

1. Gunakan perintah DESC pegawai; untuk mendapatkan struktur tabel.

STRUKTUR "

SQL

```
CREATE TABLE pegawai (  
-> NIP INT PRIMARY KEY,  
-> NDep VARCHAR(255) NOT NULL,  
-> NBlk VARCHAR(255),  
-> JK ENUM('L', 'P') NOT NULL,  
-> Alamat TEXT NOT NULL,  
-> Telp VARCHAR(255) NOT NULL,  
Jabatan ENUM('Manager', 'Sales', 'Staff'),  
    Gaji BIGINT NOT NULL,  
    NoCab VARCHAR(255) NOT NULL  
);
```

PENJELASAN :

- **NIP INT PRIMARY KEY:**
 - **NIP:** Kolom ini menyimpan Nomor Induk Pegawai yang bertipe data integer (INT).
 - **PRIMARY KEY:** Menandakan bahwa kolom NIP adalah kunci utama dari tabel ini. Kunci utama harus memiliki nilai yang unik untuk setiap baris dan tidak boleh NULL.
- **NDep VARCHAR(255) NOT NULL:**
 - **NDep:** Kolom ini menyimpan Nama Depan pegawai bertipe data string dengan panjang maksimum 255 karakter (VARCHAR(255)).
 - **NOT NULL:** Menandakan bahwa kolom ini tidak boleh berisi nilai kosong (NULL); artinya, setiap baris harus memiliki nilai untuk kolom ini.
- **NBlk VARCHAR(255):**
 - **NBlk:** Kolom ini menyimpan Nama Belakang pegawai bertipe data string dengan panjang maksimum 255 karakter (VARCHAR(255)).
 - Kolom ini tidak memiliki batasan NOT NULL, sehingga nilai kosong (NULL) diizinkan.
- **JK ENUM('L', 'P') NOT NULL:**
 - **JK:** Kolom ini menyimpan Jenis Kelamin pegawai dengan tipe data ENUM. Hanya dua nilai yang diperbolehkan: 'L' (Laki-laki) atau 'P' (Perempuan).
 - **NOT NULL:** Menandakan bahwa kolom ini tidak boleh kosong; artinya, setiap baris harus memiliki nilai untuk kolom ini.

- *Alamat TEXT NOT NULL:*
 - *Alamat:* Kolom ini menyimpan alamat pegawai dengan tipe data TEXT, yang bisa menampung teks dalam jumlah besar.
 - *NOT NULL:* Menandakan bahwa kolom ini harus diisi dengan nilai; nilai kosong (NULL) tidak diperbolehkan.
- *Telp VARCHAR(255) NOT NULL:*
 - *Telp:* Kolom ini menyimpan nomor telepon pegawai dengan tipe data string dan panjang maksimum 255 karakter (VARCHAR(255)).
 - *NOT NULL:* Menandakan bahwa kolom ini harus diisi; nilai kosong (NULL) tidak diperbolehkan.
- *Jabatan ENUM('Manager', 'Sales', 'Staff'):*
 - *Jabatan:* Kolom ini menyimpan jabatan pegawai dengan tipe data ENUM. Nilai yang diperbolehkan adalah 'Manager', 'Supervisor', atau 'Staff'.
 - Kolom ini tidak memiliki batasan NOT NULL, sehingga nilai kosong (NULL) diizinkan.
- *Gaji BIGINT NOT NULL:*
 - *Gaji:* Kolom ini menyimpan gaji pegawai dengan tipe data BIGINT, yang dapat menampung bilangan bulat yang sangat besar.
 - *NOT NULL:* Menandakan bahwa kolom ini harus diisi; nilai kosong (NULL) tidak diperbolehkan.
- *NoCab VARCHAR(255) NOT NULL:*
 - *NoCab:* Kolom ini menyimpan nomor cabang pegawai dengan tipe data string dan panjang maksimum 255 karakter (VARCHAR(255)).
 - *NOT NULL:* Menandakan bahwa kolom ini harus diisi; nilai kosong (NULL) tidak diperbolehkan.

HASIL :

"Asset/gambar2.jpg" could not be found.

2. Gunakan perintah **SELECT * FROM pegawai;** untuk mendapatkan data.

STRUKTUR "

SQL

```
INSERT INTO pegawai (NIP, NDep, NBlk, JK, Alamat, Telp, Jabatan, Gaji, NoCab) VALUES
-> (10107, 'Emya', 'Salsalina', 'P', 'JL. Suci 78 Bandung', '022-555768', 'Manager', 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', 'Staff', 1750000, 'C102'),
-> (10252, 'Antoni', 'Irawan', 'L', 'JL. A. Yani 15 Jakarta', '021-555888', 'Manager', 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', 'Adrian', 'L', 'JL. Manggis 5 Semarang', '024-555236', 'Manager', 6250000, 'C103'),
-> (10415, 'Susan', 'Sumantri', 'P', 'JL. Pahlawan 24 Surabaya', '031-555120', '', 2650000, 'C104'),
-> (10407, 'Rio', 'Gunawan', 'L', 'JL. Melati 356 Surabaya', '031-555231', 'Staff', 1725000, 'C104');

```

PENJELASAN :

- *INSERT INTO pegawai:*
 - Menunjukkan bahwa Anda akan menambahkan data ke tabel bernama pegawai.
- *(NIP, NDep, NBlk, JK, Alamat, Telp, Jabatan, Gaji, NoCab):*
 - Ini adalah daftar kolom dalam tabel pegawai yang akan diisi dengan data. Kolom-kolom ini adalah:
 - NIP (Nomor Induk Pegawai)
 - NDep (Nama Depan)
 - NBlk (Nama Belakang)
 - JK (Jenis Kelamin)
 - Alamat (Alamat)
 - Telp (Telepon)
 - Jabatan (Jabatan)
 - Gaji (Gaji)
 - NoCab (Nomor Cabang)
- *VALUES:*
 - Menunjukkan data yang akan dimasukkan ke dalam tabel. Data untuk setiap baris harus sesuai dengan urutan kolom yang disebutkan sebelumnya.
- *Data yang Dimasukkan:*
 - Baris pertama:
 - NIP: 10107
 - NDep: 'Emya'
 - NBlk: 'Salsalina'
 - JK: 'P' (Perempuan)

- Alamat: 'JL. Suci 78 Bandung'
- Telp: '022-555768'
- Jabatan: 'Manager'
- Gaji: 5250000
- NoCab: 'C101'
- Baris kedua dan seterusnya mengikuti pola yang sama, dengan data yang berbeda.
- *Catatan:*
 - Pada baris 10415 untuk Susan Sumantri, kolom Jabatan tidak diisi ("), yang bisa menyebabkan masalah jika kolom Jabatan adalah ENUM dan tidak termasuk nilai kosong. Pastikan kolom Jabatan memiliki nilai yang valid.
 - Jika ada nilai kosong (") dalam kolom Jabatan, Anda mungkin ingin memperbarui baris ini dengan jabatan yang sesuai, seperti 'Staff', atau menyesuaikan schema tabel untuk mengizinkan nilai kosong jika perlu.

HASIL :

"Asset/gambar1.jpg" could not be found.

PRAKTIKKUM 5

GAMBAR :

```
MariaDB [company_ariel]> SELECT
-> COUNT(NIP) AS JumlahPegawai,
-> COUNT(Jabatan) AS JumlahJabatan
-> FROM pegawai;
```

JumlahPegawai	JumlahJabatan
9	9

1 row in set (0.001 sec)

STRIKTUR :

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

PENJELASAN :

- `SELECT` = untuk memilih kolom apa saja yang ingin dipilih (untuk dihitung).
- `COUNT (NIP)` = untuk menghitung Jumlah barisan data yang mempunyai dari kolom Yang dipilih. NIP adalah nama kolom Yang dipilih untuk dihitung.

- `AS` = untuk mengubah nama dari suatu kolom untuk sementara. Jumlah Pegawai = 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 dan kolom yang dipilih. Jabatan adalah nama kolom Yang dipilih untuk dihitung
- `AS` = untuk mengubah nama dari suatu kolom untuk sementara. Jumlah Jabatan = merupakan nama sementara dari perintah `COUNT (Jabatan)`.
- `FROM Pegawai` = merupakan dari tabel mana datanya yang digunakan Pegawai 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 Jumlah pesawai. Kolom Jabatan Jusa dihitung, akan tetapi ada satu data yang berisi Null (kosong), oleh karena itu hanya ada 8 data ditampilkan sebagai Jumlah Jabatan.

Latihan-2

GAMBAR :

```

MariaDB [company_ariel]> select count(NIP) AS JumlahPegawai
-> FROM pegawai
-> WHERE NoCab = 'C102';
+-----+
| JumlahPegawai |
+-----+
|              3 |
+-----+
1 row in set (0.001 sec)

```

STRIKTUR :

```

select count(NIP) AS JumlahPegawai
-> FROM pegawai
-> WHERE NoCab = 'C102';

```

PENJELASAN :

- `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 untuk sementara.
- `Jumlah Pesawai` = 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 dengan query `COUNT (NIP)`.
- `(Nocab = 'C102')` = adalah kondisi dari `WHERE` Yang harus dipenuhi, Jadi hanya barisan data yang memiliki Clo2 di kolom "vocab Yang bisa dihitung.
- **Hasilnya** = Di 9 barisan data Yangaja Pada tabel Pegawai, kita ingin menghitung Jumlah barisan data yang memiliki nilai 'clo2 Pada kolom "Nocab" nya dengan menggunakan `COUNT`. Jadi Yang muncul adalah 3 0 barisan data. kita Juga ingin mengubah nama dari kolom hasil Perintah `COUNT` secara sementara dengan Perintah `AS`, namanya adalah Jumlah Pegawai.

Latihan-3

GAMBAR :

```
MariaDB [company_ariel]> SELECT SUM(Gaji) AS Gaji_Manager
-> FROM pegawai
-> WHERE Jabatan = 'Manager';
+-----+
| Gaji_Manager |
+-----+
|      17250000 |
+-----+
1 row in set (0.001 sec)
```

STRIKTUR :

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

PENJELASAN :

- `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` = merupakan nama sementara dari kolom hasil `COUNT (NIP)`.
- `FROM Pegawai` dari tabel mana Yang data kolomnya 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 hama kolom Yang dipilih untuk datanya dikelompokkan.
- Hasilnya = Berdasarkan. 9 barisan data, masing-masing. nilai, dalam kolom Nocab

Latihan-4

GAMBAR :

```
MariaDB [company_ariel]> 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 (0.001 sec)

STRIKTUR :

```
SELECT NoCab, COUNT(NIP) AS Jumlah_pegawai
-> FROM pegawai
-> GROUP BY NoCab;
```

PENJELASAN :

- **SELECT** = untuk memilih Kolom mana saja yang dipilih untuk ditampilkan/dijumlahkan. Nocab = adalah nama kolom yang ingin 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 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 kolom yang dipilih.
- **Nocab** = nama kolom Yang datanya dipilih untuk dikelompokkan.
- **Hasilnya** = Jadi, berdasarkan kolom Nocab, barisan data yang kolom Nocab nya berisi clol maka kolom Gaji dari barisan data itu dijumlahkan bersama barisan data Yang memiliki Nocab clol dua. Maka kolom Gaji dijumlahkan sesuai dengan kolom Nocab nya masing-masing, mulai dari c101 memiliki 2 kolom Gaji Yang bisa dijumlahkan. Sama dengan

c103 dan c104. Adapun c102 memiliki 3 kolom Gaji yang dapat dijumlahkan. Total Gaji merupakan hasil Perintah dari AS untuk mengubah nama kolom hasil dari Sum(Gaji).

Latihan-5

GAMBAR :

```
MariaDB [company_ariel]> 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 (0.001 sec)
```

STRIKTUR :

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

PENJELASAN :

- **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 Yang data kolomnya 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 dikelompokkan datanya.
- **HAVING** = untuk menentukan kondisi (Yang harus dipenuhi) oleh suatu kelompok data agar bisa ditampilkan.
- **(COUNT (NIP) >= 3)** = merupakan kondisi Yang 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 Yang ingin ditampilkan adalah hasil hitungan yang lebih dari atau sama dengan 3. Yaitu Nocab C102 Yang ada 3. Yang lain c101 ada 2, c103 ada 2, c104 ada 2.

Latihan-6

GAMBAR :

```
MariaDB [company_ariel]> SELECT SUM(Gaji) AS Total_Gaji
-> FROM pegawai;
+-----+
| Total_Gaji |
+-----+
| 30575000 |
+-----+
1 row in set (0.000 sec)
```

STRIKTUR : **

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

PENJELASAN :

- **SELECT** = untuk memilih kolom mana saja yang dipilih untuk dijumlahkan.
- **SUM (Gaji)** untuk menghitung Jumlah data (khusus angka) Pada kolom Yang 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 tabel mana Yang kolom datanya akan digunakan. Pegawai adalah nama dari tabel Yang dipilih.
- **Hasilnya** = kolom gaji Yang isi datanya berupa angka-angka, semuanya dijumlahkan menjadi satu seperti ditotalkan (Sama seperti matematika Pada umumnya). hasilnya adalah 30 575 000. Adapun nama kolom dari hasil Jumlah tersebut diubah dari SUM(Gaji) menjadi Total-gaji.

Latihan-7

GAMBAR :

```
MariaDB [company_ariel]> SELECT SUM(Gaji) AS Gaji_Manager
-> FROM pegawai
-> WHERE Jabatan = 'Manager';
+-----+
| Gaji_Manager |
+-----+
| 17250000 |
+-----+
1 row in set (0.001 sec)
```

STRIKTUR : **

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

PENJELASAN :

- **SELECT** = untuk memilih kolom mana saja yang dipilih untuk dijumlahkan.
- **Sum (Gaji)** = untuk menghitung Jumlah isi 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) secara sementara.
- **Gaji_Manager** = merupakan nama Sementara dari Perintah AS.
- **FROM Pegawai** = untuk memilih dari tabel mana Yang kolom datanya akan digunakan. Pegawai adalah nama dari table yang dipilih.
- **WHERE** = kondisi Yang harus dipenuhi oleh suatu kolom agar datanya bisa dijumlah.
- **(Jabatan = "manager")** = merupakan kondisi dari WHERE. Hanya barisan data yang kolom Jabatannya berisi kolom Gajinya bisa diJumlahkan.
- **Hasilnya** = barisan data Yang kolom Jabatannya berisi manajer akan dijumlah kolom Gajinya menjadi. 17250 000. Jadi hanya beberapa kolom. Saja yang dijumlah

Latihan-8

GAMBAR :

```
MariaDB [company_ariel]> SELECT NoCab, SUM(Gaji) AS TotalGaji  
-> FROM pegawai  
-> GROUP BY NoCab;  
+-----+-----+  
| NoCab | TotalGaji |  
+-----+-----+  
| C101 | 7750000 |  
| C102 | 9450000 |  
| C103 | 9000000 |  
| C104 | 4375000 |  
+-----+-----+  
4 rows in set (0.001 sec)
```

STRIKTUR : **

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

PENJELASAN :

- `SELECT` = untuk memilih Kolom mana saja yang dipilih untuk ditampilkan/dijumlahkan.
- `Nocab` = adalah nama kolom yang ingin 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 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 kolom yang dipilih.
- `Nocab` = nama kolom Yang datanya dipilih untuk dikelompokkan.
- **Hasilnya** = Jadi, berdasarkan kolom `Nocab`, barisan data yang kolom `Nocab` nya berisi c101 maka kolom `Gaji` dari barisan data itu dijumlahkan bersama barisan data Yang memiliki `Nocab` c101 dua. Maka kolom `Gaji` dijumlahkan sesuai dengan kolom `Nocab` nya masing-masing, mulai dari c101 memiliki 2 kolom `Gaji` Yang bisa dijumlahkan. Sama dengan c103 dan c104. Adapun c102 memiliki 3 kolom `Gaji` yang dapat dijumlahkan. Total `Gaji` merupakan hasil Perintah dari `AS` untuk mengubah nama kolom hasil dari `Sum(Gaji)`.

Latihan-9

GAMBAR :

```
MariaDB [company_ariel]> 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 (0.002 sec)
```

STRIKTUR : **

```
SELECT NoCab, SUM(Gaji) AS Total_Gaji
-> FROM pegawai
-> GROUP BY NoCab HAVING SUM(Gaji) >= 8000000;
```

PENJELASAN :

- `SELECT` = untuk memilih kolom mana saja yang dipilih untuk ditampilkan / dijumlahkan. 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 kolom hasil Sum (Gaji) untuk sementara.
- `Total_Gaji` = 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 agar bisa ditampilkan.
- `(SUM (Gaji) >= 8000000)` = Kondisi dari HAVING, Hasil dari Penjumlahan Gaji Yang hanya bisa ditampilkan adalah Hasil yang lebih dari atau sama dengan 8000000.
- **Hasilnya** = Sama seperti sebelumnya, tetapi No cab Yang memenuhi kondisi tersebut hanya clo2 dan c103 karena hasil Jumlah kolom Gaji nya lebih dari atau sama dengan 8000000. Adapun hasil kolom `SUM(Gaji)` diganti Jadi `Total_Gaji`.

Latihan-10

GAMBAR :

```
MariaDB [company_ariel]> SELECT AVG(Gaji) AS Rata_rata
-> FROM pegawai;
+-----+
| Rata_rata |
+-----+
| 3397222.2222 |
+-----+
1 row in set (0.000 sec)
```

STRIKTUR : **

```
SELECT AVG(Gaji) AS Rata_rata
-> FROM pegawai;
```

PENJELASAN :

- `SELECT` = untuk memilih kolom mana Sara 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 (Gari)` 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.

Latihan-11

GAMBAR :

```
MariaDB [company_ariel]> SELECT AVG(Gaji) AS GajiRataMgr
-> FROM pegawai
-> WHERE Jabatan = 'Manager';
+-----+
| GajiRataMgr |
+-----+
| 5750000.0000 |
+-----+
1 row in set (0.001 sec)
```

STRIKTUR : **

```
SELECT AVG(Gaji) AS GajiRataMgr
-> FROM pegawai
-> WHERE Jabatan = 'Manager';
```

PENJELASAN :

- **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.
- **Gaji Ratamgr** = nama sementara dari Perintah AS.
- **FROM Pegawai** = untuk memilih dan tabel mana Yang data kolomnya ingin digunakan Pegawai adalah nama dari tabel Yang dipilih.
- **WHERE** = Kondisi Yang harus dipenuhi oleh suatu kolom agar datanya bisa dihitung rataratanya (Jabatan = 'Manajer') kondisi dari **WHERE** . Barisan data yang kolom Jabatannya Manajer akan dihitung rata-rata kolom Gajinya.
- **Hasilnya** = 5750000.0000 merupakan hasil hitung rata-rata dari barisan data yang memiliki manajer di kolom Jabatan nya, dari situ kolom Gaji nya di hitung.

Latiahn-12

GAMBAR :

```
MariaDB [company_ariel]> 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 (0.001 sec)

STRIKTUR : **

```
SELECT NoCab, AVG(Gaji) AS RataGaji
-> FROM pegawai
-> GROUP BY NoCab;
```

PENJELASAN :

- **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(Gajii) untuk sementara.
- **Rata Gaji** = adalah nama sementara dari Perintah As.
- **FROM Pegawai** = untuk memilih dan 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 Yong dipilih.
- **Nocab** = nama kolom Yang dipilih untuk datanya dikelompokkan.
- **Hasilnya** = Hampir sama seperti no.7, masing-masing kolom Nocab dihitung rata-ratanya

Latiahn-13

GAMBAR :

```

MariaDB [company_ariel]> 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 (0.001 sec)

```

STRIKTUR : **

```

SELECT NoCab, AVG(Gaji) AS RataGaji
-> FROM pegawai
-> GROUP BY NoCab HAVING NoCab = 'C101' OR NoCab = 'C102';

```

PENJELASAN :

- **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.
- **rata Gaji** = nama sementara dari perintah AS.
- **FROM Pegawai** untuk memilih dari tabel mana yang datanya 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.
- **(Nocab101' 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.

Latiahn-14

GAMBAR :


```
MariaDB [company_ariel]> SELECT MAX(Gaji) AS GajiTerbesar, MIN(Gaji) AS GajiTerkecil
-> FROM pegawai;
+-----+-----+
| GajiTerbesar | GajiTerkecil |
+-----+-----+
|      6250000 |      1725000 |
+-----+-----+
1 row in set (0.003 sec)
```

STRIKTUR : **

```
SELECT MAX(Gaji) AS GajiTerbesar, MIN(Gaji) AS GajiTerkecil
-> FROM pegawai;
```

PENJELASAN :

- `SELECT` = untuk memilih kolom mana saja yang dipilih untuk ditampilkan.
- `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 Min(Gaji), menjadi nama Sementara Yaitu Gaji Terbesar.
- `MIN (Gaji)` = untuk menampilkan nilai minimum atau terkecil/terendah dari suatu data dalam kolom yg dipilih. Gaji adalah nama kolom Yang dipilih.
- `As Gaji Terkecil` = untuk mengganti nama dari kolom hasil MIN (Gaji) menjadi Gaji Terkecil untuk sementara.
- `FROM Pegawai` = untuk memilih dari tabel mana yang dat kolomnya ingin ditampilkan.
- **Hasilnya** = Jadi dari 9 nilai yang ada di kolom Gaji, Gaji maksimumnya adalah 6250000 dan namanya diubah menjadi Gajiterbesar. Gajii minimumnya adalah 1725000 dan namanya diubah menadi Gajiterkecil.

Latiahn-15

GAMBAR :

```
MariaDB [company_ariel]> 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)
```


STRIKTUR : **

```
SELECT NoCab, Max(Gaji) AS GajiTerbesar, MIN(Gaji) AS GajiTerkecil  
-> FROM pegawai  
-> GROUP BY NoCab;
```

PENJELASAN :

- `SELECT` = untuk memilih kolom mana saja yang dipilih untuk ditampilkan.
- `Nocab` = nama kolom Youd ingin ditampilkan.
- `MAX (Gaji)` = untuk menampilkan nilai terbesar dari suatu data dalam kolom Yong dish Gari nama kolom Yang dipilih.
- `AS GajiTerbesar` = untuk mengganti nama kolom hasil `MAX (Gaji)` menjadi Gaji terbesar untuk sementara.
- `MIN (Gaji)` = untuk menampilkan nilai terkecil dari suatu data dalam kolom Yang dipilih Gaji nama kolom Yang dipilih.
- `As Gajiterkecil` = untuk mengganti nama kolom hasil `MIN (Gaji)` menjadi Gaji Terkecil untuk sementara.
- `FROM Pegawai` = untuk memilih dari tabel mana Yang data kolomnya ingin ditampilkan. Pegawai adalah nama tabel Yang dipilih untuk ditampilkan.
- `GROUP BY` = untuk mengelompokkan data berdasarkan nilai data yang telah ditentukan Pada kolom Yan dipilih.
- `Nocab` = nama Kolom yang ingin dikelompokkan.
- **Hasilnya** = masing-masing Nocab dicari nilai maksimum dan minimumnya. mulai dari clol, c102, c103, c104. dan nama hasil kolannya diubah Jadi Gajiterbesar dan GajiTerkecil.

Latiahn-16

GAMBAR :

```
MariaDB [company_ariel]> 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 (0.001 sec)
```

STRIKTUR : **

```
SELECT NoCab, Max(Gaji) AS GajiTerbesar, MIN(Gaji) AS GajiTerkecil  
-> FROM pegawai
```

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

PENJELASAN :

- `SELECT` untuk memilih kolom mana saja Yang dipilih untuk ditampilkan.
- `Nocab` = nama kolom Yang ingin ditampilkan.
- `MAX (Gaji)` = untuk menampilkan nilai terbesar dan suatu data dalam kolom Yang dipilih. Gat adalah nama kolom Yang dipilih.
- `AS Gaiterbesar` = untur menssanti nama kolom hasil `MAX (Gaji)` menjadi GariTerbesar Untuk Sementara.
- `MIN (Gaji)` = untuk menampilkan nilai terkecil dari suatu data dalam kolom yg dipilih. Gaji adalah nama Kolam Yang dipilih.
- `AS Gaji` = Terkecil untuk mengganti nama kolom hasil `MIN (Gaji)` menjadi Gajiterkecil untuk sementara.
- `FROM Pegawai` = adalah untuk memilih dari tabel mana Yang data kolomnya ingin ditampilkan, Pegawai adalah nama tabel Yang dipilih.
- `GROUP BY` = untuk mengelompokkan data Pada kolom Yang dipilih.
- `Nocab` = nama kolom Yand dipilih untuk dikelompokkan. Having = Kondisi Yang harus dipenuhi oleh suatu kelompok data.
- `(COUNT(NIP) >= 3)` = kondisi dri 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 Nocab C102 Jadi hanya itu Yandg dicari Nilai maksimum dan adalah minimumnya Pada kolom Gaji.

Latiahn-17

GAMBAR :

```
MariaDB [company_ariel]> 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	30575000	3397222.2222	6250000	1725000

```
1 row in set (0.001 sec)
```

STRIKTUR : **

```
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 ditampilkan.
- `COUNT (NIP)` = untuk menghitung jumlah barisan data yang ada pada kolom Yang dipilih.
- `AS Jumlah Pegawai` = 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 Total Gaji` 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 mengganti nama dari kolom hasil `MAX(Gaji)` menjadi Gajimaks. untuk Sementara. `MIN (Gaji)` untuk menampilkan nilai terkecil dari suatu data Yang dipilih. Gaji nama kolom Yang dipilih.
- `AS GajiMIN` = 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.
- **Hasilnya** = Dihitung berapa NIP, Di Jumlahkan semua data Pada kolom Gaji, Dihitung Rata-rata dari kolom Gaji, Ditampilkan nilai terbesar Pada kolom Gaji, dan Nilai terkecil dalam Kolom Gaji.

Latiahn-18

GAMBAR :

```
MariaDB [company_ariel]> SELECT COUNT(NIP) AS Jumlahpegawai, SUM(Gaji) AS TotalGaji,
-> AVG(Gaji) AS RataGaji, Max(Gaji) AS GajiMks, MIN(Gaji) AS GajiMin
-> FROM pegawai
-> WHERE Jabatan = 'Staff' OR Jabatan = 'Sales'
-> GROUP BY NoCab HAVING SUM(Gaji) <= 2600000;
+-----+-----+-----+-----+-----+
| Jumlahpegawai | TotalGaji | RataGaji      | GajiMks | GajiMin |
+-----+-----+-----+-----+-----+
| 1             | 2500000   | 2500000.0000  | 2500000 | 2500000 |
+-----+-----+-----+-----+-----+
1 row in set (0.001 sec)
```

STRIKTUR : **

```
SELECT COUNT(NIP) AS Jumlahpegawai, SUM(Gaji) AS TotalGaji,  
-> AVG(Gaji) AS RataGaji, Max(Gaji) AS GajiMks, MIN(Gaji) AS GajiMin  
-> FROM pegawai  
-> WHERE Jabatan = 'Staff' OR Jabatan = 'Sales'  
-> GROUP BY NoCab HAVING SUM(Gaji) <= 2600000;
```

PENJELASAN :

- `SELECT` = untuk memilih kolom mana saja yang ingin digunakan.
- `COUNT (NIP)` = untuk menghitung barisan data yang ada Pada kolom Yang dipilih. AS Jumlah
- `pegawai` = 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 Total Gaji` = untuk mengganti nama dari kolom hasil `SUM(Gaji)` menjadi Total Gaji Untuk Sementara. `AVG (Gaji)` untuk menghitung rata-rata dari kolom Yand dipilih. Gaji adalah nama kolom Yang dipilih.
- `As Rata Gaji` = untuk mengganti nama dari kolom hasil `AVG (Gaji)` menjadi Rata Gaji Untuk sementara. `MAX (Gaji)` untuk menampilkan nilai 2 terbesar dan suatu data dalam kolom Yang dipilih Gaji adalah nama kolom. Yang dipilih.

TABEL KESELURUHAN PADA PRAKTIK

company_ariel	odproductsc
OrderID	int(11)
ProductID	int(11)
ProductName	varchar(255)
Quantity	int(11)
UnitPrice	decimal(10,2)

company_ariel	pegawai
NIP	int(11)
NDep	varchar(255)
NBik	varchar(255)
JK	enum('L', 'F')
Alamat	text
Telp	varchar(255)
Jabatan	enum('Manager', 'Sales', 'Staff')
Gaji	bigint(20)
NoCab	varchar(255)

company_ariel	custorderemp
CustomerID	varchar(5)
CompanyName	varchar(255)
ContactName	varchar(255)
OrderID	int(11)
OrderDate	date
EmpID	int(11)
LastName	varchar(100)
FirstName	varchar(100)

company_ariel	customers
CustomerID	varchar(5)
CompanyName	varchar(255)
ContactName	varchar(255)
Address	varchar(255)
City	varchar(100)
PostalCode	varchar(20)
Country	varchar(100)
Phone	varchar(20)

company_ariel	products
ProductID	int(11)
ProductName	varchar(255)
SupplierID	int(11)
QuantityPerUnit	varchar(50)
UnitPrice	decimal(10,2)
UnitsInStock	int(11)

company_ariel	orders
OrderID	int(11)
CustomerID	varchar(5)
EmpID	int(11)
OrderDate	date
RequiredDate	date
ShippedDate	date
ShipVia	int(11)

company_ariel	orderdetails
OrderID	int(11)
ProductID	int(11)
UnitPrice	decimal(10,2)
Quantity	int(11)
Discount	decimal(3,2)

company_ariel	employees
EmpID	int(11)
LastName	varchar(100)
FirstName	varchar(100)
Title	varchar(100)
Address	varchar(255)
City	varchar(100)
Country	varchar(100)
HomePhone	varchar(20)