

Group By, Order By dan Having Pada Fungsi Aggregate

Dalam menjalankan sebuah *query* untuk melakukan pengelompokan data dapat menggunakan Fungsi Group by dan diikuti oleh Fungsi Agregat.

Group By

Group By digunakan untuk pengelompokan/menampilkan atau memilih sekumpulan data berdasarkan atribut tertentu. Sebagai contoh Anda diminta untuk menampilkan banyaknya barang yang ada di transaksi barang dan total jumlah barang yang di order di transaksi barang untuk setiap barang.

1. Pengelompokan nya biasa nya di sertai oleh Agregat Fuction
2. Dalam Implementasi nya Agregat Function harus di ikuti oleh Group by bila terdapat Field lain yang dijadikan kriteria pengelompokan

Aggregate Function

Suatu fungsi yang digunakan untuk mendapatkan nilai tunggal berdasarkan hasil perhitungan nilai-nilai pada suatu kolom, contohnya adalah sebagai berikut :

MIN()

Digunakan untuk mendapatkan nilai minimum atau nilai terkecil dari sebuah data record di tabel.

```
Syntax:  
SELECT Field-1,...Field-n,  
MIN (nama_field) AS nama  
FROM tabel  
GROUP BY (Field-1,...Field-n)
```

Contoh:

Tersedia tabel Pasok sebagai berikut

	kode_pasok character(8)	kode_barang character(8)	kode_suplier character(8)	tanggal_pasok date	jumlah_pasok numeric(10,0)
1	pas001	brg001	sup002	2015-10-05	50
2	pas002	brg003	sup002	2015-10-02	350

```
SELECT kode_barang,  
MIN (jumlah_Pasok) AS Jumlah  
From PASOK  
GROUP BY (kode_barang)
```

Hasilnya:

	kode_barang character(8)	jumlah numeric
1	brg001	50
2	brg003	350

MAX()

Digunakan untuk mendapatkan nilai maximum atau nilai terbesar dari sebuah data record di tabel.

```
Syntax:  
SELECT Field-1,...Field-n,  
MAX (nama_field) AS nama  
FROM tabel  
GROUP BY (Field-1,...Field-n)
```

Contoh:

Tersedia tabel Pasok sebagai berikut

	kode_pasok character(8)	kode_barang character(8)	kode_suplier character(8)	tanggal_pasok date	jumlah_pasok numeric(10,0)
1	pas001	brg001	sup002	2015-10-05	50
2	pas002	brg003	sup002	2015-10-02	350

```
SELECT kode_barang,
MAX (jumlah_Pasok) AS Jumlah
From PASOK
GROUP BY (kode_barang)
```

Hasilnya:

	kode_barang character(8)	jumlah numeric
1	brg001	50
2	brg003	350

Pada min() dan max(), jika ada 2 data pada field yang sama maka dari data itu hanya akan dimunculkan yang terkecil atau terbesarnya saja.

Contoh:

Tersedia tabel Pasok sebagai berikut

	kode_pasok character(8)	kode_barang character(8)	kode_suplier character(8)	tanggal_pasok date	jumlah_pasok numeric(10,0)
1	pas001	brg001	sup002	2015-10-05	50
2	pas002	brg003	sup002	2015-10-02	350
3	pas003	brg003	sup002	2015-10-07	400
4	pas004	brg003	sup002	2015-10-08	30
5	pas005	brg001	sup002	2015-10-09	75

```
SELECT kode_barang,
MIN (jumlah_Pasok) AS Jumlah
From PASOK
GROUP BY (kode_barang)
```

	kode_barang character(8)	jumlah numeric
1	brg001	50
2	brg003	30

```
SELECT kode_barang,
MAX (jumlah_Pasok) AS Jumlah
From PASOK
GROUP BY (kode_barang)
```

	kode_barang character(8)	jumlah numeric
1	brg001	75
2	brg003	400

AVG()

Fungsi AVG digunakan untuk mencari nilai rata-rata (*average*) dari sebuah data atau record di tabel.

Syntax:

1

```
SELECT AVG (nama_field),  
AS rata-rata  
FROM tabel
```

2

```
SELECT Field-1,...Field-n,  
AVG (nama_field) AS rata-rata  
FROM tabel  
GROUP BY (Field-1,...Field-n)
```

Contoh:

Tersedia tabel Pasok sebagai berikut

	kode_pasok character(8)	kode_barang character(8)	kode_suplier character(8)	tanggal_pasok date	jumlah_pasok numeric(10,0)
1	pas001	brg001	sup002	2015-10-05	50
2	pas002	brg003	sup002	2015-10-02	350
3	pas003	brg003	sup002	2015-10-07	400
4	pas004	brg003	sup002	2015-10-08	30
5	pas005	brg001	sup002	2015-10-09	75

```
SELECT kode_barang,  
AVG (jumlah_Pasok) AS rata_rata  
From PASOK  
GROUP BY (kode_barang)
```

Hasilnya:

	kode_barang character(8)	rata_rata numeric
1	brg001	62.5000000000000000
2	brg003	260.0000000000000000

SUM()

Fungsi SUM digunakan untuk jumlah total nilai dari sebuah data atau record di tabel.

```
Syntax:  
SELECT Field-1,...Field-n,  
SUM (nama_field) AS jumlah  
FROM tabel  
GROUP BY (Field-1,...Field-n)
```

Contoh:

Tersedia tabel Pasok sebagai berikut

	kode_pasok character(8)	kode_barang character(8)	kode_suplier character(8)	tanggal_pasok date	jumlah_pasok numeric(10,0)
1	pas001	brg001	sup002	2015-10-05	50
2	pas002	brg003	sup002	2015-10-02	350
3	pas003	brg003	sup002	2015-10-07	400
4	pas004	brg003	sup002	2015-10-08	30
5	pas005	brg001	sup002	2015-10-09	75

```
SELECT kode_barang,  
SUM (jumlah_Pasok) AS Jumlah  
From PASOK  
GROUP BY (kode_barang)
```

Hasilnya:

	kode_barang character(8)	jumlah numeric
1	brg001	125
2	brg003	780

COUNT()

Fungsi COUNT digunakan untuk mencari jumlah dari sebuah data atau record di tabel.

```
Syntax:  
SELECT Field-1,...Field-n,  
COUNT (nama_field) AS jumlah  
FROM tabel  
GROUP BY (Field-1,...Field-n)
```

Contoh:

Tersedia tabel Pasok sebagai berikut

	kode_pasok character(8)	kode_barang character(8)	kode_suplier character(8)	tanggal_pasok date	jumlah_pasok numeric(10,0)
1	pas001	brg001	sup002	2015-10-05	50
2	pas002	brg003	sup002	2015-10-02	350
3	pas003	brg003	sup002	2015-10-07	400
4	pas004	brg003	sup002	2015-10-08	30
5	pas005	brg001	sup002	2015-10-09	75

```
SELECT kode_barang,  
COUNT (jumlah_Pasok) AS Jumlah  
From PASOK  
GROUP BY (kode_barang)
```

Hasilnya:

	kode_barang character(8)	jumlah bigint
1	brg001	2
2	brg003	3

ORDER BY

Fungsi *ordering* digunakan untuk mengurutkan tampilan field/data sesuai ketentuan yang digunakan/berdasarkan atribut tertentu. Misal diminta untuk ditampilkan nama barang dan nama sub mereknya yang diurutkan berdasarkan nama sub mereknya.

```
Syntax:  
ASCENDING  
SELECT Field-1,...Field-n  
FROM tabel  
ORDER BY field  
  
DESCENDING  
SELECT Field-1,...Field-n  
FROM tabel  
ORDER BY field DESC
```

Contoh:

```
SELECT kode_barang, jumlah_pasok  
FROM PASOK  
ORDER BY jumlah_pasok
```

	kode_barang character(8)	jumlah_pasok numeric(10,0)
1	brg003	30
2	brg001	50
3	brg001	75
4	brg003	350
5	brg003	400

```
SELECT kode_barang, jumlah_pasok  
FROM PASOK  
ORDER BY jumlah_pasok DESC
```

	kode_barang character(8)	jumlah_pasok numeric(10,0)
1	brg003	400
2	brg003	350
3	brg001	75
4	brg001	50
5	brg003	30

HAVING :

Having digunakan untuk menggantikan WHERE dan biasanya digunakan untuk menyeleksi data berdasarkan kriteria tertentu ketika harus dipasangkan dengan fungsi agregat, di mana kriteria berupa fungsi agregat. Sebagai contoh diminta untuk menampilkan data barang yang memiliki lebih dari satu transaksi barang.

Contoh:

```
SELECT kode_barang,  
COUNT (jumlah_pasok) AS cacah_pasok  
FROM PASOK  
GROUP BY kode_barang  
HAVING COUNT (*)=3
```

Hasilnya:

	kode_barang character(8)	cacah_pasok bigint
1	brg003	3