

## MODUL 5

## FUNGSI AGREGASI

### MODUL 5 : FUNGSI AGREGASI

#### Tujuan :

Setelah menyelesaikan modul ini, mahasiswa diharapkan dapat : Mengetahui dan mampu menggunakan perintah fungsi agregasi dengan baik.

#### Fungsi Agregasi

Fungsi agregasi atau *aggregate function* merupakan fungsi yang ada di dalam mysql yang digunakan untuk melakukan perhitungan pada *query*. Pada umumnya dikombinasikan dengan klausa GROUP BY untuk menghasilkan nilai yang dikelompokkan berdasarkan kolom tertentu. Fungsi agregasi diantaranya yaitu : COUNT, SUM, AVG, MIN, MAX dan GROUP CONCAT

##### a. C O U N T

Perintah yang digunakan untuk menghitung jumlah baris suatu kolom pada tabel.

Query yang digunakan :

```
SELECT COUNT(namafield) FROM nama_tabel;
```

##### b . SUM

Perintah yang digunakan untuk menghitung jumlah nilai suatu kolom pada tabel. Query yang digunakan :

```
SELECT SUM(namafield) FROM nama_tabel;
```

##### c . AVG

Perintah yang digunakan untuk menghitung rata- rata dari nilai suatu kolom pada tabel.

Query yang digunakan :

```
SELECT AVG(namafield) FROM nama_tabel;
```

d . MIN

Perintah yang digunakan untuk menampilkan nilai terkecil dari suatu kolom pada tabel.

Query yang digunakan :

```
SELECT MIN(namafield) FROM nama_tabel;
```

e . MAX

Perintah yang digunakan untuk menampilkan nilai terbesar dari suatu kolom pada tabel.

Query yang digunakan :

```
SELECT MAX(namafield) FROM nama_tabel;
```

f. GROUP CONCAT

Merupakan fungsi yang berguna untuk menyambung beberapa baris dari suatu kolom menjadi string tunggal. Query yang digunakan :

```
SELECT GROUP_CONCAT(namafield) FROM nama_tabel;
```

## **FUNGSI QUERY LAINNYA**

- GROUP BY : berfungsi untuk mengelompokkan data, dengan syarat data yang akan dikelompokkan mempunyai data yang sejenis atau sama.
- HAVING : penggunaan having terkait dengan group by, kegunaannya untuk menentukan kondisi bagi group by, dimana kelompok yang memenuhi kondisi saja yang akan dihasilkan.
- CASE : perintah ini digunakan untuk menampilkan nilai tertentu dari beberapa barisan data dengan syarat-syarat dari kondisi yang akan diberikan. Penggunaan perintah ini akan membentuk kolom baru dengan hasil data operasi yang dimasukkan kedalamnya.
- VIEW : tampilan tabel dengan data yang ingin ditampilkan bisa kita pilih. Kelebihannya dapat menyimpan perintah query dan dapat mewakili sebuah subset dari tabel asli dan memilih kolom tertentu dari tabel biasa.

## LANGKAH PRAKTIKUM

1. Untuk contoh agregasi buat tabel seperti dibawah ini :

NIM	Nama	JenisKelamin	TanggalLahir	Prodi	JumlahSKS
H101	Andi	L	1999-10-01	Kimia	45
H102	Erika	P	1999-10-07	Kimia	39
H103	Fika	P	1997-10-05	Kimia	70
H201	Budi	L	1998-12-10	Fisika	45
H202	Citra	P	1997-02-03	Fisika	65
H301	Dian	P	1998-05-12	Biologi	42
H302	Gito	L	1998-03-01	Biologi	44
H303	Ira	P	1999-01-10	Biologi	35
H401	Henri	L	1998-07-21	Sistem Informasi	40
H402	Joko	L	1997-01-28	Sistem Informasi	55

2. Untuk menghitung Jumlah SKS yang paling tertinggi dan terendah dari tabel dataMhs menggunakan perintah MIN dan MAX.

```
mysql> select MIN(JumlahSKS) from dataMhs;
+-----+
| MIN(JumlahSKS) |
+-----+
| 35              |
+-----+
```

```
mysql> select MAX(JumlahSKS) from dataMhs;
+-----+
| MAX(JumlahSKS) |
+-----+
| 70              |
+-----+
```

3. Menghitung rata-rata Jumlah SKS pada tabel dataMhs, menggunakan perintah AVG.

```
mysql> select AVG(JumlahSKS) from dataMhs;
+-----+
| AVG(JumlahSKS) |
+-----+
| 48              |
+-----+
```

4. Untuk menghitung jumlah mahasiswa yang ada di prodi biologi digunakan fungsi **count**.

```
mysql> select COUNT(*) as Total_Mahasiswa_Biologi from dataMhs
-> where Prodi = 'Biologi';
+-----+
| Total_Mahasiswa_Biologi |
+-----+
| 3                        |
+-----+
```

5. Untuk menghitung jumlah mahasiswa di prodi fisika dan memiliki jumlah\_sks = 45.

```
mysql> select COUNT(*) as Jumlah_Mahasiswa from dataMhs
-> where Prodi='Fisika' AND JumlahSKS='45';
```

Jumlah_Mahasiswa
1

6. Untuk mengetahui Jumlah SKS paling sedikit, paling banyak dan rata-rata Jumlah SKS menggunakan fungsi **MIN**, **MAX** dan **AVG**. Ketiga fungsi tersebut dapat digunakan dalam satu kali query seperti berikut.

```
mysql> select MIN(JumlahSKS), MAX(JumlahSKS), AVG(JumlahSKS) from dataMhs;
```

MIN(JumlahSKS)	MAX(JumlahSKS)	AVG(JumlahSKS)
35	70	48

7. Untuk mengetahui distribusi jumlah mahasiswa setiap prodi pada tabel dataMhs dapat dikombinasikan dengan klausa **GROUP BY**.

```
mysql> select Prodi,COUNT(*) as JumlahMahasiswa from dataMhs
-> GROUP BY Prodi;
```

Prodi	JumlahMahasiswa
Biologi	3
Fisika	2
Kimia	3
Sistem Informasi	2

8. Penggunaan fungsi **HAVING**.

```
mysql> select Prodi,COUNT(*) as JumlahMahasiswa from dataMhs
-> GROUP BY Prodi HAVING COUNT(*)<3;
```

Prodi	JumlahMahasiswa
Fisika	2
Sistem Informasi	2

9. Penggunaan fungsi CASE.

```
mysql> select NIM, Nama, Prodi, CASE
-> when JenisKelamin='L' Then 'Laki-laki'
-> else 'Perempuan' End as JenisKelamin From dataMhs;
```

NIM	Nama	Prodi	JenisKelamin
H101	Andi	Kimia	Laki-laki
H102	Erika	Kimia	Perempuan
H103	Fika	Kimia	Perempuan
H201	Budi	Fisika	Laki-laki
H202	Citra	Fisika	Perempuan
H301	Dian	Biologi	Perempuan
H302	Gito	Biologi	Laki-laki
H303	Ira	Biologi	Perempuan
H401	Henri	Sistem Informasi	Laki-laki
H402	Joko	Sistem Informasi	Laki-laki

10. Penggunaan VIEW untuk menampilkan data NIM, NAMA dan Prodi pada tabel dataMhs.

```
mysql> create view MahasiswaProdi as Select NIM, Nama, Prodi
-> from dataMhs;
Query OK, 0 rows affected (0.02 sec)
```

```
mysql> select * from MahasiswaProdi;
```

NIM	Nama	Prodi
H101	Andi	Kimia
H102	Erika	Kimia
H103	Fika	Kimia
H201	Budi	Fisika
H202	Citra	Fisika
H301	Dian	Biologi
H302	Gito	Biologi
H303	Ira	Biologi
H401	Henri	Sistem Informasi
H402	Joko	Sistem Informasi

10 rows in set (0.00 sec)

11. Contoh penggunaan GROUP CONCAT pada kolom prodi tabel dataMhs.

```
mysql> select Group_Concat(Prodi) from dataMhs;
```

Group_Concat(Prodi)
Kimia,Kimia,Kimia,Fisika,Fisika,Biologi,Biologi,Biologi,Sistem Informasi,Sistem Informasi

1 row in set (0.00 sec)