|  |  |  |
| --- | --- | --- |
|  | **DATA MANIPULATION LANGUAGE**  **(DML)** | BASISDATA |
| Nama : Qusnul Qotimah |
| Instruktur :  Budi Sulistiyo, S. Kom., M. Kom. | Kelas : XI RPL C |
| Tanggal : 27 Januari 2022 |

1. **Indikator Pencapaian Kompetensi:**
2. Siswa mampu menjalankan MySQL di XAMPP
3. Siswa mampu menggunakan fitur-fitur yang terdapat di XAMPP
4. Siswa mampu membuat database dan tabel.
5. **Tujuan Pembelajaran:**

Setelah melaksanakan pembelajaran diharapkan siswa mampu :

1. Membuat database dan tabel dengan data definition language
2. Mampu memodifikasi tabel
3. **Alat dan Bahan**
   1. PC
   2. Software XAMPP
4. **DASAR TEORI**

Desain Basis Data memiliki banyak sekali perintah yang bisa kita gunakan untuk mendesain Basis Data tersebut. Intinya kita diberikan banyak sekali perintah dan operator yang bisa kita gunakan agar basis data yang kita buat lebih mudah untuk kita kases lagi. Disini kita akan membahas sedikit tentang Agregasi dan View

**1. AGREGASI**

Agregasi adalah salah satu perintah yang bisa digunakan untuk melakukan penghitungan menjadi sebuah nilai dari beberapa nilai input. Aggregate dapat digabungkan dengan sebuah parameter seperti WHERE untuk menghasilkan suatu hasil yang lebih kompleks lagi. Ada beberaapa fungsi agregasi yang di sediakan ;

* COUNT(\*) untuk Menghitung jumlah baris
* SUM(NAMA KOLOM) untuk Menghitung penjumlahan data
* MAX(NAMA\_KOLOM) untuk Mencari nilai terbesar
* MIN(NAMA KOLOM) untuk Mencari nilai terkecil
* AVG(NAMA KOLOM) untuk Mencari nilai rata-rata

Untuk menggunakan peritah yang mengandung agregasi kita bisa menggunakan perintah SELECT, berikut query dasar perintah agregasi :

* Select count(\*) from namatabel;
* Select max(namakolom) from namatabel;
* Select min(namakolom) from namatabel;
* Select avg(namakolom) from namatabel;

**2.  GROUP BY**

Group by ini digunakan untuk mengelompokkan kolom tertentu berdasarkan perintah SELECT. Group by ini sering digunakan bersama dengan Agregasi, digunakan sebagai pelengkap perintah yang lebih kompleks dan spesifik. Query dasar yang bisa digunakan :

* Select namakolom from namatabel group by namakolom;

Berikut ini query yang diunakan bersama dengan salah satu perintah Agregasi

* Select namakolom, count(\*)  from namatabel group by namakolom; //menggunakan perintah agregasi count

**3.  HAVING**

Pemakaian HAVING bersama dengan GROUP BY, kegunaanya adalah untuk menentukan kondisi bagi GROUP BY, dimana kelompok yang memenuhi kondisi saja yang akan di hasilkan. Jadi apabila pada perintah SELECT yang digunakan untk menentukan kondisi adalah WHERE, tapi untuk perintah GROUP BY ini yang digunakan adalah HAVING. Contoh Query yang menggunakan perintah HAVING :

* Select namakolom, sum(namakolom)  from namatabel group by namakolom having sum(namakolom) = kondisi;

**4.  CASE**

Walaupun SQL bukanlah bahasa perograman, tapi dalam prosesnya kita dapat bebas mengontrol data untuk kita tampilkan kembali. Dalam perintah SELECT kita mengenal perintah WHERE  yang digunakan menentukan kondisi dari data yang ingin kita tampilkan, sedangkan CASE perbandingan dalam bentuk output kolom. Jadi intinya penggunaan CASE untuk menampilkan data tersendiri berupa kolom baru dengan data dari kolom sebelumnya yang kita manipilasi isisnya.Struktur didalam Select seperti berikut :

CASE WHEN condition THEN result  
[WHEN …]  
[ELSE result] END

**5. VIEW**

Views merupakan tabel duplikasi tetapi bukan temporary table, bukan juga merupakan sebuah tabel yang asli, tapi view ini cuma menduplikasi tabel awal dengan adanya modifikasi. Suatu view adalah suatu relasi virtual yang tidak perlu ada database tetapi dapat diproduksi atas permintaan oleh pemakai tertentu, pada ketika permintaan. Satu lagi kelebihan yang dimiliki oleh view yaitu dapat menyimpan perintah query, dan dapat mewakili sebuah subset dari tabel asli dan memilih kolom atau row tertentu dari tabel biasa. Berbeda dengan temporary tabel, view ini tidak hilang ketika kita ingin mengaksesnya lagi. Berikut query dasar untuk menampilkan view :

* create view nama\_tabel\_view as query;

1. **B. PRAKTIKUM**

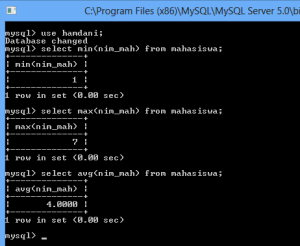
Baiklah, sekarang langsung saja kita praktekan materi tentang agregasi dan view diatas. Disini kita akan menggunakan DBMS MySQL seperti teori di atas.

1. Penguaan Perintah Agregasi

Bagi anda yang selalu mengikuti artikel saya, bukalah kembali tabel mahasiswa dan fakultas pada bahasan artikel sebelumnya [( Baca : Perintah Dasar SQL )](https://hamdanilampuhias.wordpress.com/2014/10/02/perintah-dasar-sql/). Dari tabel tersebut cari nim yang paling kecil, paling tinggi dan rata rata dari nim yang ada.

# Berikut query yang bisa kita tuliskan untuk menyelesaikan soal diatas

* mysql> Select min(nim\_mah) from mahasiswa;
* mysql> Select max(nim\_mah) from mahasiswa;
* mysql> Select avg(nim\_mah) from mahasiswa;

[](https://hamdanilampuhias.files.wordpress.com/2014/10/15.png)

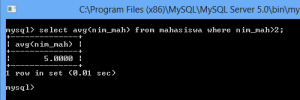
Perintah Agregasi, Praktikum Desain Basis Data

2. Menampilkan Rata-Rata NIM

Menampilkan rata-rata NIM ynag lebih dari 2.

# Berikut query yang bisa kita tuliskan untuk menyelesaikan soal diatas

* mysql> Select avg(nim\_mah) from mahasiswa where nim\_mah>2;

[](https://hamdanilampuhias.files.wordpress.com/2014/10/22.png)

Perintah Agregasi, Praktikum Desain Basis Data,

3. Modifikasi Perintah GROUP BY

Apabila pada dasar teori kita diberikan dasar query dari perintah group  by, maka disini kita akan mencoba query yang lebih kompleks dari perintah group by. Menampilkan jumlah mahasiswa berdasarkan fakultas. Sehingga akan menghasilkan seperti ini ;

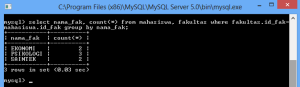
Nama Fakultas         COUNT(\*)

SAINTEK                          2

PSIKOLOGI                      1

# Berikut query yang bisa kita tuliskan untuk menyelesaikan soal diatas

* mysql> Select nama\_fak, count(\*) from mahasiswa, fakultas where fakultas.id\_fak=mahasiswa.id\_fak group by nama\_fak;

[](https://hamdanilampuhias.files.wordpress.com/2014/10/32.png)

Perintah GROUP BY, Desain Basis Data

4. Modifikasi Perintah HAVING

Menampilkan data sama seperti nomor 3 dengan persyaratan jumlah mahasiswa yang lebih dari 2 saja yang ditampilkan.

# Berikut query yang bisa kita tuliskan untuk menyelesaikan soal diatas

* mysql> Select nama\_fak, count(\*) from mahasiswa, fakultas where fakultas.id\_fak=mahasiswa.id\_fak group by nama\_fak having count(\*)>2;

[](https://hamdanilampuhias.files.wordpress.com/2014/10/42.png)

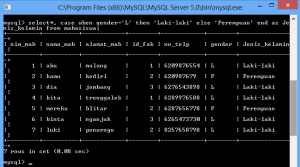
Perintah HAVING, Desain Basis Data

5. Modifikasi Perintah CASE

Mungkin bagi yang melihat dasar teori diatas, pada kutipan SQL dasar dari perintah CASE akan bingung dengan bagaimana urutannya, disini kita akan langsung mempraktekannya. Menampilkan data mahasiswa dengan persyaratan, jika jenis kelaminnya “L” maka tertulis laki-laki dan bila “P” maka tertulis perempuan.

# Berikut query yang bisa kita tuliskan untuk menyelesaikan soal diatas

* mysql> Select\*, case when gender=’L’ then ‘Laki-laki’ else ‘Perempuan’ end as Jenis\_kelamin from mahasiswa;

[](https://hamdanilampuhias.files.wordpress.com/2014/10/52.png)

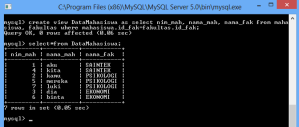
Perintah Case, Desain Basis Data

6. Perintah VIEW

Membuat view untuk query penampilan data mahasiswa, fakultas. Ambil berdasarkan nim, nama mahasiswa, nama fakultas.

# Berikut query yang bisa kita tuliskan untuk menyelesaikan soal diatas

* mysql> create view DataMahasiswa as Select nim\_mah, nama\_mah, nama\_fak from mahasiswa, fakultas where mahasiswa.id\_fak=fakultas.id\_fak;

[](https://hamdanilampuhias.files.wordpress.com/2014/10/61.png)

1. **C. PERBEDAAN, KESAMAAN, KELEBIHAN DAN KEKURANGAN PADA DBMS POSTGRESQL DAN MYSQL**

Seperti biasanya, pada bahasan ketiga kita akan membahas tentang perbedaan antara dua DBMS, yaitu MySQL dan PostgreSQL. Sebetulnya kenapa sih kita perlu membahas ini, sepertinya kita cukup tahu MySQL aja sudah. Sebenarnya tujuan kami membahas tentang perbedaan ini adalah, agar para pembaca bisa mengetahui sejauh mana perkembangan kedua DBMS ini. Sehingga ketika kita bingung untuk menggunkan DBMS apa, kita bisa memutuskannya dengan melihat perbedaan antara keduanya. Pada perintah Agregasi dan View ini kedua DBMS sama sam memilikinya, dan SQL dari kedua DBMS ini juga sama. Tetapi ada beberapa ketelitian yang harus kita perhatikan dalam menuliskan query pada kedua DBMS ini, antara lain yaitu pemberian tanda koma (,) seperti pada perintah COUNT(\*) dan CASE. Yang biasanya setelah perintah SELECT tidak ada koma, untuk penggunaan perintah COUNT(\*) dan CASE kita harus memeberikannya. Kemudian pada perintah agregasi seperti MIN, MAX dan AVG. Apabila kita salah dalam memberikan spasi pada query tersebut juga akan memberikan efek.

**D. KESIMPULAN dan SARAN**

Pada artikel sebelumnya kita telah mengetahui beberapa perintah atau operator lanjutan dari perintah SELECT. Pada bahasan artikel ini telah dibahas juha lanjutan perintah SELECT dengan tambahan perintah yang lebih kompleks. Kita bisa menggunakan DBMS MySQL maupun PostgreSQL untuk mempraktekan perintah Agregasi dan View ini, karena keduanya memiliki struktur query yang sama. Tapi saran saya sebagai penulis, ketika menuliskan query utamanya perintah Agregasi dan View ini harap bisa lebih teliti dan hati hati. Karena banyak sekali faktor faktor kecil yang bisa membuat salah dan terjadi ERrOR.