

**TUGAS MODUL 7**  
**PRAKTIKUM BASIS DATA**  
**AGREGASI**

Oleh:

Nama : Fitra Ilyasa  
Nim : 120140048  
Kelas : Basis Data RB



**PROGRAM STUDI TEKNIK INFORMATIKA**  
**JURUSAN TEKNIK ELEKTRO INFORMATIKA DAN SISTEM FISIS**  
**INSTITUT TEKNOLOGI SUMATERA**  
**LAMPUNG SELATAN**

**2022**

# BAB I

## TEORI DASAR

---

### Agregasi

Agregasi merupakan salah satu teknik dalam SQL yang berfungsi untuk mendapatkan nilai tertentu dari data yang telah dikelompokkan. Pengelompokkan data ini didasarkan pada satu atau lebih kolom yang dapat dikombinasikan dari beberapa kolom yang dipilih. Berikut ini beberapa fungsi dalam agregasi antara lain :

1. MAX, Berfungsi untuk mencari nilai terbesar dari sekelompok data. Dengan sintaks sebagai berikut :  
`Select MAX <nama_kolom> FROM <nama_tabel>;`
2. MIN, Berfungsi untuk mencari nilai terkecil dari sekelompok data. Dengan sintaks sebagai berikut :  
`Select MIN <nama_kolom> FROM <nama_tabel>;`
3. COUNT, Berfungsi untuk mencari cacah data dari sekelompok data. Dengan sintaks sebagai berikut :  
`Select COUNT <nama_kolom> FROM <nama_tabel>;`
4. SUM, Berfungsi untuk mencari jumlah dari sekumpulan data numerik. Dengan sintaks sebagai berikut :  
`Select SUM <nama_kolom> FROM <nama_tabel>;`
5. AVG, Berfungsi untuk mencari rerataan dari sekumpulan data numerik. Dengan sintaks sebagai berikut :  
`Select AVG <nama_kolom> FROM <nama_tabel>;`

### Grouping

Grouping merupakan salah satu fungsi dalam SQL yang digunakan untuk mengelompokkan suatu data berdasarkan satu kolom yang diinginkan pada suatu tabel. Berikut ini beberapa fungsi dalam Grouping antara lain :

1. ORDER BY, Berfungsi untuk menampilkan data terurut berdasarkan nilai tertentu.

Terdapat dua jenis ORDER BY yaitu ASCENDING dengan data dari nilai rendah ke tinggi dan DESCENDING dengan data dari nilai tinggi ke rendah. Dengan sintaks sebagai berikut :

```
Select * FROM <nama_tabel> ORDER BY <nama_kolom> ASC/DESC;
```

2. GROUP BY, Berfungsi untuk mengelompokkan beberapa data. Dengan sintaks sebagai berikut :

```
Select * FROM <nama_tabel> GROUP BY <nama_kolom>
```

3. HAVING, Berfungsi dalam menentukan kondisi yang diinginkan pada data, Having memiliki kemiripan dengan where, hanya saja where tidak dapat digunakan pada fungsi agregasi. Dengan sintaks sebagai berikut :

```
SELECT <fungsi agregasi> FROM <nama table> GROUP BY <ekspresi>  
HAVING <kondisi>;
```

## BAB II

### PEMBAHASAN & ANALISIS

1. Tambahkan data tabel produk sesuai soal:

**Command Line :** `insert produk values ('P150', 'Kretendeng', '80PCS', 'S002'), ('P792', 'miesadap', '30pcs', 'S005'), ('P204', 'somos', '50pcs', 'S005'), ('P561', 'Marijan', '40pcs', 'S004');`

**Command Line :** `select * from produk;`

```

XAMPP for Windows - mysql -u root -p
MariaDB [galeri_itera]> insert produk values ('P150', 'Kretendeng', '80PCS', 'S002'), ('P792', 'miesadap', '30pcs', 'S005'), ('P204', 'somos', '50pcs', 'S005'), ('P561', 'Marijan', '40pcs', 'S004');
Query OK, 4 rows affected (0.473 sec)
Records: 4  Duplicates: 0  Warnings: 0

MariaDB [galeri_itera]> select * from produk;
+-----+-----+-----+-----+
| Produk_id | Produk_nama | Jumlah_Stok | Supplier_id |
+-----+-----+-----+-----+
| P109      | The Kotak 300 ml | 40 pcs      | S002        |
| P114      | Milo 100 ml     | 800 pcs     | S001        |
| P115      | Milo 150 ml     | 50 pcs      | S003        |
| P123      | Gulaku 1 Kg     | 100 pcs     | S005        |
| P150      | Kretendeng      | 80PCS       | S002        |
| P204      | somos           | 50pcs       | S005        |
| P235      | Aqua 250 ml     | 300 pcs     | S001        |
| P311      | Grand 320 ml    | 400 pcs     | S003        |
| P333      | Sari Roti 100 gram | 30 pcs     | S005        |
| P441      | Rojo Lele 5 kg  | 60 pcs      | S002        |
| P453      | Garam 30 gram   | 20 pcs      | S006        |
| P552      | Aqua 1 L        | 300 pcs     | S001        |
| P561      | Marijan         | 40pcs       | S004        |
| P792      | miesadap        | 30pcs       | S005        |
| P882      | Indomilk 25 ml  | 200 pcs     | S004        |
+-----+-----+-----+-----+
15 rows in set (0.046 sec)

```

2. Tabel pegawai, tambahkan kolom alamat (VARCHAR 50) dan jenis kelamin (VARCHAR 15) serta menambahkan 3 orang pegawai dengan rincian sesuai soal :

**Command Line :** `alter table pegawai add Jenis_kelamin varchar(15), add column alamat varchar(50);`

**Command Line :** `insert pegawai values ('Pg_007', 'Dani', 'Staff', 'Laki-laki', 'Jl. Suka Maju'), ('Pg_008', 'Doni', 'Staff', 'Laki-laki', 'Jl. Suka Mundur'), ('Pg_009', 'Dian', 'Staff', 'Perempuan', 'Jl. Ryacudu');`

**Command Line :** `select * from pegawai;`

```
XAMPP for Windows - mysql -u root -p
MariaDB [galeri_itera]> alter table pegawai add Jenis_kelamin varchar(15), add column alamat varchar(50);
Query OK, 0 rows affected (0.624 sec)
Records: 0 Duplicates: 0 Warnings: 0

MariaDB [galeri_itera]> insert pegawai values ('Pg_007', 'Dani', 'Staff', 'Laki-laki', 'Jl. Suka Maju'), ('Pg_008', 'Doni', 'Staff', '
Pg_009', 'Dian', 'Staff', 'Perempuan', 'Jl. Ryacudu');
Query OK, 3 rows affected (0.068 sec)
Records: 3 Duplicates: 0 Warnings: 0

MariaDB [galeri_itera]> select * from pegawai;
```

Id_pegawai	Pegawai_nama	Jabatan	Jenis_kelamin	alamat
Pg_001	Santi	Cashier	NULL	NULL
Pg_002	Siska	Casier	NULL	NULL
Pg_003	Nuri	Casier	NULL	NULL
Pg_004	Jamal	Casier	NULL	NULL
Pg_007	Dani	Staff	Laki-laki	Jl. Suka Maju
Pg_008	Doni	Staff	Laki-laki	Jl. Suka Mundur
Pg_009	Dian	Staff	Perempuan	Jl. Ryacudu
Pg_011	Sinta	Manager	NULL	NULL
Pg_201	Santo	cashier	NULL	NULL

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

3. Tabel pembeli, tambahkan kolom kota (VARCHAR 25) dengan rincian kota yang dapat digunakan adalah sesuai soal :

**Command Line : alter table pembeli add Kota varchar(25);**

**Command Line : update pembeli set kota = 'Jakarta' where Id\_pembeli = 'C\_890';**

**Command Line : update pembeli set kota = 'Bandung' where Id\_pembeli = 'C\_901';**

**Command Line : update pembeli set kota = 'Yogyakarta' where Id\_pembeli = 'C\_991';**

**Command Line : select \* from pembeli;**

```
XAMPP for Windows - mysql -u root -p
MariaDB [galeri_itera]> alter table pembeli add Kota varchar(25);
Query OK, 0 rows affected (0.265 sec)
Records: 0 Duplicates: 0 Warnings: 0

MariaDB [galeri_itera]> update pembeli set kota = 'Jakarta' where Id_pembeli = 'C_890';
Query OK, 0 rows affected (0.142 sec)
Rows matched: 0 Changed: 0 Warnings: 0

MariaDB [galeri_itera]> update pembeli set kota = 'Bandung' where Id_pembeli = 'C_901';
Query OK, 1 row affected (0.057 sec)
Rows matched: 1 Changed: 1 Warnings: 0

MariaDB [galeri_itera]> update pembeli set kota = 'Yogyakarta' where Id_pembeli = 'C_991';
Query OK, 1 row affected (0.051 sec)
Rows matched: 1 Changed: 1 Warnings: 0

MariaDB [galeri_itera]> select * from pembeli;
```

Id_pembeli	pembeli_nama	pembeli_kontak	Kota
C_800	Egi	0812521221	NULL
C_810	Ardi	0862145121	NULL
C_901	Rudi	081231511	Bandung
C_991	Andi	0852120211	Yogyakarta

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

4. Tambahkan kolom kota pada tabel supplier kemudian diurutkan.

*Command Line* : **alter table supplier add Kota varchar(25);**

*Command Line* : **update supplier set kota = 'Jakarta' where supplier\_id = 'S001';**

*Command Line* : **update supplier set kota = 'Bandung' where supplier\_id = 'S002';**

*Command Line* : **update supplier set kota = 'Yogyakarta' where supplier\_id = 'S003';**

*Command Line* : **update supplier set kota = 'Bandar Lampung' where supplier\_id = 'S004';**  
**update supplier set kota = 'Jakarta' where supplier\_id = 'S005';**

*Command Line* : **update supplier set kota = 'Yogyakarta' where supplier\_id = 'S006';**

*Command Line* : **select \* from supplier;**

```
XAMPP for Windows - mysql -u root -p
MariaDB [galeri_itera]> alter table supplier add Kota varchar(25);
Query OK, 0 rows affected (0.322 sec)
Records: 0 Duplicates: 0 Warnings: 0

MariaDB [galeri_itera]> update supplier set kota = 'Jakarta' where supplier_id = 'S001';
Query OK, 1 row affected (0.170 sec)
Rows matched: 1 Changed: 1 Warnings: 0

MariaDB [galeri_itera]> update supplier set kota = 'Bandung' where supplier_id = 'S002';
Query OK, 1 row affected (0.342 sec)
Rows matched: 1 Changed: 1 Warnings: 0

MariaDB [galeri_itera]> update supplier set kota = 'Yogyakarta' where supplier_id = 'S003';
Query OK, 1 row affected (0.054 sec)
Rows matched: 1 Changed: 1 Warnings: 0

MariaDB [galeri_itera]> update supplier set kota = 'Bandar Lampung' where supplier_id = 'S004'; update supplier set kota = 'Jakarta' w
Query OK, 1 row affected (0.067 sec)
Rows matched: 1 Changed: 1 Warnings: 0

Query OK, 1 row affected (0.020 sec)
Rows matched: 1 Changed: 1 Warnings: 0

MariaDB [galeri_itera]> update supplier set kota = 'Yogyakarta' where supplier_id = 'S006';
Query OK, 1 row affected (0.044 sec)
Rows matched: 1 Changed: 1 Warnings: 0
```

```
XAMPP for Windows - mysql -u root -p
MariaDB [galeri_itera]> select * from supplier;
+-----+-----+-----+-----+
| Supplier_id | Company_nama | Nama_Kontak | Kota |
+-----+-----+-----+-----+
| S001        | Semua Terang | Sugiono     | Jakarta |
| S002        | Suka Maju    | Rahmat      | Bandung |
| S003        | Maju Terus   | Dayono      | Yogyakarta |
| S004        | Pelita Baru  | Puspa       | Bandar Lampung |
| S005        | Surya Kun    | Siti        | Jakarta |
| S006        | Ceria Kasih  | Topan       | Yogyakarta |
+-----+-----+-----+-----+
6 rows in set (0.001 sec)
```

5. Tampilkan nilai maksimal pada kolom jumlah stok.

*Command Line* : **select max(jumlah\_stok) from produk;**

```
MariaDB [galeri_itera]> select max(jumlah_stok) from produk;
+-----+
| max(jumlah_stok) |
+-----+
| 80PCS            |
+-----+
1 row in set (0.059 sec)
```

6. Tampilkan nilai rata-rata dari tabel produk pada tabel jumlah stok

*Command Line* : **select avg(jumlah\_stok) from produk;**

```
MariaDB [galeri_itera]> select avg(jumlah_stok) from produk;
+-----+
| avg(jumlah_stok) |
+-----+
| 166.6666666666666 |
+-----+
1 row in set, 15 warnings (0.033 sec)
```

## **BAB III**

### **KESIMPULAN**

---

Kesimpulannya, Agregasi berfungsi untuk mendapatkan nilai tertentu dari data yang telah dikelompokkan seperti nilai max, min, count, sum, avg, dll. Grouping digunakan untuk mengelompokkan suatu data berdasarkan satu kolom yang diinginkan pada suatu tabel seperti order by, group by, having, dll. Dan Fungsi agregasi dan grouping dapat digunakan secara bersamaan, group by secara umum hanya akan mengambil data paling awal jika terdapat lebih dari satu data grouping dengan nilai yang sama. group by dapat digunakan lebih dari satu kolom acuan.



## **DAFTAR PUSTAKA**

Modul 7 Praktikum Basis Data