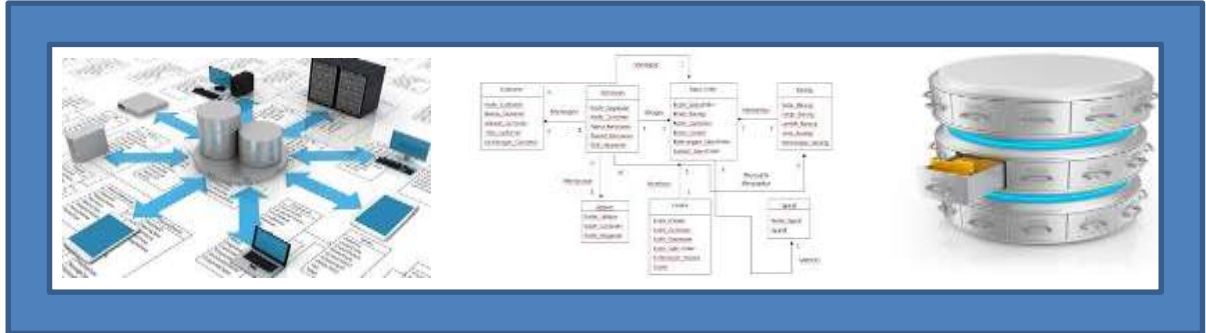


BUKU AJAR

PENGGUNAAN STRUCTURE QUERY LANGUAGE (SQL) DALAM BASIS DATA



PENYUSUN

MUKHLISULFATIH LATIEF

RAMPI YUSUF

NOVRI KANDOWANGKO

KATA PENGANTAR

Assalamu'alaikum Warahmatulahi Wabarakatuh.

Alhadulillah, segala puji dan syukur kami sampaikan ke hadirat Allah Swt., yang telah dilimpahkan rahmat serta hidayah-Nya sehingga buku yang berjudul “ Penggunaan Struktur Query Language (SQL) dalam Basis Data” karya mukhlisulfatih, dkk ini dapat terselesaikan dan kami terbitkan.

Sebagaimana yang kita tahu, SQL(Structured Query Language) adalah sebuah bahasa yang digunakan untuk mengaksesdata dalam basis data relasional. Bahasa ini secara de facto merupakan bahasa standar yang digunakan dalam manajemen basis data relasional. Saat ini hampir semua serverbasis data yang ada mendukung bahasa ini untuk melakukan manajemen datanya. Melalui buku ini, pembaca dapat mempelajari penggunaan dari perintah SQL meliputi : perintah select menggunakan kondisi, agregat, order by dan group by, join table, dan sub query.

Akhirnya, dengan diterbitkannya buku ini, diharapkan dapat memberikan sebuah pemahaman komprehensif mengenai implementasi dari perintah SQL. Selamat menikmati buku ini.

Wassalamu'alaium warahmatulahi wabarakatuh.

Hormat Kami,

Penulis

Pelajaran 1

PERINTAH SELECT

Tujuan Pembelajaran

Setelah Mempelajari Materi Pada Subbab 1 ini anda diharapkan dapat mengerti perintah select

1.1 Pendahuluan

SQL menyediakan perintah select untuk mengakses dan menampilkan data dari Tabel. Dengan sintaks sebagai berikut :

Select [All | Distinct] Pilih_daftar_Field

From Nama_tabel

[Where Kondisi]

[Group By Nama_Field]

[Having Kondisi]

[Order By Nama_Field]

Keterangan :

Pilih_daftar_field : Menyatakan pilihan terhadap kolom atau atribut dari data yang dipilih.

Nama_tabel : Tabel yang akan tampilkan datanya.

A. Perintah Select untuk Memilih Semua Kolom

Perintah select untuk menampilkan semua data dan kolom pada suatu table pada database.

sebagai contoh misalkan akan menampilkan data pada table Jurusan pada database NilaiMahasiswa, dapat menggunakan 2 (dua) cara:

1. Dengan menggunakan tanda (*)

*SELECT **

From Jurusan

B. Menampilkan data dengan kolom/atribute tertentu

Untuk menampilkan data pada kolom tertentu pada suatu table dalam database sintaksnya adalah sebagai berikut :

```
Select nama_kolom,[nama_kolom,....]
```

```
From Nama_tabel
```

Berikut ini adalah akan menampilkan data dalam kolom nim dan nama_mhs dari table Mahasiswa, maka perintah SQL nya adalah:

```
SELECT nim, nama_mhs
```

```
From Mahasiswa
```

C. Mengganti Nama Kolom/Field

Ketika Query menampilkan hasil perintah select, nama kolom/field yang ditampilkan sesuai dengan nama kolom yang dispesifikasikan dalam table. Untuk mengganti nama kolom yang dihasilkan sesuai dengan keinginan kita, sintaks perintahnya adalah :

```
Select nama_field1 field_alias1 , [nama_field2 field_alias2 ]
```

```
From nama_tabel
```

Keterangan :

Field alias merupakan judul Field yang diinginkan user.

Sebagai contoh, misalkan kita akan menampilkan nim, nama_m tampilan kolomnya menjadi 'Nama Mahasiswa', tpt_lhr_m menjadi 'Tempat Lahir', maka perintah query nya adalah:

```
SELECT nim,nama_m 'Nama Mahasiswa',tpt_lhr_m 'TempatLahir'
```

```
FROM Mahasiswa
```

D. Operator Aritmatika

Pernyataan SQL mendukung operator yang menampilkan operasi aritmatika seperti, penjumlahan, pengurangan, pembagian, perkalian, dan modulus pada kolom dengan jenis data numeric. Jenis data numeric adalah INT, Smallint, Decimal, Numeric, Float, Real dan lainnya tergantung DBMS yang digunakan.

Operator yang didukung oleh pernyataan SQL adalah :

1. + untuk penjumlahan
2. - untuk pengurangan

3. / untuk pembagian
4. * untuk perkalian
5. % untuk modulus

d.1. Operator penjumlahan (+)

Tampilkan kdmk,nama_mk,sks dari table Matakuliah, dimana kolom kdmk diganti menjadi 'Kode Matakuliah, kolom nama_mk diganti menjadi 'Matakuliah' serta kolom sks + 2, maka perintahnya adalah:

```
SELECT kdmk 'Kode Matakuliah', namamk 'Matakuliah',  
sks,sks+2 'SKS + 2'  
FROM Matakuliah
```

1.2 Latihan

1. Buatlah sebuah tabel barang dan isikan datanya sebagai berikut :

Kode_barang	Nama	Stok
M	Mouse	30
P	Printer	20
L	Lcd	5
K	Komputer	30

2. Praktekkan penggunaan masing-masing perintah diatas

1.3 Tugas

1. Masukkan data-data (record) seperti dibawah ini :

Nomor_pemilih	Nama_pemilih	Kecamatan	Umur
10	Ramdan	Barat	16
11	Rahmat	Timur	17
12	Ridwan	Utara	45
13	Yanto	Utara	48

2. Ubahlah field **kecamatan** menjadi **Selatan** untuk kecamatan = **Utara**

3. Ubahlah **nama_pemilih** menjadi "Ridwan AB" dan **umur** menjadi 40 untuk Nomor_pemilih ="12"

4. Hapus pemilih yang mempunyai **umur** dibawah 20 tahun

5. Tampilkan semua data pemilih

Pelajaran 2

PERINTAH SELECT MENGGUNAKAN KONDISI

2.1 Menyeleksi Baris dengan Kondisi

Setelah Mempelajari Materi Pada Subbab 2 ini anda diharapkan dapat menampilkan data berdasarkan kondisi tertentu

Adakalanya hanya beberapa baris saja yang perlu di-*retrieved* dari sebuah *table*. Clausa Where disediakan oleh SQL untuk menspesifikasikan kondisi tersebut. Sintaks dari pernyataan tersebut adalah :

Select Daftar_kolom

From nama_tabel

Where kondisi

Pada metode, klausa where dapat dibagi dalam beberapa kategori seperti berikut:

1. Comparison operator : seperti =,>,<,>=,<=,!=/<>
2. Range operator : seperti Between dan Not Between
3. List operator : seperti In dan Not In
4. String operator : seperti Like dan Not Like
5. Logical Operator : seperti And, Or, Not.

1. Operator Comparison

Pada operator comparison operator ini dapat dikelompokkan menjadi operator =, (sama dengan), > (lebih besar), < (lebih kecil), >= (lebih besar sama dengan), <= (lebih kecil sama dengan), != / <> (tidak sama dengan).

a. Operator =

Tampilkan nim,kdmk,nilai_absen,nilai_tugas,nilai_uts, nilai_uas dari table Nilai, dimana nilai_uas>80, maka perintah nya adalah:

```
SELECT nim,kdmk,nilai_absen,nilai_tugas,nilai_uts,
```


nilai_uas

FROM Nilai

WHERE nilai_uas=80

b. Operator != /<>

Tampilkan nim,kdmk,nilai_absen,nilai_tugas,nilai_uts, nilai_uas dari table Nilai, dimana nilai_absen<>13, maka perintah nya adalah:

SELECT nim,kdmk,nilai_absen,nilai_tugas,nilai_uts, nilai_uas

FROM Nilai

WHERE nilai_absen<>13 ;

2. Range Operator

Operator Range digunakan untuk meretrieved data yang dapat diperoleh dalam range tersebut, operator tersebut meliputi Between dan Not Between.

Sintaks untuk operator range tersebut adalah:

Select Daftar_kolom

From nama_tabel

Where nama_kolom [not] Between ekspresi1 and ekspresi 2

a. Operator Between

Tampilkan nim,nama_m,tgl_lhr_m untuk kolom nama_m diganti 'Nama Mahasiswa', kolom tgl_lhr_m diganti 'Tanggal Lahir' dimana mahasiswa yang tanggal lahirnya antara '11/17/1977' sampai '10/19/1980, maka perintah nya adalah:

SELECT nim,nama_m 'Nama Mahasiswa', tgl_lhr_m 'Tanggal Lahir'

FROM Mahasiswa

Where tgl_lhr_m BETWEEN '1977/12/10 ' and '1980/12/10'

b. Operator Not Between

Tampilkan kdmk,nama_mk,sks dari table matakuliah, untuk kolom kdmk diganti menjadi 'Kode Matakuliah' kolom nama_mk diganti 'Matakuliah'. Dimana matakuliah

yang sks nya bukan antara 1 dan 2, maka perintah nya adalah :

SELECT kdmk 'Kode Matakuliah', nama_mk, 'Matakuliah',sks

FROM Matakuliah

WHERE sks Not BETWEEN 1 and 2

3. List Operator

Operator List digunakan untuk menampilkan data yang dapat diperoleh dalam daftar (batasan) tertentu, operator tersebut meliputi In dan Not In.

Sintaks untuk operator list tersebut adalah:

Select Daftar_kolom

From nama_tabel

Where nama_kolom operator List ('Daftar_value')

a. Operator In

Tampilkan nim,nama_m,tpt_lhr_m,telpon_m untuk kolom nama_m diganti 'Nama Mahasiswa', kolom tpt_lhr_m diganti 'Tempat Lahir', kolom telpon_m diganti 'Telepon'. Dimana mahasiswa yang tempat lahirnya di Bogor, Surabaya dan Solo, maka perintah nya adalah:

SELECT nim,nama_m 'Nama Mahasiswa',tpt_lhr_m 'Tempat

Lahir',telpon_m 'Telepon'

FROM Mahasiswa

WHERE tpt_lhr_m IN ('Bogor','Surabaya','Solo')

b. Operator NOT In

Tampilkan nim,nama_m,tpt_lhr_m,telpon_m untuk kolom nama_m diganti 'Nama Mahasiswa', kolom tpt_lhr_m diganti 'Tempat Lahir', kolom telpon_m diganti 'Telepon'.

Dimana mahasiswa yang tempat lahirnya Bukan di Bekasi, Bogor dan Jakarta, maka perintah nya adalah:

SELECT nim,nama_m 'Nama Mahasiswa',tpt_lhr_m 'Tempat

Lahir', telpon_m 'Telepon'

FROM Mahasiswa

WHERE tpt_lhr_m NOT IN ('Bekasi','Bogor','Jakarta')

4. String Operator

Key Word Like digunakan untuk memilih baris-baris yang sesuai dengan karakter yang digunakan. Like menggunakan karakter Wildcard yang bisa digunakan sebagai expresi.

a. Wildcard %

Tampilkan nim,nama_m, untuk kolom nama_m diganti 'Nama Mahasiswa', dimana nama mahasiswanya untuk 2 huruf pertama adalah 'De', maka perintahnya adalah:

```
SELECT nim,nama_m ' Nama Mahasiswa'  
FROM Mahasiswa  
WHERE nama_m LIKE 'De%'
```

b. Wildcard _

Tampilkan nim,nama_m, untuk kolom nama_m diganti 'Nama Mahasiswa', dimana nama mahasiswanya setelah 3 huruf pertama mengandung huruf 'a', maka perintahnya adalah:

```
SELECT nim,nama_m 'Nama Mahasiswa'  
FROM Mahasiswa  
WHERE nama_m LIKE '___a%'
```

Tugas :

5. Logical Operator

Operator Logika yang umum disediakan didalam SQL adalah sebagai berikut:

- OR Memberikan nilai True bila kondisi yang dispesifikasikan True
- NOT Membalikkan nilai expresi yang mengikutinya
- AND Memberikan hasil True bila semua kondisi yang dispesifikasikan True.

Sintaks untuk operator logika tersebut adalah:

```
Select daftar_kolom
```

```
From nama_tabel
```

```
Where conditional expresi {AND/OR} [Not] conditional expresi
```

Contoh : tampilkan nama pegawai yang mempunyai status pegawai tetap dan memiliki gaji diatas 3 juta pada tabel pegawai.

Jawab :

Select nama_pegawai from pegawai where status='tetap' and gaji>=3000000

2.2 Latihan

Apa hasil dari perintah dibawah ini jika yang dimasukkan sebagian perintah sql seperti dibawah ini :

- a. 'A_din%'
- b. '%din_'

2.3 Tugas

1. Tampilkan NAMA MAHASISWA yang mengambil matakuliah “PEMROGRAMAN WEB” dan nilainya > 81
2. Tampilkan NAMA MAHASISWA yang namanya , HURUF ke-2 adalah ‘D’ atau yang huruf pertamanya ‘A’
3. Tampilkan NAMA NASABAH yang KOTANYA di Jakarta,bandung,bogor dan saldonya antara 2jt-3jt
4. Tampilkan NAMA NASABAH yang namanya , HURUF ke-3 adalah ‘R’ dan KOTAny (Jakarta atau bogor)
5. Tampilkan nama nasabah dan jumlah saldonya apabila terdapat tambahan bunga sebesar 10% dari saldonya

Pelajaran 3

FUNGSI AGREGAT DALAM SQL

3.1 Pendahuluan

Setelah Mempelajari Materi Pada Subbab 3 ini anda diharapkan dapat mengerti penggunaan fungsi agregat

Dalam SQL dikenal fungsi agregat. Fungsi agregat (*aggregate*) adalah fungsi yang menerima koleksi nilai dan mengembalikan nilai tunggal sebagai hasilnya.

Perintah	Fungsi
MIN	Mencari jumlah paling kecil / minimum
MAX	Mencari jumlah paling besar / maksimum
AVG	Menghitung rata-rata
SUM	Menghitung hasil penjumlahan
COUNT	Menghitung hasil pencacahan / counter

3.1.1 AVG(ekspresi)

Fungsi ini digunakan untuk mencari rata-rata nilai dalam suatu kolom dari suatu tabel atau ekspresi. Ekspresi dalam fungsi AVG umumnya adalah nama kolom dengan tipe data numerik.

Contohnya : `select AVG(Harga) from Nama_Tabel`

3.1.2 COUNT(x)

Fungsi ini digunakan untuk menghitung jumlah record/baris dari suatu kolom atau suatu tabel. X merupakan nama kolom dari tabel yang ingin dicari jumlah baris datanya.

Contoh : `select COUNT(*) from Nama_Tabel`

3.1.3 MAX(ekspresi)

Fungsi ini digunakan untuk mencari nilai terbesar dari suatu kolom dalam suatu tabel atau ekspresi. Kolom yang dicari nilai terbesarnya memiliki tipe data numerik.

Contoh : `select MAX(Harga) from Nama_Tabel`

3.1.4 MIN(ekspresi)

Fungsi MIN() ini merupakan kebalikan dari fungsi MAX(). Fungsi ini mencari nilai terkecil dari suatu kolom dalam suatu tabel atau ekspresi.

Contoh : `select MIN(Harga) from Nama_Tabel`

3.1.5 SUM(ekspresi)

Fungsi ini digunakan untuk mendapatkan nilai total dari suatu kolom dalam suatu tabel atau ekspresi.

Contoh : `select SUM(Harga) from Nama_Tabel`

Masih menggunakan tabel Karyawan, kita ingin menampilkan gaji tertinggi. Maka perintahnya:

```
select MAX(gaji)
from Karyawan;
```

Hasilnya:

Gaji
300000

Lain halnya bila kita ingin menghitung jumlah dari gaji Karyawan. Perintahnya adalah:

```
select SUM(gaji)
from Karyawan;
```

Hasilnya:

600000

Contoh lain adalah bila kita ingin mengetahui jumlah karyawan maka perintahnya sebagai berikut :

```
select COUNT(*)
from Karyawan;
```

3.2 Latihan

1. Sebelum mencoba berbagai fungsi agregat SQL, buatlah tabel dengan struktur sebagai berikut :

#	Column	Type
1	<u>kode_buku</u>	char(10)
2	<u>judul_buku</u>	varchar(100)
3	<u>pengarang</u>	varchar(50)
4	<u>penerbit</u>	varchar(50)
5	<u>tahun</u>	year(4)
6	<u>tanggal_inventaris</u>	date
7	<u>resensi</u>	text
8	<u>harga</u>	int(7)

2. Tambahkan record-record berikut ini:

kode_buku	judul_buku	pengarang	penerbit	tahun	tanggal_inventaris	resensi	harga
B000000001	Harri Potter	JK. Rowling	British Press	2002	2012-05-01	Fiksi	50000
B000000002	Sistem Basis Data	Abdul Kadir	Andi Offset	2002	2012-05-01	Buku Teks	30000
B000000003	Sistem Basis Data	Fathansyah	ITB Press	2003	2012-05-08	Buku Teks	45000
B000000004	Prophet Muhammad	Amir Abdulla	Madina Press	2003	2012-05-09	Biografi	75000
B000000006	Ketika Cinta Bertasbih	Habiburahman El-Syrasy	Madina Press	2003	2012-05-30	Fiksi	67000
B000000007	Pemrograman Basis Data	Abdul Kadir	Andi Offset	2004	2011-06-09	Buku Teks	56000

Sekarang cobalah anda gunakan perintah berikut untuk menyelesaikan soal dibawah ini.

Fungsi Agregat: COUNT

Digunakan untuk menghitung jumlah record.

Contoh:

1. Hitung jumlah record tabel buku

```
SELECT COUNT(*)  
FROM buku;
```

2. Hitung jumlah record tabel buku dengan nama kolom jum_rec

```
SELECT COUNT(*) AS jum_rec  
FROM buku;
```

3. Hitung jumlah record untuk tahun 2003

```
SELECT COUNT(*) AS jum_rec  
FROM buku
```



```
WHERE tahun = 2003;
```

Fungsi Agregat: SUM

Fungsi Agregat SUM digunakan untuk menghitung total nilai dari kolom tertentu

Contoh:

1. Hitung total harga

```
SELECT SUM(harga) AS total_harga  
FROM buku;
```

2. Hitung total harga untuk tahun 2003

```
SELECT SUM(harga) AS total_harga  
FROM buku  
WHERE tahun=2003;
```

Fungsi Agregat: MAX

Fungsi agregat MAX Digunakan untuk menampilkan nilai tertinggi dari suatu kolom

Contoh:

1. Tampilkan harga tertinggi

```
SELECT MAX(harga) AS harga_tertinggi  
FROM buku;
```

2. Tampilkan harga tertinggi untuk tahun 2003

```
SELECT MAX(harga) AS harga_tertinggi  
FROM buku  
WHERE tahun=2003;
```

Fungsi Agregat: MIN

Fungsi agregat MIN Digunakan untuk menampilkan nilai terendah dari suatu kolom

Contoh:

1. Tampilkan harga terendah

```
SELECT MIN(harga) AS harga_terendah  
FROM buku;
```

2. Tampilkan harga terendah untuk tahun 2003

```
SELECT MIN(harga) AS harga_terendah  
FROM buku  
WHERE tahun=2003;
```

Fungsi Agregat: AVG

Fungsi agregat AVG Digunakan untuk menampilkan nilai rata-rata dari suatu kolom

Contoh:

1. Tampilkan harga rata-rata

```
SELECT AVG(harga) AS harga_rerata  
FROM buku;
```

2. Tampilkan harga rata-rata untuk tahun 2003

```
SELECT AVG(harga) AS harga_rerata  
FROM buku  
WHERE tahun=2003;
```

3.3 Tugas

1. Aktifkan database yang sudah anda buat sebelumnya, yang recordnya adalah sbb :

9005	arman	tetap	7	2.000.000	10-12-1977
9006	budi	kontrak	7	500.000	11-12-1977
9007	andi	tetap	8	3.000.000	12-12-1977
9008	iwan	kontrak	9	800.000	13-12-1977
9009	santi	PNS	7	4.000.000	14-12-1977

2. Tampilkan Total gaji untuk pegawai yang mempunyai status 'kontrak'
3. Tampilkan gaji yang terkecil
4. Tampilkan jumlah pegawai yang gajinya dibawah 1.000.000

Pelajaran 4

PERINTAH ORDER BY DAN GROUP BY

4.1 Order By

Setelah Mempelajari Materi Pada Subbab 4 ini anda diharapkan dapat mengerti penggunaan order by dan group by

Order by adalah perintah yang digunakan untuk mengurutkan data pada tabel secara ascending (menaik) atau descending (menurun).

Misalkan kita memiliki tabel Karyawan seperti ini:

Kode	Nama	Gaji
KP01	Amrin	200000
KP02	Camelia	300000
KP03	Bembi	100000

Untuk menampilkan tabel diatasurut berdasarkan Nama dari A-Z (ascending) perintahnya adalah:

```
select Kode,Nama,Gaji
```

```
from Karyawan
```

```
order by Nama;
```

Perintah diatas akan menampilkan hasil seperti ini:

Kode	Nama	Gaji
KP01	Amrin	200000
KP03	Bembi	100000
KP02	Camelia	300000

Perhatikan, tabel diatas menampilkan data urut Nama dari A-Z. Sebaliknya, untuk menampilkan data urut Nama dari Z-A (descending), perintahnya adalah:

```
select Kode, Nama, Gaji  
from Karyawan  
order by Nama DESC;
```

Anda bisa juga memodifikasinya dengan perintah Where. Misal saja anda hanya ingin menampilkan data karyawan bergaji > 100000 dan urut Kode dari besar ke kecil (descending), maka anda gunakan perintah ini:

```
select Kode, Nama, Gaji  
from Karyawan  
where gaji > 100000  
order by Kode DESC;
```

Hasilnya akan tampak seperti ini:

Kode	Nama	Gaji
KP02	Camelia	300000
KP01	Amrin	200000

Tampilkan nim, nama, jenjang (dari table Mahasiswa), dimana jenjangnya = S1 dan diurutkan berdasarkan nama mahasiswa secara Descending, maka perintahnya adalah:

```
SELECT nim, nama, jenjang  
FROM Mahasiswa WHERE jenjang= 'S1'  
ORDER BY nama DESC
```

4.2 Group By

Pemakaian perintah select selain dapat digabungkan dengan anak kalimat ORDER BY yang bertujuan untuk mengurutkan data, juga dapat digabungkan dengan anak kalimat GROUP BY yang bertujuan untuk mengelompokkan data (record) yang sama. Pada kasus sederhana anak kalimat GROUP BY berfungsi seperti perintah DISTINCT, kelebihan GROUP BY adalah dapat digunakan

bersama dengan fungsi agregate yang beroperasi pada seluruh record yang akan digabung untuk membentuk satu record (baris) tunggal.

4.2 Tugas

1. Tampilkan nama jurusan (dari table Jurusan), jumlahkan MHS tiap jurusan dan ditampung pada kolom 'Jumlah Mahasiswa' dan dikelompokkan berdasarkan nama jurusan serta diurutkan berdasarkan nama jurusan secara Ascending
2. Tampilkan semester (dari table matkul), jumlahkan SKS tiap semester dan ditampung pada kolom 'Jumlah SKS' dan dikelompokkan berdasarkan semester serta diurutkan berdasarkan semester secara Descending

Pelajaran 5

PERINTAH GROUP BY DAN HAVING

5.1 Statement Having

Setelah Mempelajari Materi Pada Subbab 5 ini anda diharapkan dapat mengerti penggunaan group by dan having

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)

DARI nama_tabel

WHERE nama_kolom operator nilai

GROUP BY nama_kolom

HAVING aggregate_function (nama_kolom) Nilai operator

Contoh penggunaan HAVING dalam SQL dalam tabel pesanan seperti pada tabel dibawah ini :

No	Tanggal Order	Harga Order	Pelanggan
1	2011/11/12	1000	Ratih
2	2011/10/23	1600	Anita
3	2011/09/02	700	Ratih
4	2011/09/03	300	Ratih
5	2011/08/30	2000	Teguh
6	2011/10/04	100	Anita

Contoh 1 :

Sekarang jika kita ingin mencari pelanggan yang memiliki total order kurang dari 2000.

Maka, kita dapat menggunakan statement :

```
SELECT Pelanggan, SUM (HargaOrder) FROM Pesanan  
GROUP BY Pelanggan  
HAVING SUM (HargaOrder) <2000
```

Maka, hasilnya :

Pelanggan	SUM(HargaOrder)
Anita	1700

Contoh 2 :

Dan jika kita ingin menemukan apakah pelanggan "Ratih" atau "Teguh" memiliki total order lebih dari 1500.

Kita dapat menambahkan WHERE contohnya yaitu sebagai berikut:

```
SELECT Pelanggan, SUM (HargaOrder) FROM Pesanan  
WHERE Pelanggan 'Hansen' = OR Pelanggan 'Jensen '=  
GROUP BY Pelanggan  
HAVING SUM (HargaOrder)> 1500
```

Maka, hasilnya:

Pelanggan	SUM(HargaOrder)
Ratih	2000
Teguh	2000

Contoh 3 :

Tampilkan jumlah jam bekerja dari pegawai yang jumlahnya>21 pada tabel Pekerjaan

Nama	Jam Bekerja
Roni	12
Dewi	10
Roni	8

Dewi	15
------	----

SELECT Nama, SUM (Jam_bekerja)

FROM NamaJam_Bekerja

GROUP BY Nama

HAVING SUM (Jam_Bekerja) > 21

Maka, Hasilnya adalah :

Nama	Jam Bekerja
Roni	27

5.2 Tugas

Aktifkan database yang sudah anda buat sebelumnya, yang recordnya adalah sbb :

9005	arman	tetap	7	2.000.000	10-12-1977
9006	budi	kontrak	7	500.000	11-12-1977
9007	andi	tetap	8	3.000.000	12-12-1977
9008	iwan	kontrak	9	800.000	13-12-1977
9009	santi	PNS	7	4.000.000	14-12-1977

1. Tampilkan gol apa saja yang ada di tabel pegawai
2. Tampilkan nip,nama,status, gapok untuk pegawai yang tgl lahirnya diatas tgl '11-12-1977' . urutkan berdasarkan gapok secara desc.
3. Tampilkan status pegawai, jumlah pegawai yang dikelompokkan berdasarkan status dan dirutkan secara desc
4. Tampilkan gol pegawai, total gaji masing-masing gol yang dikelompokkan berdasarkan gol, dan dirutkan secara desc
5. Tampilkan status pegawai, total gaji yang melebihi 4.000.000 yang dikelompokkan berdasarkan status. Urutkan secara desc.

Pelajaran 6

PERINTAH JOIN TABLE

6.1 Pendahuluan

Setelah Mempelajari Materi Pada Subbab 6 ini anda diharapkan dapat mengerti penggunaan join table

Join merupakan salah satu fitur SQL yang berguna menggabungkan tabel dengan query-query yang mendapatkan kembali data. Join adalah salah satu dari beberapa operasi paling penting yang dapat Anda lakukan dengan menggunakan statemen SELECT. Dan sebagai bagian penting dalam mempelajari SQL, Anda perlu memahami dengan baik join dan sintaks join.

Alasan utama digunakannya pemisahan data ke dalam banyak tabel memungkinkan proses penyimpanan menjadi lebih efisien, manipulasi lebih mudah dan skalabilitas yang lebih besar. Tetapi manfaat tersebut ada tuntutan.

Jika data disimpan dalam banyak tabel, bagaimana Anda dapat mendapatkan kembali data dengan sebuah statemen SELECT tunggal ?

Untuk membuat join sangat sederhana. Anda harus menentukan semua tabel yang akan dimasukkan dan bagaimana tabel itu dihubungkan satu dengan yang lainnya. Lihat contoh berikut ini :

```
SELECT Supplier>Nama, Barang>Nama, Barang.Harga FROM Supplier, Barang  
WHERE Supplier.KodeSup = Barang.KodeSup
```

SINTAKS UNTUK JOIN / MENGGABUNGKAN 2 TABEL ATAU LEBIH ADALAH :

```
SELECT daftar_field (nama_tabel.nama_field)
```

```
FROM (daftar tabel )
```

```
WHERE field yang berhubungan (field A = field B)
```

Dan dapat juga ditambah dengan kriteria atau kondisi lainnya yaitu

AND kondisi

GROUP BY namafield

HAVING kondisi

ORDER BY namafield

6.2 Tugas

Buatlah desain database seperti ini, kemudian isikan datanya seperti dibawah ini :

Tabel : mhs		Tabel : mk		
nim	namaMhs	kodeMK	namaMK	sks
001	Joko	A01	Kalkulus	3
002	Amir	A02	Geometri	2
003	Budi	A03	Aljabar	3

Tabel : KHS		
nim	kodeMK	nilai
001	A01	3
001	A02	4
001	A03	2
002	A02	3
002	A03	2
003	A01	4
003	A03	3

1. Tampilkan nim yang ada di tabel KHS
2. Tampilkan kodemk yang ada di tabel KHS
3. Tampilkan nim,nama yang ada di tabel KHS
4. Tampilkan kodemk, namamk yang ada di tabel KHS
5. Tampilkan nama mhs, nama mata kuliah dan nilai yang diambil masing masing mahasiswa
6. Tampilkan nama mhs, nama mk, sks, nilai, sks x nilai oleh mahasiswa
7. Tampilkan nim dan nama mahasiswa yang mengambil Kalkulus (kodeMK = A01)
8. Tampilkan nim, nama mahasiswa dan jumlah SKS matakuliah yang diambil untuk setiap mahasiswa

Pelajaran 7

PERINTAH SUB QUERY

7.1 Sub Query

Setelah Mempelajari Materi Pada Subbab 7 ini anda diharapkan dapat mengerti penggunaan Sub Query

Sub query adalah sebuah pernyataan select di dalam sebuah Select, Insert, Up date atau Delete atau di dalam subquery lain. Itu semua dapat digunakan untuk mengembalikan data dari beberapa tabel dan sebagai alternatif untuk join. Sub query dapat bersarang atau berhubungan. Sub query yang bersarang dijalankan sekali pada saat Outer Query dijalankan, dan sub query yang berhubungan dijalankan sekali untuk setiap baris dihasilkan pada saat outer query dijalankan.

Berikut ini adalah petunjuk didalam penulisan suatu sub query didalam melakukan penelusuran data didalam database.

- Sub query ditulis dalam tanda kurung.
- Anda hanya boleh memakai sebuah ekspresi atau nama kolom pada daftar select dari sebuah sub query yang menghasilkan sebuah nilai.
- Anda dapat memakai sub query sebagai ganti sebuah ekspresi jika hasilnya adalah sebuah nilai tunggal atau daftar nilai.
- Tidak boleh memakai sub query pada kolom-kolom yang berisi tipe data Teks dan Image.

Sub query dapat dibagi dalam tiga kategori yang bergantung pada nilai pengembalian.

- Where <ekspresi> [Not] In (<Subquery>)

Hasil sub query yang memakai IN atau Not IN adalah daftar nilai yang berisi beberapa nilai atau kosong, setelah sub query memberikan hasil, outer query akan memakainya.

a. IN

Contoh berikut ini akan ditampilkan data matakuliah yang pernah dipakai untuk transaksi nilai dengan primary key kdmk, maka perintahnya adalah:

```
SELECT * FROM Matakuliah M Where M.kdmk IN (SELECT N.kdmk From Nilai N)
```

Hasil dari pernyataan query tersebut adalah 4 (empat) kolom dan record / barisnya adalah 3 (tiga), seperti tampak pada gambar berikut :

	kdmk	nama_mk	sks	semester
1	MPK131201	PENDIDIKAN AGAMA I	2	1
2	MPK131203	PENDIDIKAN PANCASILA	2	1
3	MPK131204	PENDIDIKAN KEWARGANEGARAAN	2	2

b. NOT IN

Penggunaan NOT IN dapat digunakan untuk memeriksa referential integrity antara table relasi dengan table master. Contoh berikut ini akan ditampilkan data matakuliah yang belum pernah dipakai untuk transaksi nilai dan jumlahnya sksnya, maka perintahnya adalah:

```
SELECT kdmk 'Kode Matakuliah',nama_mk 'Matakuliah',sks FROM Matakuliah M Where sks<>2 and M.kdmk NOT IN (SELECT N.kdmk From Nilai N) COMPUTE sum(sks)
```

Hasil dari pernyataan query tersebut adalah 3 (tiga) kolom dan record / barisnya adalah 5 (enam) dan jumlah sksnya = 13, seperti tampak pada gambar berikut :

	Kode Matakuliah	Matakuliah	sks
1	MKB331201	PENGANTAR TEKNOLOGI INFORMASI	3
2	MKB331203	PRAKTIKUM PAKET PROGRAM APLIKASI I	1
3	MKB331205	PERANCANGAN BASIS DATA	4
4	MKB331206	PAKET PROGRAM APLIKASI II	1
5	MKB331209	ALGORITMA PEMROGRAMAN I (PASCAL)	3
6	MKB331210	PRAK. ALGORITMA PEMROGRAMAN I (PASCAL)	1
sum			
1	13		

c. ANY dan ALL

Kata ANY dan ALL dapat digunakan pada subquery – subquery yang menghasilkan satu kolom angka – angka. Jika diawali kata ALL, syarat hanya akan bernilai TRUE jika dipenuhi semua nilai yang dihasilkan subquery ini. Jika diawali kata ANY, syarat akan bernilai TRUE jika dipenuhi sedikitnya satu nilai yang dihasilkan subquery itu.

1. ANY

Perintah berikut adalah menampilkan kolom kdmk, nama_mk,sks dari table matakuliah, untuk kolom kdmk diganti ‘Kode Matakuliah’, kolom nama_mk diganti ‘Matakuliah’, dimana hasil subquery adalah sama dengan sks lebih besar 2, maka perintahnya adalah:

```
SELECT kdmk 'Kode Matakuliah',nama_mk 'Matakuliah',sks FROM Matakuliah
Where sks = ANY (SELECT sks From Matakuliah WHERE sks>2)
```

Hasil dari pernyataan query tersebut adalah 3 (tiga) kolom dan record / barisnya adalah 3 (tiga), seperti tampak pada gambar berikut :

	Kode Matakuliah	Matakuliah	sks
1	MKB331201	PENGANTAR TEKNOLOGI INFORMASI	3
2	MKB331205	PERANCANGAN BASIS DATA	4
3	MKB331209	ALGORITMA PEMROGRAMAN I (PASCAL)	3

2. ALL

Perintah berikut adalah menampilkan kolom kdmk, nama_mk,sks dari table matakuliah, untuk kolom kdmk diganti 'Kode Matakuliah', kolom nama_mk diganti 'Matakuliah', dimana hasil subquery adalah sama dengan 3, maka perintahnya adalah :

```
SELECT kdmk 'Kode Matakuliah',nama_mk 'Matakuliah',sks FROM Matakuliah  
Where sks > ALL (SELECT sks From Matakuliah WHERE sks=3)
```

Hasil dari pernyataan query tersebut adalah 3 (tiga) kolom dan record / barisnya adalah 3 (tiga), seperti tampak pada gambar berikut :

	Kode Matakuliah	Matakuliah	sks
1	MKB331205	PERANCANGAN BASIS DATA	4

7.2 Tugas

1. Tampilkan nopeg, nama pegawai yang mempunyai kode_jab selain 's' atau gol dibawah 3
2. Berapa orang jumlah pegawai yang jabatannya 'staff'
3. Tampilkan nama, gapok pegawai yang mempunyai gapok yang bukan antara 1 jt – 2 jt
4. Tampilkan nama, tunjangan pegawai yang mempunyai tunjangan \geq 500.000
5. Tampilkan nama, dan jumlah penerimaan (gapok+tunjangan) tiap-tiap pegawai
6. Tampilkan status , total gapok yang di kelompokkan berdasarkan status dan urutkan dari total gapok dari yang besar ke yang kecil
7. Tampilkan gol, total jumlah penerimaan (gapok + tunjangan) yang dikelompokkan berdasarkan gol. Tampilkan hanya yang mempunyai total jumlah penerimaan antara 2 jt-3jt dan urutkan secara DESC
8. Tampilkan Nama pegawai yang mempunyai GAPOK terkecil
9. Siapa yang mempunyai Pendapatan (GAPOK + TUNJANGAN) Terbesar

DAFTAR KEPUSTAKAAN

- Abdul Kadir, *Konsep & Tuntunan Praktis Basis Data*, Andi Offset, Yogyakarta, 1999.
- Abdul Kadir, *Dasar Perancangan dan implementasi database relational*, Andi Offset, Yogyakarta, 2009.
- Bunawan, Suryadi H.S, *Seri Diktat Kuliah : Pengantar Basis Data*, Gunadarma, Jakarta, 1993.
- Fathansyah, *Basis Data*, Informatika, cetakan pertama, 1999.
- Hariato Kristanto, *Konsep dan Perancangan Database*, Andi Offset, Cetakan Pertama, Yogyakarta, 1994.
- Indrajani, *Sistem Basis Data dalam Paket five in One*, Elex Media Komputindo, Jakarta, 2009
- Janner Simarmata, *Perancangan Basis Data*, Andi Offset, Yogyakarta, 2007.