

LATIHAN MODUL 6
LAPORAN PRAKTIKUM BASIS DATA
“AGREGASI DAN GROUPING”



Disusun oleh:

Nama : Antonius Munthe

NIM : 121140032

Kelas : Basis Data RA

PROGRAM STUDI TEKNIK INFORMATIKA
INSTITUT TEKNOLOGI SUMATERA
LAMPUNG SELATAN
2023

BAB I

TEORI DASAR

A. FUNGSI AGREGASI DAN GROUPING

1. AGREGASI

Agregasi adalah sebuah teknik dalam SQL yang digunakan untuk mendapatkan nilai tertentu dari data yang telah dikelompokkan. Pengelompokan data dapat didasarkan pada satu kolom/kombinasi dari beberapa kolom yang dipilih. Berikut ini fungsi agregasi:

1. MAX, mencari data terbesar dari sekelompok data.
2. MIN, mencari data terkecil dari sekelompok data.
3. COUNT, mencari cacah data dari sekelompok data.
4. SUM, mencari jumlah dari sekumpulan data numerik.
5. AVG, mencari nilai rerata dari sekumpulan data numerik.

2. GROUPING

fungsi untuk mengelompokkan suatu data berdasarkan satu field/kolom yang diperlukan pada suatu table. Berikut ini adalah fungsi grouping:

1. ORDER BY, Digunakan untuk menampilkan data secara terurut berdasarkan nilai tertentu. Ada dua jenis yaitu ASCENDING dan DESCENDING.
2. GROUP BY, Digunakan untuk mengelompokkan beberapa data pada perintah SELECT.
3. HAVING, Fungsi ini hampir sama dengan fungsi WHERE, hanya saja WHERE tidak dapat digunakan dengan fungsi agregasi.

BAB II

PEMBAHASAN DAN ANALISIS

1. Buatlah database dengan nama “Akademik”

Kode di bawah adalah kode SQL yang digunakan untuk membuat database "Akademik"

```
MariaDB [(none)]> create database akademik;
Query OK, 1 row affected (0.009 sec)

MariaDB [(none)]> use akademik;
Database changed
```

2. Buatlah tabel dengan nama “mahasiswa”

Kode di bawah adalah kode SQL yang digunakan untuk membuat tabel "mahasiswa" dengan struktur kolom yang telah ditentukan.

```
MariaDB [akademik]> create table mahasiswa(
  -> nim char(3),
  -> kodemk varchar(5),
  -> thnakademik char(9),
  -> nilai char(1) NULL,
  -> bobot int(2));
Query OK, 0 rows affected (0.016 sec)
```

3. Masukkan data berikut ke dalam tabel “ Mahasiswa”

Kode di bawah adalah kode SQL yang digunakan untuk Input data ke dalam tabel menggunakan perintah "INSERT INTO" dengan nilai-nilai yang telah ditentukan.

```
MariaDB [akademik]> INSERT INTO mahasiswa VALUES
  -> ("123", "SMBD2", "2013/2014", "C", 4),
  -> ("123", "SMBD2", "2014/2015", "A", 4),
  -> ("123", "SIBW", "2014/2015", "A", 4),
  -> ("123", "DMEP", "2014/2015", "B", 2),
  -> ("456", "DMEP", "2014/2015", "A", 2),
  -> ("456", "SIBW", "2011/2012", "C", 4),
  -> ("456", "SIBW", "2012/2013", "C", 4),
  -> ("456", "SIBW", "2014/2015", "A", 4),
  -> ("789", "SMBD2", "2011/2012", "D", 4),
  -> ("789", "SMBD2", "2012/2013", "C", 4),
  -> ("789", "SMBD2", "2013/2014", "C", 4),
  -> ("789", "SMBD2", "2014/2015", "A", 4);
Query OK, 12 rows affected (0.008 sec)
Records: 12 Duplicates: 0 Warnings: 0
```

```
MariaDB [akademik]> select * from mahasiswa;
+-----+-----+-----+-----+-----+
| nim | kodemk | thnakademik | nilai | bobot |
+-----+-----+-----+-----+-----+
| 123 | SMBD2 | 2013/2014 | C | 4 |
| 123 | SMBD2 | 2014/2015 | A | 4 |
| 123 | SIBW | 2014/2015 | A | 4 |
| 123 | DMEP | 2014/2015 | B | 2 |
| 456 | DMEP | 2014/2015 | A | 2 |
| 456 | SIBW | 2011/2012 | C | 4 |
| 456 | SIBW | 2012/2013 | C | 4 |
| 456 | SIBW | 2014/2015 | A | 4 |
| 789 | SMBD2 | 2011/2012 | D | 4 |
| 789 | SMBD2 | 2012/2013 | C | 4 |
| 789 | SMBD2 | 2013/2014 | C | 4 |
| 789 | SMBD2 | 2014/2015 | A | 4 |
+-----+-----+-----+-----+-----+
12 rows in set (0.000 sec)
```

4. Tampilkan semua beban SKS yang telah ditempuh oleh mahasiswa dengan nim 456, termasuk untuk matakuliah yang pernah diulang.

Kita menggunakan fungsi agregasi SUM untuk menjumlahkan semua bobot matakuliah yang telah ditempuh oleh mahasiswa dengan nim 456.

```
MariaDB [akademik]> select bobot from mahasiswa where nim = 456;
+-----+
| bobot |
+-----+
|      2 |
|      4 |
|      4 |
|      4 |
+-----+
4 rows in set (0.000 sec)
```

5. Tampilkan nilai terbaik yang didapatkan oleh seorang mahasiswa untuk mata kuliah tertentu.

Kita mengelompokkan data berdasarkan kodemk dan menggunakan fungsi agregasi MAX untuk menampilkan nilai terbaik yang didapatkan oleh mahasiswa dengan nim 123 untuk setiap matakuliah.

```
MariaDB [akademik]> select nim, kodemk, MIN(nilai) from mahasiswa group by nim, kodemk;
+-----+-----+-----+
| nim | kodemk | MIN(nilai) |
+-----+-----+-----+
| 123 | DMEP   | B           |
| 123 | SIBW   | A           |
| 123 | SMBD2  | A           |
| 456 | DMEP   | A           |
| 456 | SIBW   | A           |
| 789 | SMBD2  | A           |
+-----+-----+-----+
6 rows in set (0.001 sec)
```

6. Tampilkan matakuliah beserta nilai yang terburuk yang pernah didapatkan oleh mahasiswa dengan nim 123.

Kita mengelompokkan data berdasarkan kodemk dan menggunakan fungsi agregasi MIN untuk menampilkan nilai terburuk yang didapatkan oleh mahasiswa dengan nim 123 untuk setiap matakuliah.

```
MariaDB [akademik]> select kodemk, MAX(nilai) from mahasiswa where nim = 123;
+-----+-----+
| kodemk | MAX(nilai) |
+-----+-----+
| SMBD2  | C           |
+-----+-----+
1 row in set (0.001 sec)
```

7. Tampilkan jumlah cacah nilai yang pernah diberikan untuk matakuliah tertentu.

Kita menggunakan fungsi agregasi COUNT(*) untuk menghitung jumlah data yang memiliki kombinasi nilai dan kodemk yang sama. Kemudian, kita grupkan berdasarkan kolom kodemk dan nilai.

```
MariaDB [akademik]> select kodemk, COUNT(nilai) from mahasiswa group by kodemk;
```

kodemk	COUNT(nilai)
DMEP	2
SIBW	4
SMBD2	6

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

8. Tampilkan seluruh mahasiswa yang pernah mengulang sebuah matakuliah.

Kita menggunakan fungsi COUNT(DISTINCT thnakademik) untuk menghitung jumlah tahun ajaran yang pernah diambil oleh mahasiswa untuk setiap kombinasi nim dan kodemk. Kita menggunakan WHERE untuk memfilter hanya data dengan nilai C, D, atau E. Kemudian, kita grupkan berdasarkan nim dan kodemk. Terakhir, kita menggunakan HAVING untuk memfilter data yang jumlah tahun ajarannya lebih dari 1, yang berarti mahasiswa pernah mengulang matakuliah tersebut.

```
MariaDB [akademik]> select nim from mahasiswa group by nim having count(kodemk) > 1;
```

nim
123
456
789

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

9. Tampilkan semua table dimana fieldnya terdiri dari nim, kodemk, thnakademik, nilai dimana nilai lebih kecil dari C.

Kita menggunakan WHERE untuk memfilter data dengan nilai kurang dari C

```
MariaDB [akademik]> select * from mahasiswa group by nilai having (nilai)>"C";
```

nim	kodemk	thnakademik	nilai	bobot
789	SMBD2	2011/2012	D	4

```
1 row in set (0.007 sec)
```

BAB III

KESIMPULAN

Berdasarkan modul praktikum SQL agregasi dan grouping, dapat disimpulkan bahwa:

1. Agregasi adalah proses penghitungan nilai-nilai statistik dari sekelompok data, seperti rata-rata, jumlah, atau nilai maksimum.
2. SQL menyediakan beberapa fungsi agregasi bawaan, seperti SUM, AVG, COUNT, MAX, dan MIN.
3. Fungsi-fungsi agregasi dapat digunakan bersamaan dengan pernyataan GROUP BY untuk mengelompokkan data berdasarkan kriteria tertentu sebelum melakukan agregasi.
4. SQL juga menyediakan fungsi GROUP BY, HAVING untuk memfilter kelompok data yang memenuhi kriteria tertentu setelah dilakukan agregasi.

DAFTAR PUSTAKA

1. Silberschatz, A., Korth, H. F., & Sudarshan, S. (2010). Database System Concepts. McGraw-Hill Education.
2. Date, C. J. (2004). An Introduction to Database Systems (8th ed.). Addison-Wesley.
3. Jukic, N., Vrbsky, S. V., & Nestorov, S. (2014). SQL Programming, Part 2: Aggregate Functions, Grouping, and Subqueries. IEEE Potentials, 33(5), 32-36.
4. MySQL. (n.d.). MySQL Documentation: Aggregate (GROUP BY) Functions. Retrieved May 12, 2023, from <https://dev.mysql.com/doc/refman/8.0/en/group-by-functions.html>