

## Modul Praktikum II Matakuliah Basis Data Lanjut

### “Fungsi Agregat pada MySQL”

Disusun oleh: **Khoirul Umam, M.Kom** (NIDN. 0711039101)

#### A. TUJUAN

- Mahasiswa mampu menggunakan fungsi COUNT untuk menghitung banyaknya data dari hasil suatu *query* di dalam basis data
- Mahasiswa mampu menggunakan fungsi SUM untuk menjumlahkan nilai dari sekumpulan data yang tersimpan di dalam basis data
- Mahasiswa mampu menggunakan fungsi AVG untuk menghitung rata-rata nilai dari sekumpulan data di dalam basis data
- Mahasiswa mampu menggunakan fungsi MAX dan MIN untuk mencari nilai terbesar dan terkecil dari sekumpulan data di dalam basis data
- Mahasiswa mampu menggunakan fungsi STDDEV dan VARIANCE untuk menghitung nilai standar deviasi dan variasi sekumpulan data di dalam basis data
- Mahasiswa mampu mengelompokkan data sekaligus juga memberikan kriteria pada kelompok data tersebut menggunakan klausa GROUP BY dan HAVING

#### B. PETUNJUK PELAKSANAAN PRAKTIKUM

- Awali setiap aktivitas dengan do’a, semoga berkah dan mendapat kemudahan.
- Pahami tujuan dan dasar teori dengan baik dan benar.
- Kerjakan latihan dan tugas-tugas praktikum dengan baik, sabar, mandiri, dan jujur.
- Tanyakan kepada asisten praktikum / dosen apabila ada hal-hal yang kurang jelas / menjumpai kesulitan saat melaksanakan kegiatan praktikum.

#### C. DASAR TEORI

Fungsi agregat merupakan fungsi yang berhubungan dengan sekumpulan data pada basis data. Oleh karena itu fungsi tersebut dapat pula disebut sebagai fungsi grup atau fungsi ringkasan. Fungsi agregat merupakan fungsi yang menerima dan memproses nilai dari sekumpulan data kemudian mengembalikan nilai tunggal sebagai hasilnya.

##### 1. Macam-Macam Fungsi Agregat

Pada MySQL terdapat beberapa macam fungsi agregat. Namun pada modul praktikum ini hanya akan dibahas 7 buah fungsi yang paling lazim digunakan, yaitu (1) AVG, (2) COUNT, (3) SUM, (4) MAX, (5) MIN, (6) STDDEV, dan (7) VARIANCE. Penjelasan mengenai masing-masing fungsi tersebut dijabarkan pada sub-pembahasan berikutnya.

### 1.1 Fungsi AVG

Fungsi AVG merupakan fungsi agregat pada MySQL yang dapat digunakan untuk menghitung nilai rata-rata dari sekumpulan data numerik. Nama "AVG" sendiri merupakan akronim dari *average* yang berarti rata-rata.

### 1.2 Fungsi COUNT

Fungsi COUNT merupakan fungsi agregat yang dapat digunakan untuk melakukan pencaharian terhadap data. Fungsi ini akan mengembalikan nilai berupa berapa banyak data yang ada pada sekumpulan data. Penggunaan fungsi ini tidak terbatas pada data numerik saja, melainkan dapat pula digunakan pada data non-numerik.

### 1.3 Fungsi SUM

Fungsi SUM merupakan fungsi agregat yang dapat digunakan untuk menjumlahkan seluruh nilai dari sekumpulan data numerik. Perlu diperhatikan bahwa fungsi ini berbeda dengan fungsi COUNT.

### 1.4 Fungsi MAX dan MIN

Fungsi MAX merupakan fungsi agregat yang dapat digunakan untuk mencari nilai terbesar dari sekumpulan data, sedangkan fungsi MIN sebaliknya, digunakan untuk mencari nilai terkecil dari sekumpulan data. Kedua fungsi tidak hanya berlaku bagi data numerik saja, melainkan juga dapat diterapkan pada data non-numerik.

### 1.5 Fungsi STDDEV dan VARIANCE

Fungsi STDDEV dan VARIANCE merupakan fungsi agregat yang berhubungan dengan statistika. Fungsi STDDEV berguna untuk menghitung nilai standar deviasi suatu kumpulan data, sedangkan fungsi VARIANCE berguna untuk menghitung nilai variasi sekumpulan data tersebut.

## 2. Fungsi Agregat untuk Data Unik

Pengolahan sekumpulan data menggunakan fungsi agregat terkadang dapat menjumpai adanya data-data kembar. Pada suatu kasus mungkin saja data-data tersebut tidak menjadi masalah sehingga fungsi agregat dapat digunakan sebagaimana mestinya. Akan tetapi mungkin juga pengguna basis data menginginkan agar data-data yang diolah hanyalah data-data yang unik saja. Dengan demikian maka diperlukan adanya mekanisme pengeliminasian data-data kembar dari sekumpulan data.

Untuk mengeliminasi data-data kembar, maka klausa **DISTINCT** dapat digunakan di dalam fungsi agregat. Dengan demikian fungsi agregat tersebut hanya akan mengolah data-data yang unik saja dan mengabaikan data-data kembar.

## 3. Pengelompokan Data

Hasil pengambilan data menggunakan perintah SELECT pada MySQL dapat dikelompokkan berdasarkan kolom tertentu. Hal ini dapat memudahkan pengguna basis data untuk memperoleh informasi terkait kelompok data tertentu secara bersamaan. Selain itu data yang telah dikelompokkan juga dapat ditampilkan berdasarkan kriteria tertentu saja.

Pengelompokkan data pada MySQL dapat dilakukan dengan menyertakan klausa **GROUP BY** di dalam *query* SELECT. Sedangkan pembatasan atau pengkriteriaan pada data yang telah dikelompokkan dapat dilakukan dengan menambahkan klausa **HAVING** di dalam *query* tersebut.

## 2.1 Klausa GROUP BY

Penggunaan klausa GROUP BY di dalam *query* SELECT akan mengakibatkan data-data yang diperoleh dikelompokkan berdasarkan suatu kolom. Format penggunaan klausa tersebut adalah sebagai berikut:

```
SELECT kolom FROM tabel
GROUP BY nama_kolom_acuan;
```

## 2.2 Klausa HAVING

Fungsi dari klausa HAVING mirip dengan klausa WHERE pada umumnya. Namun klausa HAVING khusus digunakan berpasangan dengan klausa GROUP BY. Dengan demikian dapat dikatakan bahwa klausa HAVING merupakan pengganti klausa WHERE saat klausa GROUP BY digunakan pada suatu *query* SELECT. Format penggunaan klausa tersebut adalah sebagai berikut:

```
SELECT kolom FROM tabel
GROUP BY nama_kolom_acuan
HAVING kriteria;
```

## D. LATIHAN

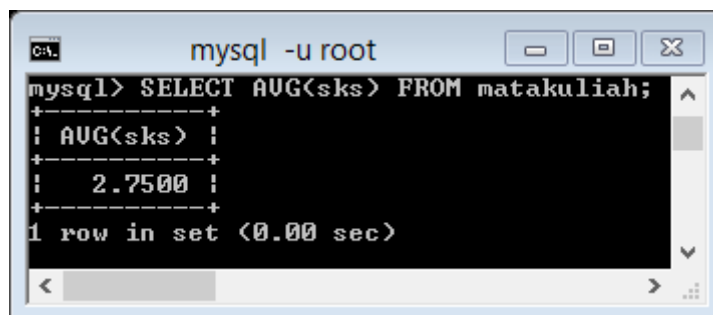
Pada latihan di dalam modul ini gunakan kembali basis data **kampus** yang telah Anda buat pada bagian Latihan Modul 1.

### 1. Penggunaan Fungsi AVG

Contoh penggunaan fungsi AVG pada basis data **kampus** adalah untuk menghitung rata-rata SKS dari seluruh matakuliah yang tercatat pada basis data tersebut. *Query* untuk mendapatkan nilai rata-rata tersebut adalah sebagai berikut:

```
SELECT AVG(sks) FROM matakuliah;
```

Jika *query* tersebut dijalankan maka akan diperoleh hasil sebagai berikut:



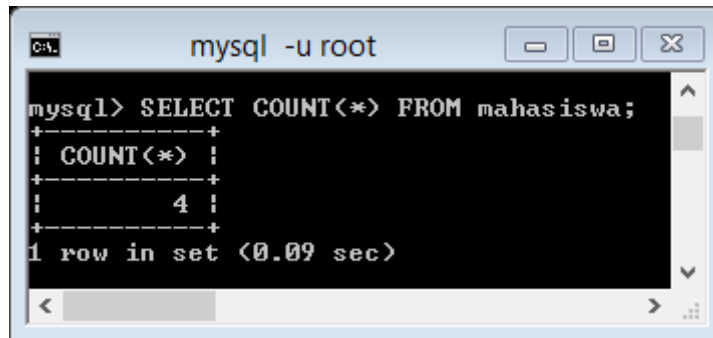
Dari hasil tersebut dapat diketahui bahwa matakuliah yang tercatat di dalam basis data **kampus** memiliki rata-rata SKS sebesar 2,75.

## 2. Penggunaan Fungsi COUNT

Contoh penggunaan fungsi COUNT adalah untuk menghitung ada berapa banyak mahasiswa yang dimiliki oleh suatu kampus. Jika dianggap jumlah data mahasiswa yang tercatat di dalam basis data **kampus** menunjukkan jumlah mahasiswa yang dimiliki oleh kampus tersebut, maka *query* yang digunakan adalah sebagai berikut:

```
SELECT COUNT(*) FROM mahasiswa;
```

*Query* tersebut akan memberikan hasil sebagai berikut:



```
mysql -u root

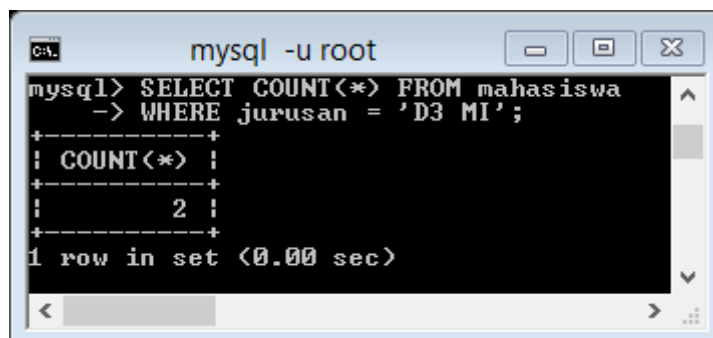
mysql> SELECT COUNT(*) FROM mahasiswa;
+-----+
| COUNT(*) |
+-----+
|         4 |
+-----+
1 row in set (0.09 sec)
```

Dari hasil tersebut dapat diketahui bahwa kampus tersebut hanya memiliki 4 orang mahasiswa.

Jika informasi yang ingin diketahui adalah jumlah mahasiswa yang memilih jurusan D3 MI, maka perlu ditambahkan pengkondisian dengan memanfaatkan klausa WHERE pada *query* yang digunakan untuk memperoleh informasi tersebut. *Query* lengkapnya adalah sebagai berikut:

```
SELECT COUNT(*) FROM mahasiswa
WHERE jurusan = 'D3 MI';
```

Jika *query* tersebut dijalankan, maka akan diperoleh hasil sebagai berikut:



```
mysql -u root

mysql> SELECT COUNT(*) FROM mahasiswa
-> WHERE jurusan = 'D3 MI';
+-----+
| COUNT(*) |
+-----+
|         2 |
+-----+
1 row in set (0.00 sec)
```

## 3. Penggunaan Fungsi SUM

Contoh penggunaan fungsi SUM adalah untuk menghitung jumlah total SKS dari seluruh matakuliah. Untuk menghitung hal tersebut, maka *query* yang digunakan adalah sebagai berikut:

```
SELECT SUM(sks) FROM matakuliah;
```

Hasil dari *query* tersebut adalah sebagai berikut:

```
mysql -u root
mysql> SELECT SUM(sks) FROM matakuliah;
+-----+
| SUM(sks) |
+-----+
|        11 |
+-----+
1 row in set (0.11 sec)
```

#### 4. Penggunaan Fungsi MAX dan MIN

Apabila ingin diketahui berapa SKS terbesar sekaligus juga yang terkecil dari sekumpulan matakuliah yang ada di basis data **kampus**, maka fungsi MAX juga MIN dapat digunakan untuk memperoleh informasi tersebut. *Query* yang digunakan untuk memperoleh informasi tersebut adalah sebagai berikut:

```
SELECT MAX(sks), MIN(sks) FROM matakuliah;
```

*Query* tersebut akan memberikan hasil sebagai berikut:

```
mysql -u root
mysql> SELECT MAX(sks), MIN(sks) FROM matakuliah;
+-----+-----+
| MAX(sks) | MIN(sks) |
+-----+-----+
|         4 |         2 |
+-----+-----+
1 row in set (0.08 sec)
```

#### 5. Penggunaan Fungsi STDDEV dan VARIANCE

Sama dengan fungsi-fungsi agregat sebelumnya, fungsi STDDEV dan VARIANCE juga dapat digunakan pada basis data **kampus**. Contohnya adalah untuk mengetahui nilai standar deviasi dan variasi dari SKS seluruh matakuliah. *Query* yang digunakan untuk hal tersebut adalah sebagai berikut:

```
SELECT STDDEV(sks), VARIANCE(sks) FROM matakuliah;
```

Hasilnya adalah sebagai berikut:

```
mysql -u root
mysql> SELECT STDDEV(sks), VARIANCE(sks) FROM matakuliah;
+-----+-----+
| STDDEV(sks) | VARIANCE(sks) |
+-----+-----+
| 0.82915619758885 |         0.6875 |
+-----+-----+
1 row in set (0.00 sec)
```

## 6. Penggunaan Klausa DISTINCT pada Fungsi Agregat

Tabel **kelas** pada basis data **kampus** mencatat pembagian tugas mengampu suatu mata kuliah kepada setiap dosen. Dari tabel tersebut dapat diketahui ada berapa banyak dosen yang sudah mendapatkan tugas mengampu suatu matakuliah. Untuk mendapatkan informasi mengenai hal tersebut, maka dapat digunakan *query* sebagai berikut:

```
SELECT COUNT(DISTINCT id_dosen) FROM kelas;
```

Dengan menjalankan *query* tersebut maka akan diperoleh hasil sebagai berikut:

```
mysql -u root
mysql> SELECT COUNT(DISTINCT id_dosen) FROM kelas;
+-----+
| COUNT(DISTINCT id_dosen) |
+-----+
| 3 |
+-----+
1 row in set (0.02 sec)
```

Perhatikan bahwa penggunaan klausa DISTINCT pada fungsi COUNT akan menyebabkan fungsi tersebut hanya memproses data-data yang unik saja. Selain itu fungsi COUNT akan mengabaikan nilai *null*.

## 7. Pengelompokan Data

Contoh kasus pengelompokan data pada basis data **kampus** adalah untuk mengetahui ada berapa banyak mahasiswa yang tercatat pada masing-masing jurusan di kampus tersebut. *Query* yang digunakan untuk kasus tersebut adalah sebagai berikut:

```
SELECT jurusan, COUNT(jurusan)
FROM mahasiswa
GROUP BY jurusan;
```

Hasil dari *query* tersebut adalah sebagai berikut:

```
mysql -u root
mysql> SELECT jurusan, COUNT(jurusan)
-> FROM mahasiswa
-> GROUP BY jurusan;
+-----+-----+
| jurusan | COUNT(jurusan) |
+-----+-----+
| D3 MI   | 2 |
| S1 TI   | 2 |
+-----+-----+
2 rows in set (0.11 sec)
```

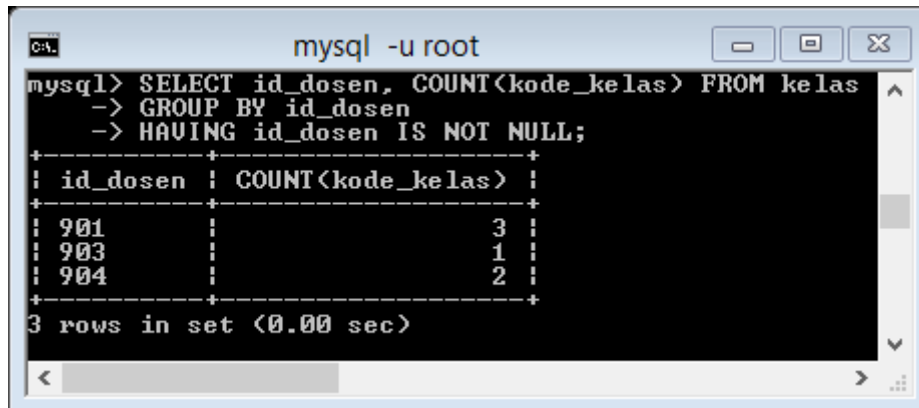
Dari hasil tersebut dapat diketahui bahwa pada masing-masing jurusan baru terdapat 2 mahasiswa.

## 8. Pengelompokan Data dengan Kriteria

Contoh kasus pengelompokan data yang memiliki suatu kriteria adalah mengetahui jumlah kelas yang diampu oleh masing-masing dosen. *Query* untuk kasus tersebut adalah sebagai berikut:

```
SELECT id_dosen, COUNT(kode_kelas) FROM kelas
GROUP BY id_dosen
HAVING id_dosen IS NOT NULL;
```

Jika *query* tersebut dijalankan akan diperoleh hasil sebagai berikut:



The screenshot shows a MySQL command window titled 'mysql -u root'. The user has entered the following query:

```
mysql> SELECT id_dosen, COUNT(kode_kelas) FROM kelas
-> GROUP BY id_dosen
-> HAVING id_dosen IS NOT NULL;
```

The output of the query is displayed in a table format:

id_dosen	COUNT(kode_kelas)
901	3
903	1
904	2

At the bottom of the window, it says '3 rows in set (0.00 sec)'.

Perhatikan bahwa klausa **HAVING** akan menyebabkan data-data yang dimunculkan hanyalah data-data yang sesuai dengan suatu kriteria. Pada contoh di atas, data kelas yang belum memiliki dosen pengampu tidak dimunculkan.

## E. TUGAS

Gunakan kembali basis data **onlineshop** yang sudah Anda buat pada Tugas Modul 1 untuk mengerjakan tugas-tugas praktikum berikut. Tuliskan *query* untuk menyelesaikan kasus-kasus berikut pada laporan praktikum! Gambarkan pula hasil dari *query* tersebut!

1. Hitung rata-rata harga seluruh produk!
2. Hitung total harga seluruh produk!
3. Hitung total berat seluruh produk yang harga jualnya di bawah 1 juta!
4. Hitung total pembayaran yang harus dilakukan oleh pembeli dengan kode B0002 atas transaksi pembelian barang yang ia lakukan!
5. Hitung ada berapa macam kategori produk yang dapat dipilih oleh penjual saat ingin menjual barangnya!
6. Hitung ada berapa banyak jenis barang yang dijual oleh masing-masing penjual!
7. Hitung ada berapa macam bank berbeda yang digunakan oleh pembeli!
8. Hitung ada berapa banyak produk baru yang dijual untuk masing-masing kategori produk!
9. Tampilkan harga produk termahal!
10. Tampilkan berat produk paling ringan!
11. Tampilkan nilai standar deviasi sekaligus variasi dari harga jual produk!