

Hasil



Showing rows 0 - 0 (1 total, Query took 0.0527 seconds.)

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

☐ Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

☐ Show all | Number of rows: 25 | Filter rows: Search this table

Extra options

	pemilik	total_pendapatan
<input type="checkbox"/> Edit <input type="image"/> Copy <input type="image"/> Delete lbe		350000

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

Query

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

Analisis

- `SELECT pemilik, SUM(harga_rental) AS total_pendapatan`: Memilih kolom `pemilik` dan menghitung total pendapatan (`SUM(harga_rental)`) untuk setiap pemilik. Hasil perhitungan ini diberi alias `total_pendapatan`.
- `FROM mobil`: Mengambil data dari tabel `mobil`.
- `GROUP BY pemilik`: Mengelompokkan hasil berdasarkan kolom `pemilik`.

Kesimpulan

`SELECT pemilik, SUM(harga_rental) AS total_pendapatan FROM mobil GROUP BY pemilik;` digunakan untuk menghitung dan menampilkan total pendapatan dari harga rental untuk setiap pemilik, dengan hasil dikelompokkan berdasarkan pemilik.

Hasil

✓ Showing rows 0 - 2 (3 total, Query took 0.0354 seconds.)

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

☐ Profiling [[Edit inline](#)] [[Edit](#)] [[Explain SQL](#)] [[Create PHP code](#)] [[Refresh](#)]

☐ Show all | Number of rows: 25 ▾ | Filter rows:

Extra options

				pemilik	total_pendapatan
<input type="checkbox"/>	Edit	Copy	Delete	Baim	50000
<input type="checkbox"/>	Edit	Copy	Delete	Ibe	350000
<input type="checkbox"/>	Edit	Copy	Delete	Ibrahim	50000

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

Query

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

```
) AS subquery  
GROUP BY pemilik;
```

Analisis

- `SELECT` : Perintah untuk memilih data dari tabel.
- `pemilik` : Kolom yang ingin ditampilkan.
- `MAX(total_pendapatan) AS pemasukan_terbesar` : Menghitung nilai maksimum dari `total_pendapatan` dan memberikan alias `pemasukan_terbesar`.
- `MIN(total_pendapatan) AS pemasukan_terkecil` : Menghitung nilai minimum dari `total_pendapatan` dan memberikan alias `pemasukan_terkecil`.
- `FROM` : Menunjukkan tabel sumber data.
- `(...) AS subquery` : Subquery yang digunakan untuk menghitung `total_pendapatan` untuk setiap `pemilik` yang memiliki lebih dari satu mobil dan memberikan alias `subquery`.
- `SELECT pemilik, SUM(harga_rental) AS total_pendapatan` : Subquery ini memilih kolom `pemilik` dan menghitung jumlah `harga_rental` sebagai `total_pendapatan`.
- `FROM mobil` : Menentukan tabel `mobil` sebagai sumber data.
- `GROUP BY pemilik` : Mengelompokkan data berdasarkan kolom `pemilik`.
- `HAVING COUNT(id_mobil) > 1` : Kondisi yang memastikan hanya pemilik dengan lebih dari satu mobil yang diikutsertakan.
- `GROUP BY pemilik` : Mengelompokkan hasil akhir berdasarkan kolom `pemilik`.
- `;` : Menandakan akhir dari perintah SQL.

Kesimpulan

`SELECT pemilik, MAX(total_pendapatan) AS pemasukan_terbesar, MIN(total_pendapatan) AS pemasukan_terkecil FROM (SELECT pemilik, SUM(harga_rental) AS total_pendapatan FROM mobil GROUP BY pemilik HAVING COUNT(id_mobil) > 1) AS subquery GROUP BY pemilik;` digunakan untuk menghitung dan menampilkan pemilik, pemasukan terbesar, dan pemasukan terkecil dari pemilik yang memiliki lebih dari satu mobil, dengan hasil dikelompokkan berdasarkan pemilik.

Hasil

✓ Showing rows 0 - 0 (1 total, Query took 0.0460 seconds.)

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

☐ Profiling [\[Edit inline \]](#) [\[Edit \]](#) [\[Explain SQL \]](#) [\[Create PHP code \]](#) [\[Refresh \]](#)

☐ Show all

Number of rows:

25

Filter rows:

Search this table

Extra options

pemilik	pemasukan_terbesar	pemasukan_terkecil
lbe	350000	350000

✓ Showing rows 0 - 0 (1 total, Query took 0.0460 seconds.)

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

[\[Profiling \]](#) [\[Edit inline \]](#) [\[Edit \]](#) [\[Explain SQL \]](#) [\[Create PHP code \]](#) [\[Refresh \]](#)