

NAMA ANGGOTA

1. MIRSYA EMIYANTI
2. SITI NURKHAERANI
3. BINTANG
4. SUCI AZIZAH

Tabel keaktifan Kelompok 4

no	Nama kelompok	Skor	peran
1.	Mirsya emiyanti	3	Mencari materi
2.	Siti Nurkhaerani	3	Membuat soal dan menambahkan materi
3.	Jiliyan bintang k.	3	Membuat materi dan membuat soal
4.	Suci azizah	0	-

Pengantar

SQL adalah bahasa yang digunakan untuk mengelola dan memanipulasi basis data relasional. Dalam materi yang akan di pelajari kali ini yaitu GROUP BY , HAVING , dan COUNT. Pada SQL mereka adalah klausa penting yang sering digunakan untuk melakukan agregasi dan penyaringan data dalam kelompok.

GROUP BY

PENGERTIAN

GROUP BY digunakan untuk mengelompokkan baris yang memiliki nilai yang sama dalam satu atau lebih kolom. Klausa ini sering digunakan bersama fungsi agregat (seperti COUNT , SUM , AVG , dll.) untuk melakukan operasi pada setiap kelompok.

CONTOH KODE

SINTAKS

```
SELECT kolom1, fungsi_agregat(kolom2) FROM nama_tabel GROUP BY kolom1;
```

contoh penggunaan

misalnya kita memiliki table pegawai

```
MariaDB [company_herani]> SELECT * FROM pegawai ORDER BY NDep DESC;
```

NIP	NDep	NBlk	jenis_kelamin	alamat	Telp	jabatan	Gaji	NoCab
10415	Susan	Sumantri	Perempuan	Jl.Pahlawan 24 Surabaya	031-555120		2650000	C104
10407	Rio	Gunawan	Laki-laki	Jl.Melati 356 Surabaya	031-555231	Staf	17250000	C104
10324	Martin	Susanto	Laki-laki	Jl.Bima 51 Jakarta	021-555785	Staf	1750000	C102
10307	Erik	Andrian	Laki-laki	Jl.Manggis 5 Semarang	024-555236	Manajer	6250000	C103
10107	Emya	Salsalina	Perempuan	Jl.Suci 78 Bandung	022-555768	Manajer	5250000	C101
10246	Dian	Anggraini	Perempuan	Jl.Mawar 5 Semarang	024-555102	Sales	2750000	C103
10176	Diah	Wahyuni	Perempuan	Jl.Maluku 56 Bandung	022-555934	Sales	2500000	C101
10314	Ayu	Rahmadani	Perempuan	Jl.Malaka 342 Jakarta	022-555098	Sales	1950000	C102
10252	Antoni	Irawan	Laki-laki	Jl.A.Yani 15 Jakarta	021-555888	Manajer	5750000	C102

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

```
SELECT NoCab, COUNT(NIP) AS jumlah_pegawai FROM pegawai GROUP BY NoCab;
```

HASIL :

```
MariaDB [company_herani]> 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 (0.007 sec)
```

```
MariaDB [company_herani]>
```

- `SELECT NoCab` : digunakan untuk menentukan kolom `NoCab` yang ingin tampilkan dalam hasil query.
- `COUNT(NIP)` : digunakan untuk menghitung jumlah kolom pada tabel `pegawai` yang di mana `(NIP)` berfungsi menghitung jumlah baris yang ada di dalamnya.
- `AS jumlah_pegawai` : Hasil perhitungan diberi nama `jumlah_pegawai`, yang menjelaskan bahwa ini adalah jumlah total pegawai.
- `FROM pegawai` : digunakana untuk menunjukkan dari tabel mana data tersebut diambil, data diambil dari tabel `pegawai`.
- `GROUP BY NoCab` : digunakan untuk mengelompokkan hasil berdasarkan kolom `NoCab`. yang dimana akan dihitung jumlah pegawai (`jumlah_pegawai`) untuk setiap

cabang.

HAVING

PENGERTIAN

HAVING digunakan untuk memfilter hasil dari GROUP BY . Klausa ini mirip dengan WHERE , tetapi digunakan untuk memfilter kelompok, bukan baris individu. Perbedaan antara WHERE dan HAVING

- **WHERE:** Digunakan untuk memfilter baris sebelum pengelompokan terjadi (sebelum GROUP BY dijalankan).
- **HAVING:** Digunakan untuk memfilter hasil setelah pengelompokan dan agregasi selesai (setelah GROUP BY dijalankan).

CONTOH KODE

SINTAKS

```
SELECT kolom1, fungsi_agregat(kolom2) FROM nama_tabel GROUP BY kolom1 HAVING  
fungsi_agregat(kolom2) operator nilai;
```

Contoh penggunaan

Untuk menampilkan kolom gaji dan kolom jumlah pegawai lebih kecil dari atau sama dengan 10, kita bisa menggunakan query berikut :

```
SELECT Gaji, COUNT(NIP) AS jumlahpegawai  
FROM pegawai  
GROUP BY NoCab HAVING COUNT(NIP) <= 10;
```

HASIL :

```

MariaDB [company_herani]> SELECT Gaji, COUNT(NIP) AS jumlahpegawai
-> FROM pegawai
-> GROUP BY NoCab HAVING COUNT(NIP) <= 10;
+-----+-----+
| Gaji    | jumlahpegawai |
+-----+-----+
| 5250000 | 2             |
| 5750000 | 3             |
| 2750000 | 2             |
| 1725000 | 2             |
+-----+-----+
4 rows in set (0.016 sec)

MariaDB [company_herani]>

```

ANALISIS :

- **SELECT** : digunakan untuk menentukan kolom apa yang ingin tampilkan dalam hasil query. Kolom **Gaji** dipilih untuk ditampilkan.
- **COUNT(NIP)** : **COUNT** digunakan untuk menghitung jumlah baris yang memiliki nilai. Kolom **NIP** adalah Nomor Induk Pegawai, yang merupakan identifikasi unik untuk setiap pegawai.
- **AS jumlahpegawai** : Hasil perhitungan diberi nama **jumlahpegawai**, yang menjelaskan bahwa ini adalah jumlah total pegawai.
- **FROM pegawai** : digunakan untuk menentukan tabel yang akan diambil, data diambil dari tabel **pegawai**.
- **GROUP BY NoCab** : digunakan untuk mengelompokkan hasil berdasarkan kolom **NoCab**. yang dimana akan dihitung jumlah pegawai (**jumlah_pegawai**) untuk setiap cabang.
- **HAVING COUNT(NIP) >= 10** : digunakan untuk memfilter hasil dari grup yang sudah dibentuk oleh **GROUP BY**. Dalam hal ini grup-grup tersebut akan disaring dan hanya menampilkan cabang-cabang di mana jumlah pegawai (**jumlah_pegawai**) sama dengan atau lebih dari 10.

COUNT

PENGERTIAN

COUNT dalam SQL digunakan untuk menghitung jumlah baris yang sesuai dengan kondisi tertentu dalam sebuah query. Fungsi ini sangat berguna untuk menghitung banyaknya data yang memenuhi syarat tertentu tanpa perlu menampilkan data itu sendiri.

Contoh Kode

sintaks

```
SELECT COUNT(kolom) FROM nama_tabel WHERE kondisi;
```

contoh penggunaan

```
SELECT NoCab, COUNT(NIP) AS jumlahpegawai FROM pegawai WHERE NoCab = 'C102'
```

HASIL

```
MariaDB [company_herani]> SELECT COUNT(NIP) AS jumlahpegawai FROM pegawai where NoCab = 'C102';
+-----+
| jumlahpegawai |
+-----+
|              3 |
+-----+
1 row in set (0.010 sec)

MariaDB [company_herani]>
```

analisis :

- **SELECT** : digunakan untuk menentukan kolom apa yang ingin tampilkan dalam hasil query.
- **COUNT(NIP)** : digunakan untuk menghitung jumlah kolom pada tabel `pegawai` yang di mana `(NIP)` berfungsi menghitung jumlah baris yang ada di dalamnya.
- **AS jumlahpegawai** : Hasil perhitungan diberi nama `jumlahpegawai`, yang menjelaskan bahwa ini adalah jumlah total pegawai.
- **FROM pegawai** : digunakan untuk menunjukkan dari tabel mana data tersebut diambil, data diambil dari tabel `pegawai`.
- **WHERE NoCab = 'C102'** : digunakan untuk menyeleksi data pada baris dalam tabel `pegawai` yang memiliki data `NoCab` sama dengan `C102` yang akan dihitung.

penggabungan COUNT, HAVING, dan GROUP BY

Hubungan antara HAVING, GROUP BY, dan COUNT dalam SQL, terutama ketika kita ingin melakukan pengelompokan data dan kemudian memfilter hasil pengelompokan berdasarkan jumlah (count) dari elemen-elemen dalam kelompok tersebut.

1. GROUP BY:

- GROUP BY digunakan untuk mengelompokkan baris data berdasarkan satu atau lebih kolom. Setelah pengelompokan, fungsi agregat seperti COUNT, SUM, AVG, dll., dapat diterapkan pada setiap kelompok.

2. COUNT:

- COUNT adalah fungsi agregat yang digunakan untuk menghitung jumlah baris dalam setiap kelompok yang dihasilkan oleh GROUP BY. Kita bisa menghitung jumlah total baris (COUNT()) atau menghitung berapa kali nilai tertentu muncul dalam kolom (COUNT(kolom)).

3. HAVING:

- HAVING digunakan untuk memfilter hasil yang telah dikelompokkan oleh GROUP BY, berdasarkan kondisi yang melibatkan fungsi agregat seperti COUNT.

Hubungan antara GROUP BY, COUNT, dan HAVING:

1. Pengelompokan Data (GROUP BY):

- Data dikelompokkan berdasarkan satu atau lebih kolom menggunakan GROUP BY.

2. Menghitung Jumlah Elemen dalam Kelompok (COUNT):

- Setelah data dikelompokkan, COUNT digunakan untuk menghitung jumlah baris atau elemen dalam setiap kelompok yang dihasilkan oleh GROUP BY.

3. Memfilter Kelompok Berdasarkan Kondisi (HAVING):

- HAVING kemudian digunakan untuk memfilter kelompok berdasarkan hasil fungsi agregat, seperti COUNT. Ini memungkinkan kita hanya menampilkan kelompok yang memenuhi kondisi tertentu.

contoh penggunaan

CODE

```
SELECT MAX(Gaji) AS Gaji_Terbesar, MIN(Gaji) AS Gaji_Terkecil,  
FROM pegawai GROUP BY NoCab HAVING COUNT(NIP) >= 3;
```

HASIL :

```

MariaDB [company_herani]> SELECT MAX(Gaji) AS Gaji_Terbesar, MIN(Gaji) AS Gaji_Terkecil
-> FROM pegawai GROUP BY NoCab HAVING COUNT(NIP) >= 3;
+-----+-----+
| Gaji_Terbesar | Gaji_Terkecil |
+-----+-----+
|      5750000 |      1750000 |
+-----+-----+
1 row in set (0.001 sec)

MariaDB [company_herani]>

```

ANALISIS :

- **SELECT** : digunakan untuk menentukan kolom mana yang ingin ditampilkan dalam hasil.
- **MAX(Gaji)** : **MAX()** digunakan untuk menemukan nilai maksimum (nilai terbesar) dari kolom. **Gaji** adalah kolom yang berisi nilai gaji dari setiap pegawai.
- **AS Gaji_Terbesar** : Hasil perhitungan diberi nama **Gaji_Terbesar** , yang menjelaskan bahwa ini adalah maksimum atau nilai terbesar gaji pada kolom pegawai.
- **MIN(Gaji)** : **MIN()** digunakan untuk menentukan nilai minimum atau nilai terkecil dari kolom. **Gaji** adalah kolom yang berisi nilai gaji dari setiap pegawai.
- **AS Gaji_Terkecil** : Hasil perhitungan diberi nama **Gaji_Terkecil** , yang menjelaskan bahwa ini adalah minimum atau nilai terkecil gaji pada kolom pegawai.
- **FROM pegawai** : digunakan untuk menunjukkan dari tabel mana data tersebut diambil, data diambil dari tabel **pegawai** .
- **GROUP BY NoCab** : digunakan untuk mengelompokkan data berdasarkan kolom **NoCab**
- **HAVING COUNT(NIP) >= 3** : digunakan untuk menyaring kelompok hasil setelah proses **GROUP BY** . **COUNT(NIP)** yang digunakan untuk menghitung jumlah baris dalam setiap kelompok yang memiliki nilai **NIP** kondisi **>= 3** berarti hanya yang memiliki tiga atau lebih pegawai yang akan ditampilkan dalam hasil.

Fungsi agregat

1. SUM

sum merupakan fungsi agregat yang digunakan untuk menghitung jumlah total dari nilai dalam kolom yang sudah ditentukan. fungsi ini berguna untuk menjumlahkan nilai dari semua baris dalam kelompok data.

contoh :

```
SELECT NoCab, SUM(Gaji) AS Total_Gaji
FROM pegawai
GROUP BY jabatan;
```

```
FROM pegawai GROUP BY jabatan - at line 1
MariaDB [company_herani]> SELECT jabatan, SUM(Gaji) AS Total_Gaji
-> FROM pegawai GROUP BY jabatan;
+-----+-----+
| jabatan | Total_Gaji |
+-----+-----+
|         |            |
| Manajer | 2650000    |
| Staf    | 17250000   |
| Sales   | 3475000    |
|         | 7200000    |
+-----+-----+
4 rows in set (0.018 sec)

MariaDB [company_herani]>
```

2. AVG

AVG merupakan fungsi agregat yang digunakan untuk menghitung rata-rata nilai dari kolom numerik. Fungsi ini memberikan nilai rata-rata berdasarkan baris yang dipilih atau dikelompokkan .

contoh

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

```
MariaDB [company_herani]> SELECT AVG(Gaji) AS Rata_rata FROM pegawai;
+-----+
| Rata_rata |
+-----+
| 3397222.2222 |
+-----+
1 row in set (0.001 sec)

MariaDB [company_herani]> _
```

3. MAX

MAX merupakan fungsi agregat yang digunakan untuk mengembalikan nilai maksimum dari kolom numerik. fungsi ini berguna untuk menemukan nilai terbesar dalam data.

contoh


```
SELECT MAX(Gaji) AS Gaji_terbesar  
FROM pegawai;
```

```
MariaDB [company_herani]> SELECT MAX(Gaji)  
-> AS Gaji_terbesar FROM pegawai;  
+-----+  
| Gaji_terbesar |  
+-----+  
|          625000 |  
+-----+  
1 row in set (0.263 sec)
```

MIN

CONTOH

```
SELECT MIN(Gaji) AS Gaji_terkecil FROM pegawai;
```

```
MariaDB [company_herani]> SELECT MIN(Gaji)  
-> AS Gaji_terkecil FROM pegawai;  
+-----+  
| Gaji_terkecil |  
+-----+  
|          172500 |  
+-----+  
1 row in set (0.002 sec)
```