

NO	NAMA	KEAKTIFAN	PERAN
1	CHAIRIL ABIZALI	3	Membuat PPT dan Mencari Jawaban
2	HANSAR	3	Mencari soal dan sebagai logistik
3	JORDAN	3	Mencari Tema ppt dan sebagai logistik
4	MUH.TAUFIK	3	Membantu mencari soal
5	ABD RAHMAN	3	Mencari jawaban dan Membeli gorengan

## Nomor 1

## Program

```
select count(NIP) as jumlahpegawai,count(jabatan) as jumlahjabatan
from pegawai;
```

## Hasil

```
MariaDB [company_hayril]> select count(NIP) as jumlahpegawai,count(jabatan) as jumlahjabatan from pegawai;
+-----+-----+
| jumlahpegawai | jumlahjabatan |
+-----+-----+
|          9   |          9   |
+-----+-----+
1 row in set (0.004 sec)

MariaDB [company_hayril]> _
```

## Analisis

- ♦ **SELECT** : menentukan kolom yang ingin Anda ambil dari tabel.
- ♦ **COUNT(NIP)** : menghitung berapa banyak baris yang memiliki nilai di kolom **NIP**.
- ♦ **AS jumlahpegawai** : memberi nama hasil hitungan ini sebagai **jumlahpegawai**.
- ♦ **COUNT(jabatan)** : menghitung berapa banyak baris yang memiliki nilai di kolom **jabatan**.
- ♦ **AS jumlahjabatan** : memberi nama hasil hitungan ini sebagai **jumlahjabatan**.
- ♦ **From PEGAWAI** : mengambil data dari tabel **pegawai**.

## Kesimpulan

Query di atas adalah dari program SQL untuk menghitung jumlah pegawai dan jumlah jabatan yang ada dalam tabel pegawai.

## Nomor 2

---

### Program

---

```
select count(nip) as jumlahpegawai  
→ from pegawai  
→ where nocab='c102';
```

### Hasil

---

```
MariaDB [company_hayril]> select count(nip) as jumlahpegawai  
→ from pegawai  
→ where nocab='c102';  
+-----+  
| jumlahpegawai |  
+-----+  
|          3    |  
+-----+  
1 row in set (0.107 sec)  
  
MariaDB [company_hayril]>
```

### Analisis

---

- ♦ **SELECT** : menentukan kolom yang ingin Anda ambil dari tabel.
- ♦ **COUNT(NIP)** : menghitung berapa banyak baris yang memiliki nilai di kolom **NIP**.
- ♦ **AS jumlahpegawai** : memberi nama hasil hitungan ini sebagai **jumlahpegawai**.
- ♦ **From PEGAWAI** : mengambil data dari tabel **pegawai**.
- ♦ **WHERE** : digunakan untuk mencari letak data yang akan di tampilkan.
- ♦ **nocab='C102'** : kondisi data yang hanya akan di tampilkan.

### Kesimpulan

---

Query di atas adalah perintah untuk mengetahui berapa banyak pegawai yang terdaftar dalam kategori atau cabang tertentu, yaitu yang memiliki **nocab** 'c102'.

## Nomor 3

---

### Program

```
select nocab , count(nip) as jumlah_pegawai
from pegawai
group by nocab;
```

## Hasil

```
MariaDB [company_hayril]> 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 (0.098 sec)

MariaDB [company_hayril]> _
```

## Analisis

- ◆ **SELECT** : menentukan kolom yang ingin Anda ambil dari tabel.
- ◆ **COUNT(NIP)** : menghitung berapa banyak baris yang memiliki nilai di kolom **NIP**.
- ◆ **AS jumlahpegawai** : memberi nama hasil hitungan ini sebagai **jumlahpegawai**.
- ◆ **from PEGAWAI** : mengambil data dari tabel **pegawai**.
- ◆ **GROUP BY nocab** : Mengelompokkan baris-baris dalam tabel berdasarkan nilai di kolom **nocab**.

## Kesimpulan

Query di atas adalah perintah yang digunakan untuk menghitung jumlah pegawai (berdasarkan **nip**) yang mengelompokkannya sesuai dengan noCab nya.

## Nomor 4

## Program

```
select nocab, count(nip) as jumlah_pegawai
from pegawai
```

```
group by nocab having count(nip) ≥ 3;
```

## Hasil

```
MariaDB [company_hayril]> 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 (0.017 sec)

MariaDB [company_hayril]>
```

## Analisis

- ◆ **SELECT** : menentukan kolom yang ingin Anda ambil dari tabel.
- ◆ **nocab** : Kolom ini akan ditampilkan dalam hasil query.
- ◆ **COUNT(NIP)** : menghitung berapa banyak baris yang memiliki nilai di kolom **NIP**.
- ◆ **AS jumlahpegawai** : memberi nama hasil hitungan ini sebagai **jumlahpegawai**.
- ◆ **from PEGAWAI** : mengambil data dari tabel **pegawai**.
- ◆ **GROUP BY nocab** : Mengelompokkan baris-baris dalam tabel berdasarkan nilai di kolom **nocab**.
- ◆ **HAVING** : digunakan untuk memfilter grup yang dihasilkan oleh klausa **GROUP BY**.
- ◆ **COUNT(nip) ≥ 3** : memilih jumlah pegawai (**COUNT(nip)**) dan menampilkan nocab yang memiliki 3 data pegawai yang sama.

## Kesimpulan

Query di atas adalah perintah untuk membantu mengetahui berapa banyak pegawai yang ada dalam setiap kelompok **nocab**, tetapi hanya menampilkan kelompok **nocab** yang memiliki 3 atau lebih pegawai.

## Nomor 5

### Program

```
select sum(gaji) as total_gaji
from pegawai;
```

## Hasil

---

```
MariaDB [company_hayril]> select sum(gaji) as total_gaji
-> from pegawai;
+-----+
| total_gaji |
+-----+
| 24950000 |
+-----+
1 row in set (0.099 sec)

MariaDB [company_hayril]> _
```

## Analisis

---

- ◆ **SELECT** : menentukan kolom yang ingin Anda ambil dari tabel.
- ◆ **SUM(gaji)** : Fungsi agregat ini menghitung total atau jumlah seluruh nilai dalam kolom **gaji**.
- ◆ **AS total\_gaji** : ini memberikan nama agar lebih mudah dipahami hasil dari perintah **SUM(gaji)** sebagai **total\_gaji**. Alias digunakan untuk membuat nama kolom hasil lebih mudah dipahami.
- ◆ **from PEGAWAI** : mengambil data dari tabel **pegawai**.

## Kesimpulan

---

Query di atas adalah perintah untuk mengetahui total jumlah gaji keseluruhan dari nilai gaji yang ada di tabel tersebut.

## Nomor 6

---

### Program

---

```
select sum(gaji) as total_gaji
from pegawai
where jabatan = 'manajer';
```

## Hasil

---

```
MariaDB [company_hayril]> select sum(gaji) as total_gaji
-> from pegawai
-> where jabatan = 'manajer';
+-----+
| total_gaji |
+-----+
| 11625000 |
+-----+
```

## Analisis

---

- ♦ **SELECT** : menentukan kolom yang ingin Anda ambil dari tabel.
- ♦ **SUM(gaji)** : Fungsi agregat ini menghitung total atau jumlah seluruh nilai dalam kolom **gaji**.
- ♦ **AS total\_gaji** : ini memberikan nama agar lebih mudah dipahami hasil dari perintah **SUM(gaji)** sebagai **total\_gaji**. Alias digunakan untuk membuat nama kolom hasil lebih mudah dipahami.
- ♦ **from PEGAWAI** : mengambil data dari tabel **pegawai**.
- ♦ **WHERE** : digunakan untuk mencari letak data yang akan di tampilkan.
- ♦ **jabatan='manajer'** : kondisi data yang ingin dijumlahkan.

## Kesimpulan

---

Query di atas adalah perintah yang digunakan untuk menghitung total gaji semua pegawai yang memiliki jabatan 'manajer' dalam tabel **pegawai**.

## Nomor 7

---

## Program

---

```
select nocab,sum(gaji) as totalgaji
from pegawai
group by nocab;
```

## Hasil

---

```

MariaDB [company_hayril]> select nocab,sum(gaji) as totalgaji
-> from pegawai
-> group by nocab;
+-----+-----+
| nocab | totalgaji |
+-----+-----+
| C101  | 7750000  |
| C102  | 9450000  |
| C103  | 3375000  |
| C104  | 4375000  |
+-----+-----+
4 rows in set (0.001 sec)

```

## Analisis

---

- ♦ **SELECT** : menentukan kolom yang ingin Anda ambil dari tabel.
- ♦ **SUM(gaji)** : Fungsi agregat ini menghitung total atau jumlah seluruh nilai dalam kolom **gaji** .
- ♦ **AS total\_gaji** : ini memberikan nama agar lebih mudah dipahami hasil dari perintah **SUM(gaji)** sebagai **total\_gaji** . Alias digunakan untuk membuat nama kolom hasil lebih mudah dipahami.
- ♦ **from PEGAWAI** : mengambil data dari tabel **pegawai** .
- ♦ **group by noCab** : Mengelompokkan baris-baris dalam tabel berdasarkan nilai di kolom **nocab** .

## Kesimpulan

---

Query di atas adalah perintah yang digunakan untuk menghitung total gaji pegawai untuk setiap kelompok **nocab** dalam tabel **pegawai** .

## Nomor 8

---

### Program

---

```

select nocab,sum(gaji) as total_gaji
from pegawai
group by nocab having sum(gaji) ≥ 8000000;

```

## Hasil

---

```

MariaDB [company_hayril]> select nocab,sum(gaji) as total_gaji
-> from pegawai
-> group by nocab having sum(gaji) >=8000000;
+-----+-----+
| nocab | total_gaji |
+-----+-----+
| C102  | 9450000    |
+-----+-----+
1 row in set (0.238 sec)

MariaDB [company_hayril]> _

```

## Analisis

---

- ♦ **SELECT** : menentukan kolom yang ingin Anda ambil dari tabel.
- ♦ **SUM(gaji)** : Fungsi agregat ini menghitung total atau jumlah seluruh nilai dalam kolom **gaji**.
- ♦ **AS total\_gaji** : ini memberikan nama agar lebih mudah dipahami hasil dari perintah **SUM(gaji)** sebagai **total\_gaji**. Alias digunakan untuk membuat nama kolom hasil lebih mudah dipahami.
- ♦ **from PEGAWAI** : mengambil data dari tabel **pegawai**.
- ♦ **HAVING SUM(gaji) ≥ 8000000:** : digunakan untuk memfilter grup yang dihasilkan oleh klausa **GROUP BY** dan menampilkan hanya hasil penjumlahan gaji yang lebih dari 8.000.000 akan di tampilkan.

## Kesimpulan

---

Query di atas adalah perintah yang digunakan untuk menghitung total gaji pegawai untuk setiap kelompok **nocab**, dan hanya menampilkan kelompok **nocab** yang memiliki total gaji 8.000.000 atau lebih.

## Nomor 9

---

### Program

---

```

select avg(gaji) as rata_rata
from pegawai;

```

## Hasil

---



```

MariaDB [company_hayril]> select avg(gaji) as rata_rata
-> from pegawai;
+-----+
| rata_rata |
+-----+
| 2772222.2222 |
+-----+
1 row in set (0.034 sec)

MariaDB [company_hayril]> _

```

## Analisis

---

- ♦ **SELECT** : menentukan kolom yang ingin Anda ambil dari tabel.
- ♦ **AVG(gaji)** : Fungsi agregat ini menghitung nilai rata rata dalam kolom **gaji**.
- ♦ **AS rata\_rata** : ini memberikan nama agar lebih mudah dipahami hasil dari perintah **AVG(gaji)** sebagai **rata\_rata**. Alias digunakan untuk membuat nama kolom hasil lebih mudah dipahami.
- ♦ **from PEGAWAI** : mengambil data dari tabel **pegawai**.

## Kesimpulan

---

Query di atas adalah perintah untuk mengetahui nilai rata rata keseluruhan dari nilai gaji yang ada di tabel tersebut.

## Nomor 10

---

## Program

---

```

SELECT avg(gaji) as gajiRataMgr
from pegawai
where jabatan = 'manajer';

```

## Hasil

---

```

MariaDB [company_hayril]> SELECT avg(gaji) as gajiRataMgr
-> from pegawai
-> where jabatan = 'manajer';
+-----+
| gajiRataMgr |
+-----+
| 3875000.0000 |
+-----+
1 row in set (0.113 sec)

MariaDB [company_hayril]> _

```

## Analisis

---

- ♦ **SELECT** : menentukan kolom yang ingin Anda ambil dari tabel.
- ♦ **AVG(gaji)** : Fungsi agregat ini menghitung jumlah rata rata seluruh nilai dalam kolom **gaji**.
- ♦ **AS GajiRataMgr** : ini memberikan nama agar lebih mudah dipahami hasil dari perintah **AVG(gaji)** sebagai **GajiRataMgr**. Alias digunakan untuk membuat nama kolom hasil lebih mudah dipahami.
- ♦ **from PEGAWAI** : mengambil data dari tabel **pegawai**.
- ♦ **WHERE** : digunakan untuk mencari letak data yang akan di tampilkan.
- ♦ **jabatan='manajer'** : kondisi data yang ingin di cari rata-rata.

## Kesimpulan

---

Query di atas adalah perintah yang digunakan untuk menghitung nilai rata rata gaji semua pegawai yang memiliki jabatan 'manajer' dalam tabel **pegawai**.

## Nomor 11

---

### Program

---

```

select nocab,avg(gaji) as ratagaji
from pegawai
group by nocab;

```

## Hasil

---

```

MariaDB [company_hayril]> select nocab,avg(gaji) as ratagaji
-> from pegawai
-> group by nocab;
+-----+-----+
| nocab | ratagaji |
+-----+-----+
| C101  | 3875000.0000 |
| C102  | 3150000.0000 |
| C103  | 1687500.0000 |
| C104  | 2187500.0000 |
+-----+-----+
4 rows in set (0.107 sec)

MariaDB [company_hayril]>

```

## Analisis

---

- ◆ **SELECT** : menentukan kolom yang ingin Anda ambil dari tabel.
- ◆ **AVG(gaji)** : Fungsi agregat ini menghitung nilai rata rata jumlah seluruh nilai dalam kolom **gaji** .
- ◆ **AS RataGaji** : ini memberikan nama agar lebih mudah dipahami hasil dari perintah **AVG(gaji)** sebagai **RataGaji** . Alias digunakan untuk membuat nama kolom hasil lebih mudah dipahami.
- ◆ **from PEGAWAI** : mengambil data dari tabel **pegawai** .
- ◆ **group by noCab** : Mengelompokkan baris-baris dalam tabel berdasarkan nilai di kolom **nocab** .

## Kesimpulan

---

Query di atas adalah perintah yang digunakan untuk menghitung nilai rata rata gaji pegawai untuk setiap kelompok **nocab** dalam tabel **pegawai** .

## Nomor 12

---

### Program

---

```

SELECT nocab, AVG(gaji) AS rata_gaji
FROM pegawai
GROUP BY nocab
HAVING nocab = 'c101' OR nocab = 'c102';

```

## Hasil

---

```
MariaDB [company_hayril]> SELECT nocab, AVG(gaji) AS rata_gaji FROM pegawai GROUP BY nocab HAVING nocab = 'c101' OR nocab = 'c102';
```

nocab	rata_gaji
C101	3875000.0000
C102	3150000.0000

2 rows in set (0.445 sec)

## Analisis

- ♦ **SELECT** : menentukan kolom yang ingin Anda ambil dari tabel.
- ♦ **COUNT(NIP)** : menghitung berapa banyak baris yang memiliki nilai di kolom **NIP**.
- ♦ **AS jumlahpegawai** : memberi nama hasil hitungan ini sebagai **jumlahpegawai**.
- ♦ **from PEGAWAI** : mengambil data dari tabel **pegawai**.
- ♦ **GROUP BY nocab** : Mengelompokkan baris-baris dalam tabel berdasarkan nilai di kolom **nocab**.
- ♦ **HAVING nocab = 'c101' OR nocab = 'c102'** : Ini menyaring hasil grup yang telah dibentuk untuk hanya menyertakan grup dengan **nocab** yang bernilai 'c101' atau 'c102'. Klausa **HAVING** digunakan setelah pengelompokan (**GROUP BY**) untuk menyaring grup tersebut.

## Kesimpulan

Query di atas adalah perintah yang bertujuan untuk menampilkan rata-rata gaji dari karyawan yang diidentifikasi oleh kode cabang atau departemen (**nocab**) tertentu, yaitu 'c101' dan 'c102'.

## Nomor 13

### Program

```
select max(gaji) as gajiTerbesar , min(gaji) as gajiTerkecil
from pegawai;
```

## Hasil

```
MariaDB [company_hayril]> select max(gaji) as gajiTerbesar , min(gaji) as gajiTerkecil
-> from pegawai;
+-----+-----+
| gajiTerbesar | gajiTerkecil |
+-----+-----+
|      5750000 |      6250000 |
+-----+-----+
1 row in set (0.130 sec)

MariaDB [company_hayril]>
```

## Analisis

---

- ♦ **SELECT** : menentukan kolom yang ingin Anda ambil dari tabel.
- ♦ **max(gaji) AS gajiTerbesar** : Ini menghitung nilai maksimum dari kolom **gaji** dan memberikan nama hasilnya sebagai **gajiTerbesar**.
- ♦ **min(gaji) AS gajiTerkecil** : Ini menghitung nilai minimum dari kolom **gaji** dan memberikan nama hasilnya sebagai **gajiTerkecil**.
- ♦ **from PEGAWAI** : mengambil data dari tabel **pegawai**.

## Kesimpulan

---

Query di atas adalah perintah memilih dua nilai yang dihitung, yaitu nilai maksimum **gaji** (diberi nama **gajiTerbesar**) dan nilai minimum **gaji** (diberi nama **gajiTerkecil**).

## Nomor 14

---

## Program

---

```
select max(gaji) as gajiTerbesar , min(gaji) as gajiterkecil
from pegawai
where jabatan = 'manajer';
```

## Hasil

---

```

MariaDB [company_hayril]> select max(gaji) as gajiTerbesar , min(gaji) as gajiterkecil
-> from pegawai
-> where jabatan = 'manajer';
+-----+-----+
| gajiTerbesar | gajiterkecil |
+-----+-----+
|      5750000 |       625000 |
+-----+-----+
1 row in set (0.105 sec)

MariaDB [company_hayril]>

```

## Analisis

- ♦ **SELECT** : menentukan kolom yang ingin Anda ambil dari tabel.
- ♦ **max(gaji) AS gajiTerbesar** : Ini menghitung nilai maksimum dari kolom **gaji** dan memberikan nama hasilnya sebagai **gajiTerbesar**.
- ♦ **min(gaji) AS gajiterkecil** : Ini menghitung nilai minimum dari kolom **gaji** dan memberikan nama hasilnya sebagai **gajiterkecil**.
- ♦ **from PEGAWAI** : mengambil data dari tabel **pegawai**.
- ♦ **WHERE** : digunakan untuk mencari letak data yang akan di tampilkan.
- ♦ **jabatan='manajer'** : kondisi data yang ingin di cari jumlah maksimal dan minimal nya.

## Kesimpulan

Query di atas adalah perintah yang bertujuan untuk menampilkan gaji terbesar dan gaji terkecil dari semua karyawan dengan jabatan 'manajer' yang ada di tabel pegawai.

## Nomor 15

### Program

```

select nocab,max(gaji) as gajiTerbesar , min(gaji) as
gajiterkecil
→ from pegawai
→ group by nocab;

```

## Hasil

```

MariaDB [company_hayril]> 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	2750000	625000
C104	2650000	1725000

```

4 rows in set (0.001 sec)

```

## Analisis

- ♦ **SELECT** : menentukan kolom yang ingin Anda ambil dari tabel.
- ♦ **max(gaji) AS gajiTerbesar** : Ini menghitung nilai maksimum dari kolom **gaji** dan memberikan nama hasilnya sebagai **gajiTerbesar**.
- ♦ **min(gaji) AS gajiTerkecil** : Ini menghitung nilai minimum dari kolom **gaji** dan memberikan nama hasilnya sebagai **gajiTerkecil**.
- ♦ **from PEGAWAI** : mengambil data dari tabel **pegawai**.
- ♦ **group by noCab** : Mengelompokkan baris-baris dalam tabel berdasarkan nilai di kolom **nocab**.

## Kesimpulan

Query di atas adalah perintah yang bertujuan untuk menampilkan gaji terbesar dan gaji terkecil dari karyawan yang dikelompokkan berdasarkan **nocab**, sehingga Anda dapat melihat nilai tertinggi dan terendah dari gaji dalam setiap cabang atau departemen yang diidentifikasi oleh **nocab**.

## Nomor 16

### Program

```

select nocab,max(gaji) as gajiTerbesar , min(gaji) as gajiTerkecil
from pegawai
group by nocab having count(nip) ≥ 3;

```

## Hasil

```
MariaDB [company_hayril]> 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 (0.109 sec)

MariaDB [company_hayril]>
```

## Analisis

- ◆ **SELECT** : menentukan kolom yang ingin Anda ambil dari tabel.
- ◆ **nocab** : Kolom ini dipilih langsung dari tabel **pegawai**.
- ◆ **max(gaji) AS gajiTerbesar** : Menghitung nilai maksimum dari kolom **gaji** dalam setiap grup **nocab** dan memberikan nama hasilnya sebagai **gajiTerbesar**.
- ◆ **min(gaji) AS gajiTerkecil** : Menghitung nilai minimum dari kolom **gaji** dalam setiap grup **nocab** dan memberikan nama hasilnya sebagai **gajiTerkecil**.
- ◆ **from PEGAWAI** : mengambil data dari tabel **pegawai**.
- ◆ **HAVING count(nip) ≥ 3** : Menyaring grup yang telah dikelompokkan untuk hanya menyertakan grup dengan jumlah karyawan (dihitung dengan **count(nip)**) sebanyak 3 atau lebih.

## Kesimpulan

Query di atas adalah perintah yang bertujuan untuk menampilkan gaji terbesar dan gaji terkecil dari setiap cabang atau departemen (**nocab**) yang memiliki setidaknya 3 karyawan.

## Nomor 17

### Program

```
select count(nip) as jumlahPegawai , sum(gaji) as Totalgaji,
AVG(gaji) as ratagaji , max(gaji) as gajiMaks , Min(gaji) as gajiMin
from pegawai;
```

## Hasil



```

MariaDB [company_hayril]> 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 |
+-----+-----+-----+-----+-----+
| 9 | 24950000 | 2772222.2222 | 5750000 | 625000 |
+-----+-----+-----+-----+-----+
1 row in set (0.001 sec)

MariaDB [company_hayril]>

```

## Analisis

- ♦ **SELECT** : menentukan kolom yang ingin Anda ambil dari tabel.
- ♦ **count(nip) AS jumlahPegawai** : Menghitung jumlah total baris (karyawan) berdasarkan kolom **nip** dan memberi nama hasilnya sebagai **jumlahPegawai**.
- ♦ **sum(gaji) AS Totalgaji** : Menghitung jumlah total dari kolom **gaji** dan memberi nama hasilnya sebagai **Totalgaji**.
- ♦ **AVG(gaji) AS ratagaji** : Menghitung nilai rata-rata dari kolom **gaji** dan memberi nama hasilnya sebagai **ratagaji**.
- ♦ **max(gaji) AS gajiMaks** : Menghitung nilai maksimum dari kolom **gaji** dan memberi nama hasilnya sebagai **gajiMaks**.
- ♦ **min(gaji) AS gajiMin** : Menghitung nilai minimum dari kolom **gaji** dan memberi nama hasilnya sebagai **gajiMin**.
- ♦ **from PEGAWAI** : mengambil data dari tabel **pegawai**.

## Kesimpulan

Query di atas adalah perintah yang bertujuan untuk memberikan gambaran umum tentang data gaji di tabel **pegawai**, dengan menghitung jumlah karyawan, total gaji, rata-rata gaji, gaji maksimum, dan gaji minimum.

## Nomor 18

### Program

```

select count(nip) as jumlahPegawai ,Sum(gaji) as totalGaji,
Avg(gaji) as ratagaji , Max(gaji) as gajiMaks ,Min(Gaji) as MinGaji
from pegawai
where jabatan = 'staf' or jabatan = 'sales'
group by nocab having sum(gaji) ≤ 2600000;

```