

MODUL BASIS DATA

AGREGASI , GROUPING, JOIN

Garis besar modul:

1. Perintah Agregasi
2. Perintah Grouping
3. Perintah Join



<http://creativecommons.org/licenses/by-nc-sa/3.0/>

Modul ini bebas dicopy, didistribusikan, ditransmit dan diadaptasi/modifikasi/diremik dengan syarat tidak untuk komersial, pembuat asal tetap dicantumkan dan hasil modifikasi dishare dengan lisensi yang sama.

AGREGASI , GROUPING, JOIN

Setelah minggu kemarin kita belajar mengenai DML atau yang biasa disebut dengan Data Manipulation Language, sekarang kita akan belajar mengenai Agregasi dan Grouping pada SQL.

Agregasi merupakan fungsi ringkasan yang 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.

Grouping merupakan sekumpulan perintah untuk mengelompokkan data sesuai dengan kebutuhan pengguna.

1. Agregasi

Beberapa DBMS memiliki fungsi agregat, yaitu fungsi-fungsi khusus yang melibatkan sekelompok data (agregat). Secara umum fungsi agregat adalah:

- SUM untuk menghitung total nominal data
- COUNT untuk menghitung jumlah kemunculan data
- AVG untuk menghitung rata-rata sekelompok data
- MAX dan MIN untuk mendapatkan nilai maksimum/minimum dari sekelompok data.

Fungsi agregat digunakan pada bagian SELECT. Syarat untuk fungsi agregat diletakkan pada bagian **HAVING**, bukan WHERE.

Untuk contoh penerapannya kita akan menggunakan empat table pada pertemuan kemarin, yaitu table mahasiswa, dosen, matakuliah dan nilai.

Isikan masing-masing table tersebut seperti gambar dibawah ini:

Tabel

```
mysql> show tables;
+-----+
| Tables_in_basdat |
+-----+
| dosen             |
| mahasiswa         |
| matkul            |
| nilai             |
+-----+
4 rows in set (0.09 sec)
```

Mahasiswa

nim	nama	alamat	usia	angkatan
1005333	Mustamiin	Cirebon	20	2010
1005344	Indana	Bandung	19	2010
1005355	Anshar	Bandung	20	2010
1005366	Husein	Bandung	19	2010
1005377	Hasan	Bandung	19	2010

5 rows in set (0.00 sec)

Dosen

nip	nama_dosen	alamat	gelar
1001	Bu Rosa	Bandung	S2
1002	Pak Munir	Bandung	Doktor
1003	Pak Budi	Solo	S2
1004	Pak Herbert	Medan	S2
1005	Bu Novi	Bandung	S2

5 rows in set (0.00 sec)

Matkul

kode_matkul	nip	nama_matkul	semester
IK-001	1001	Alpro2	3
IK-002	1002	Konsep Teknologi	1
IK-003	1002	MPPL	5
IK-004	1003	Basis Data	3
IK-005	1005	Strukdat	5
IK-006	1004	Disweb	3
IK-007	1003	Provis	3

7 rows in set (0.00 sec)

Nilai

kode_nilai	nim	kode_matkul	nilai
N-001	1005377	IK-001	100
N-002	1005377	IK-002	100
N-003	1005377	IK-003	100
N-004	1005377	IK-004	100
N-005	1005377	IK-005	100
N-006	1005377	IK-006	100
N-007	1005377	IK-007	100
N-008	1005366	IK-001	80
N-009	1005366	IK-002	90
N-010	1005366	IK-006	70
N-011	1005366	IK-004	75
N-012	1005366	IK-007	85
N-013	1005355	IK-001	90
N-014	1005355	IK-002	80
N-015	1005355	IK-004	85
N-016	1005355	IK-005	100
N-017	1005355	IK-006	95
N-018	1005355	IK-007	70
N-019	1005344	IK-001	90
N-020	1005344	IK-002	60
N-021	1005344	IK-004	85
N-022	1005344	IK-005	75
N-023	1005344	IK-006	90
N-024	1005344	IK-007	80
N-025	1005333	IK-001	65
N-026	1005333	IK-002	95
N-027	1005333	IK-003	50
N-028	1005333	IK-004	60
N-029	1005333	IK-005	70
N-030	1005333	IK-006	80
N-031	1005333	IK-007	90

31 rows in set (0.00 sec)

Kemudian kita akan belajar menggunakan fungsi agregasi :

SUM

Digunakan untuk menghitung jumlah total nilai pada suatu tabel

Syntax umum:

```
Select sum(nama_field) from nama_tabel;
```

Contoh:

```
Select sum(nilai) from nilai;
```

```
mysql> Select sum(nilai) from nilai;
+-----+
| sum(nilai) |
+-----+
|         2610 |
+-----+
1 row in set (0.00 sec)
```

COUNT

Digunakan untuk menghitung banyaknya data pada suatu tabel

Syntax umum:

```
Select count(nama_field) from nama_tabel;
```

Contoh:

```
Select count(nama) from mahasiswa;
```

```
mysql> select count(nilai) from nilai;
+-----+
| count(nilai) |
+-----+
|           31 |
+-----+
1 row in set (0.00 sec)
```

AVG

Digunakan untuk menghitung nilai rata-rata dari suatu nilai pada table

Syntax umum:

```
Select avg (nama_field) from nama_tabel;
```

Contoh:

```
Select avg(nilai) from nilai;
```

```
mysql> Select avg(nilai) from nilai;
+-----+
| avg(nilai) |
+-----+
|    84.1935 |
+-----+
1 row in set (0.04 sec)
```

MAX dan MIN

Fungsi agregasi max adalah kebalikan dari fungsi agregasi min, dimana data yang ditampilkannya adalah data yang terbesar

Syntax umum nya :

```
Select max(nama_field) from nama_tabel;
```

Contoh:

```
Select max(nilai) from nilai;
```

```
mysql> Select max(nilai) from nilai;
+-----+
| max(nilai) |
+-----+
|         100 |
+-----+
1 row in set (0.05 sec)
```

Fungsi agregasi min digunakan untuk menampilkan satu nilai yang terkecil di dalam tabel

Syntax umum nya:

```
Select min(nama_field) from nama_tabel;
```

Contoh :

```
Select min(nilai) from nilai;
```

```
mysql> Select min(nilai) from nilai;
+-----+
| min(nilai) |
+-----+
|          50 |
+-----+
1 row in set (0.00 sec)
```

Alias

Meskipun bukan hanya untuk agregasi saja, tapi fungsi ini wajib dipelajari juga.

Syntax umum nya:

```
Select (fungsi agregasi) as nama_lain from nama_tabel;
```

Contoh :

```
Select avg(nilai) as rata2 from nilai;
```

```
mysql> Select avg(nilai) as rata2 from nilai;
+-----+
| rata2 |
+-----+
| 84.1935 |
+-----+
1 row in set (0.04 sec)
```

2. GROUPING

Ada beberapa jenis grouping, yaitu order by, group by, having, dan view.

Order by

Order by merupakan perintah yang ada di sql untuk mengurutkan data berdasarkan ascending ataupun descending.

Syntax umum nya:

```
Select *from nama_tabel order by atribut asc/desc;
```

Contoh :

```
Select *from mahasiswa order by nim asc;
```

```
mysql> Select *from mahasiswa order by nim asc;
+----+-----+-----+-----+-----+
| nim   | nama   | alamat | usia | angkatan |
+----+-----+-----+-----+-----+
| 1005333 | Mustamiin | Cirebon | 20 | 2010 |
| 1005344 | Indana   | Bandung | 19 | 2010 |
| 1005355 | Anshar   | Bandung | 20 | 2010 |
| 1005366 | Husein   | Bandung | 19 | 2010 |
| 1005377 | Hasan    | Bandung | 19 | 2010 |
+----+-----+-----+-----+-----+
5 rows in set (0.00 sec)
```

Group by

Group By merupakan fungsi yang digunakan untuk melakukan pengelompokan dari perintah SELECT. Group by seringkali diperlukan untuk menjalankan agregate menjadi sebuah kelompok dari hasil Query.

Syntax umum nya:

```
Select *from nama_tabel group by atribut;
```

Contoh :

```
Select *from nilai group by nim;
```

```
mysql> Select *from nilai group by nim;
+-----+-----+-----+-----+
| kode_nilai | nim   | kode_matkul | nilai |
+-----+-----+-----+-----+
| N-025      | 1005333 | IK-001      | 65    |
| N-019      | 1005344 | IK-001      | 90    |
| N-013      | 1005355 | IK-001      | 90    |
| N-008      | 1005366 | IK-001      | 80    |
| N-001      | 1005377 | IK-001      | 100   |
+-----+-----+-----+-----+
5 rows in set (0.01 sec)
```

Having

Pemakaian HAVING terkait dengan GROUP BY, kegunaanya adalah untuk menentukan kondisi bagi GROUP BY, dimana kelompok yang memenuhi kondisi saja yang akan di hasilkan.

Statement HAVING telah ditambahkan ke SQL karena keyword WHERE tidak dapat digunakan dengan fungsi agregat.

Bentuk umum HAVING:

```
SELECT nama_kolom, aggregate_function (nama_kolom)
FROM nama_tabel
WHERE nama_kolom operator nilai
GROUP BY nama_kolom
HAVING aggregate_function (nama_kolom) Nilai operator
```

Contoh :

```
Select nim, kode_matkul, nilai from nilai group by nim,
kode_matkul having avg(nilai)>85;
```

```
mysql> Select nim, kode_matkul, nilai from nilai group by nim, kode_matkul havin
g avg(nilai)>85;
+-----+-----+-----+
| nim      | kode_matkul | nilai |
+-----+-----+-----+
| 1005333  | IK-002      | 95    |
| 1005333  | IK-007      | 90    |
| 1005344  | IK-001      | 90    |
| 1005344  | IK-006      | 90    |
| 1005355  | IK-001      | 90    |
| 1005355  | IK-005      | 100   |
| 1005355  | IK-006      | 95    |
| 1005366  | IK-002      | 90    |
| 1005377  | IK-001      | 100   |
| 1005377  | IK-002      | 100   |
| 1005377  | IK-003      | 100   |
| 1005377  | IK-004      | 100   |
| 1005377  | IK-005      | 100   |
| 1005377  | IK-006      | 100   |
| 1005377  | IK-007      | 100   |
+-----+-----+-----+
15 rows in set (0.04 sec)
```

View

DBMS menyediakan suatu mekanisme untuk menyembunyikan detail data tertentu.

View merupakan sebuah tabel semu / tabel logik, dimana datanya berasal dari satu atau lebih tabel lain yang disebut sebagai tabel sumber. View biasa dibuat untuk memudahkan user menampilkan data.

Untuk membuat view, gunakan perintah berikut:

create view v as <query expression>

dimana: – v adalah nama view

Contoh :

```
Create view nilai_total as select a.nama, b.nama_matkul,
c.nama_dosen, d.nilai
from mahasiswa a, matkul b, dosen c, nilai d
where d.nim=a.nim and d.kode_matkul=b.kode_matkul and
b.nip=c.nip;
```

```
mysql> select *from nilai_total;
```

nama	nama_matkul	nama_dosen	nilai
Mustamiin	Alpro2	Bu Rosa	65
Mustamiin	Konsep Teknologi	Pak Munir	95
Mustamiin	MPPL	Pak Munir	50
Mustamiin	Basis Data	Pak Budi	60
Mustamiin	Strukdat	Bu Novi	70
Mustamiin	Disweb	Pak Herbert	80
Mustamiin	Provis	Pak Budi	90
Indana	Alpro2	Bu Rosa	90
Indana	Konsep Teknologi	Pak Munir	60
Indana	Basis Data	Pak Budi	85
Indana	Strukdat	Bu Novi	75
Indana	Disweb	Pak Herbert	90
Indana	Provis	Pak Budi	80
Anshar	Alpro2	Bu Rosa	90
Anshar	Konsep Teknologi	Pak Munir	80
Anshar	Basis Data	Pak Budi	85
Anshar	Strukdat	Bu Novi	100
Anshar	Disweb	Pak Herbert	95
Anshar	Provis	Pak Budi	70
Husein	Alpro2	Bu Rosa	80
Husein	Konsep Teknologi	Pak Munir	90
Husein	Basis Data	Pak Budi	75
Husein	Disweb	Pak Herbert	70
Husein	Provis	Pak Budi	85
Hasan	Alpro2	Bu Rosa	100
Hasan	Konsep Teknologi	Pak Munir	100
Hasan	MPPL	Pak Munir	100
Hasan	Basis Data	Pak Budi	100
Hasan	Strukdat	Bu Novi	100
Hasan	Disweb	Pak Herbert	100
Hasan	Provis	Pak Budi	100

31 rows in set (0.00 sec)

LATIHAN

1. Tampilkan nama mahasiswa yang berawalan dengan huruf h
2. Tampilkan nama mahasiswa yang hanya mengontrak mata kuliah MMPL dan strukdat
3. Tampilkan jumlah nilai total setiap mahasiswa
4. Tampilkan nilai rata-rata tiap mahasiswa
5. Tampilkan mata kuliah yang memberikan nilai rata-rata terbesar

3. JOIN

Adalah salah satu syntaks SQL yang berfungsi untuk menghubungkan dua tabel atau merelasikan beberapa tabel yang berbeda untuk mengambil beberapa fields yang dibutuhkan berdasarkan hubungan antar field tertentu dalam table atau bisa disebut dengan **foreign key**. Ada beberapa teknik join pada SQL, diantaranya :

1. Inner Join

Inner join merupakan jenis join yang paling umum yang dapat digunakan pada semua database. Jenis ini dapat digunakan bila ingin merelasikan dua set data yang ada di tabel, letak relasinya setelah pada perintah ON pada join.

Syntax umum :

```
SELECT nama_field
FROM nama_tabel1 INNER JOIN nama_tabel2
ON nama_tabel1.nama_field=nama_tabel2.nama_field
```

Bisa juga tidak memakai kata inner pada join di atas, sehingga querynya menjadi:

```
SELECT nama_field
FROM nama_tabel1 JOIN nama_tabel2 ON
nama_tabel1.nama_field=nama_tabel2.nama_field
```

contoh :

- **Dengan memakai kata inner join**

```
select matkul.nama_matkul, dosen.nama_dosen
from matkul inner join dosen
on matkul.nip =dosen.nip;
```



```
mysql> select matkul.nama_matkul, dosen.nama_dosen
-> from matkul inner join dosen
-> on matkul.nip = dosen.nip;
```

nama_matkul	nama_dosen
Alpro2	Bu Rosa
Konsep Teknologi	Pak Munir
MPPL	Pak Munir
Basis Data	Pak Budi
Provis	Pak Budi
Disweb	Pak Herbert
Strukdat	Bu Novi

7 rows in set (0.00 sec)

- **Dengan tidak memakai kata inner**

```
select matkul.nama_matkul, dosen.nama_dosen
from matkul join dosen
on matkul.nip =dosen.nip;
```

```
mysql> select matkul.nama_matkul, dosen.nama_dosen
-> from matkul join dosen
-> on matkul.nip = dosen.nip;
```

nama_matkul	nama_dosen
Alpro2	Bu Rosa
Konsep Teknologi	Pak Munir
MPPL	Pak Munir
Basis Data	Pak Budi
Provis	Pak Budi
Disweb	Pak Herbert
Strukdat	Bu Novi

```
7 rows in set (0.00 sec)
```

- **Dengan memakai where**

```
select matkul.nama_matkul, dosen.nama_dosen
from matkul, dosen
where matkul.nip =dosen.nip;
```

```
mysql> select matkul.nama_matkul, dosen.nama_dosen
-> from matkul, dosen
-> where matkul.nip = dosen.nip;
```

nama_matkul	nama_dosen
Alpro2	Bu Rosa
Konsep Teknologi	Pak Munir
MPPL	Pak Munir
Basis Data	Pak Budi
Provis	Pak Budi
Disweb	Pak Herbert
Strukdat	Bu Novi

```
7 rows in set (0.00 sec)
```

2. Left Join

Left join akan mengeluarkan semua field pada tabel di kiri(pertama) meskipun pada tabel kanan(kedua) tidak memiliki kesamaan

Syntax umum :

```
SELECT nama_field
FROM nama_tabel1 LEFT JOIN nama_tabel2
ON nama_tabel1.nama_field=nama_tabel2.nama_field
```

Contoh:

```
select matkul.nama_matkul,dosen.nama_dosen
from matkul left join dosen
on matkul.nip = dosen.nip;
```

```
mysql> select matkul.nama_matkul,dosen.nama_dosen
-> from matkul left join dosen
-> on matkul.nip = dosen.nip;
+-----+-----+
| nama_matkul | nama_dosen |
+-----+-----+
| Alpro2      | Bu Rosa   |
| Konsep Teknologi | Pak Munir |
| MPPL        | Pak Munir |
| Basis Data  | Pak Budi  |
| Strukdat    | Bu Novi   |
| Disweb      | Pak Herbert |
| Provis      | Pak Budi  |
+-----+-----+
7 rows in set (0.00 sec)
```

3. Right Join

Right join adalah kebalikan dari left join, dimana right join akan mengeluarkan semua field pada tabel kanan(kedua) meskipun pada tabel kiri(pertama) tidak memiliki pasangannya

Syntax umum :

```
SELECT nama_field FROM nama_tabel1 RIGHT JOIN
nama_tabel2 ON
nama_tabel1.nama_field=nama_tabel2.nama_field
```

Contoh :

```
select matkul.nama_matkul,dosen.nama_dosen
from matkul right join dosen
on matkul.nip = dosen.nip;
```

```
mysql> select matkul.nama_matkul,dosen.nama_dosen
-> from matkul right join dosen
-> on matkul.nip = dosen.nip;
```

nama_matkul	nama_dosen
Alpro2	Bu Rosa
Konsep Teknologi	Pak Munir
MPPL	Pak Munir
Basis Data	Pak Budi
Provis	Pak Budi
Disweb	Pak Herbert
Strukdat	Bu Novi

```
7 rows in set (0.00 sec)
```

- ❖ nb : Bila ingin menampilkan semua data yang ada di tabel sebelah kiri baik yang mempunyai pasangan ataupun yang tidak mempunyai pasangan dengan data pada tabel di sebelah kanan, maka gunakan left join. Sedangkan bila yang terjadi adalah kebalikannya, yaitu ingin menampilkan semua data yang ada di sebelah kanan baik yang mempunyai pasangan ataupun yang tidak mempunyai pasangan dengan data pada tabel di sebelah kiri, maka gunakan right join.