



**BINA INSANI
UNIVERSITY**

IMPLEMENTASI SQL - SELECT -

Pertemuan 13

Bigger Better Higher



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 terebut adalah:

SELECT *daftar_kolom*

FROM *nama_tabel*

WHERE *conditional expresi* {AND/OR} [Not]
conditional expresi



LOGICAL OPERATOR OR

Operator ini digunakan untuk menguji apakah salah satu kedua ekspresi logika yang diberikan memiliki nilai TRUE, dengan aturan penulisan:

Eksprisi1 OR Ekspresi2



Operator OR pada SQL akan menghasilkan nilai sesuai dengan table berikut:

Ekspresi1	Ekspresi2	Hasil
True	True	True
True	False	True
True	Null	True
False	True	True
False	False	False
False	Null	False
Null	True	True
Null	False	False
Null	Null	Null



Tampilkan nim,nama_m,tpt_lhr_m,alm_m,kota_m, untuk kolom nama_m diganti 'Nama Mahasiswa', kolom tpt_lhr_m diganti 'Tempat Lahir', kolom alm_m diganti 'Alamat' dan kolom kota_m diganti 'Kota'. Dimana mahasiswa yang tempat lahirnya di 'Bekasi' atau kota alamatnya adalah 'Cikarang', maka perintah nya adalah:

```
SELECT  nim,nama_m 'Nama Mahasiswa`  
          ,tpt_lhr_m 'Tempat Lahir`  
          ,alm_m 'Alamat',kota_m 'Kota'  
  
FROM    Mahasiswa  
  
WHERE    tpt_lhr_m='Bekasi` OR  
          kota_m='Cikarang'
```



LOGICAL OPERATOR OR LANJUTAN...

	nim	Nama Mahasiswa	Tempat Lahir	Alamat	Kota
1	I01031001	Devi Murbaini	Bekasi	Jl. Dahlia I Blok BC 2/3	Bekasi Utara
2	I01031002	Deni Hermawan	Jakarta	Jl. Seruni Raya No.3	Cikarang
3	T01031002	Fitria Choirunissa	Bekasi	Jl. Gandaria Selatan I No.8	Jakarta Timur
4	I01031003	Rini Hapsari	Bogor	Jl. Cikarang Baru No.2	Cikarang
5	S01031001	Ira Sulistyaningsih	Bekasi	Jl. Purna 11 No. 7	Tambun
6	I01031006	Yonita Veronika	Bekasi	Jl. Bunda Harapan No.36	Bekasi Barat



Tampilkan nama_m,j_kelamin (dari table mahasiswa), nama_jur,jenjang (dari table jurusan), dimana kolom nama_m diganti 'Nama Mahasiswa', kolom j_kelamin diganti 'Jenis Kelamin, kolom nama_jur diganti 'Jurusan', untuk mahasiswa yang j_kelamin nya 'Wanita' atau nama_jur = 'Teknik Informatika', maka perintah nya adalah:

```
SELECT  nama_m 'Nama Mahasiswa', j_kelamin  
         'Jenis Kelamin', nama_jur  
         'Jurusan', jenjang  
  
FROM    Mahasiswa INNER JOIN Jurusan ON  
         Mahasiswa.kode_jur=Jurusan.kode_jur  
  
WHERE   j_kelamin='Wanita' OR  
         nama_jur='Teknik Informatika'
```



LOGICAL OPERATOR OR LANJUTAN...

	Nama Mahasiswa	Jenis Kelamin	Jurusan	jenjang
1	Lestary Ramiati	Wanita	Manajemen Informatika	Diploma 3
2	Ani Lusiamah	Wanita	Manajemen Informatika	Diploma 3
3	Dewi Murbaini	Wanita	Teknik Informatika	Strata 1
4	Deni Hermawan	Pria	Teknik Informatika	Strata 1
5	Fitria Choirunissa	Wanita	Teknik Komputer	Diploma 3
6	Rini Hapsari	Wanita	Teknik Informatika	Strata 1
7	Sylvia Dwita Ningrum	Wanita	Teknik Informatika	Strata 1
8	Ira Sulistyaningsih	Wanita	Sistem Informasi	Strata 1
9	Susilo Mahono	Pria	Teknik Informatika	Strata 1
10	Yonita Veronika	Wanita	Teknik Informatika	Strata 1
11	Yuni Nurvati	Wanita	Teknik Komputer	Diploma 3

Operator ini digunakan untuk mendapatkan nilai kebalikan dari suatu logika atau ekspresi. Dengan demikian akan dihasilkan nilai TRUE hanya jika nilai ekspresi bernilai FALSE, serta nilai akan menghasilkan NULL jika ekspresi memiliki nilai NULL, dengan aturan penulisan:

NOT Ekspresi

Operator NOT pada SQL akan menghasilkan nilai sesuai dengan table

berikut:

Ekspresi	Hasil
True	False
False	True
Null	Null



a. Untuk 1 (satu) *table*

Tampilkan `nim,nama_m,tpt_lhr_m,alm_m,kota_m,` untuk kolom `nama_m` diganti 'Nama Mahasiswa', kolom `tpt_lhr_m` diganti 'Tempat Lahir', kolom `alm_m` diganti 'Alamat' dan kolom `kota_m` diganti 'Kota'. Dimana mahasiswa yang tempat lahirnya bukan di 'Bekasi', 'Cikarang', 'Jakarta' dan 'Bogor', maka perintah nya adalah:



```
SELECT  nim,nama_m 'Nama Mahasiswa`  
          ,tpt_lhr_m 'Tempat Lahir`  
          ,alm_m 'Alamat',kota_m 'Kota`  
  
FROM    Mahasiswa  
  
WHERE    tpt_lhr_m NOT in  
          ('Bekasi','Cikarang','Jakarta',  
          'Bogor')
```



OPERATOR NOT LANJUTAN...

	nim	Nama Mahasiswa	Tempat Lahir	Alamat	Kota
1	E01031001	Ahmad Sofyan	Surabaya	Jl. SMP I Nurul Huda No.4	Bekasi Timur
2	E01031002	Ani Lusiamah	Bandung	Jl. Bulak Damai 3 No. 34	Cibinong
3	E01031002	Daffa Alrozaq	Solo	Jl. Muara Indah II No.1	Bekasi Barat
4	T01031001	Chandra Khirana	Cirebon	Jl. Masjid Al-Hidayah No.7	Bekasi Barat
5	I01031004	Sylvia Dwita Ningrum	Semarang	Jl. Sultan Agung No.23	Bekasi Barat
6	S01031002	Yedi Ridwan	Subang	Jl. Kebun Kelapa I No.3	Cibitung



b. Untuk join *table*

Tampilkan nama_m (dari table mahasiswa), nama_jur,jenjang (dari table jurusan), dimana kolom nama_m diganti 'Nama Mahasiswa', kolom nama_jur diganti 'Jurusan', untuk mahasiswa yang nama_jurusannya bukan 'Teknik Informatika' dan 'Teknik Komputer', maka perintah nya adalah:



```
SELECT  nama_m  'Nama Mahasiswa',  nama_jur  
          'Jurusan', jenjang  
  
FROM    nama_m  'Nama Mahasiswa',  nama_jur  
          'Jurusan', jenjang  
  
WHERE    Mahasiswa INNER JOIN Jurusan ON  
          Mahasiswa.kode_jur=Jurusan.kode_jur  
          WHERE nama_jur NOT in('Teknik  
          Informatika', 'Teknik Komputer')
```



OPERATOR NOT LANJUTAN...

	Nama Mahasiswa	Jurusan	jenjang
1	Lestary Ramianti	Manajemen Informatika	Diploma 3
2	Ahmad Sofyan	Komputerisasi Akuntansi	Diploma 3
3	Ani Lusiamah	Manajemen Informatika	Diploma 3
4	Daffa Alrozak	Komputerisasi Akuntansi	Diploma 3
5	Ira Sulistyaningsih	Sistem Informasi	Strata 1
6	Yudi Ridwan	Sistem Informasi	Strata 1



OPERATOR AND

Operator ini digunakan untuk menguji beberapa ekspresi logika yang diberikan memiliki nilai TRUE, dengan aturan penulisan:

`Ekspres1 AND Ekspresi2`

Operator AND pada SQL akan menghasilkan nilai sesuai dengan table berikut:

Ekspresi1	Ekspresi2	Hasil
True	True	True
True	False	False
False	True	False
False	False	False
Null	True/False/Null	Null
True/False/Null	Null	Null



a. Untuk 1 (satu) *table*

Tampilkan `nim,nama_m,tpt_lhr_m,j_kelamin` untuk kolom `nama_m` diganti 'Nama Mahasiswa', kolom `tpt_lhr_m` diganti 'Tempat Lahir' dan kolom `j_kelamin` diganti 'Jenis Kelamin'. Dimana mahasiswa yang tempat lahirnya di 'Bekasi' dan jenis kelaminnya adalah 'Wanita', maka perintah nya adalah:



```
SELECT  nim,nama_m 'Nama Mahasiswa`  
          ,tpt_lhr_m 'Tempat Lahir`  
          ,j_kelamin 'Jenis Kelamin`  
  
FROM    Mahasiswa  
  
WHERE    tpt_lhr_m='Bekasi' AND  
          j_kelamin='Wanita'
```



OPERATOR AND LANJUTAN...

	nim	Nama Mahasiswa	Tempat Lahir	Jenis Kelamin
1	I01031001	Dewi Nurbaini	Bekasi	Wanita
2	T01031002	Fitria Choirunissa	Bekasi	Wanita
3	S01031001	Ira Sulistyaningsih	Bekasi	Wanita
4	I01031006	Yonita Veronika	Bekasi	Wanita



b. Untuk join *table*

Tampilkan nama_m,j_kelamin (dari table mahasiswa), nama_jur,jenjang (dari table jurusan), dimana kolom nama_m diganti 'Nama Mahasiswa', kolom j_kelamin diganti 'Jenis Kelamin', kolom nama_jur diganti 'Jurusan', untuk mahasiswa yang jenis kelaminnya 'Wanita' dan nama_jurusannya 'Teknik Informatika' dan 'Teknik Komputer', maka perintah nya adalah:



```
SELECT  nama_m 'Nama Mahasiswa`  
          ,j_kelamin 'Jenis Kelamin`  
          ,nama_jur 'Jurusan',jenjang  
  
FROM    Mahasiswa  
  
WHERE    INNER JOIN Jurusan ON  
          Mahasiswa.kode_jur=Jurusan.kode_jur  
          WHERE j_kelamin='Wanita' AND  
          nama_jur IN('Teknik Informatika`  
          , 'Teknik Komputer')
```



OPERATOR AND LANJUTAN...

	Nama Mahasiswa	Jenis Kelamin	Jurusan	jenjang
1	Dewi Nurbaini	Wanita	Teknik Informatika	Strata 1
2	Fitria Choirunissa	Wanita	Teknik Komputer	Diploma 3
3	Rini Hapsari	Wanita	Teknik Informatika	Strata 1
4	Sylvia Dwita Ningrum	Wanita	Teknik Informatika	Strata 1
5	Yonita Veronika	Wanita	Teknik Informatika	Strata 1
6	Yuni Nurvati	Wanita	Teknik Komputer	Diploma 3



MENGUJI NILAI NULL

Unknown Values atau lebih dikenal dengan Null, akan diberikan apabila tidak ada input data. Nilai Null dapat ditampilkan dari tabel dengan menggunakan kunci Is Null dalam Klausa Where.

Sintaks untuk ekspresi ini adalah sebagai berikut:

```
SELECT   Daftar_kolom  
  
FROM     nama_tabel  
  
WHERE    nama_kolom . operator_Unknown_Values
```




MENGUJI NILAI NULL LANJUTAN...

Tampilkan nama_m,j_kelamin (dari table mahasiswa), nama_jur,jenjang (dari table jurusan), dimana kolom nama_m diganti 'Nama Mahasiswa', kolom j_kelamin diganti 'Jenis Kelamin', kolom nama_jur diganti 'Jurusan'. Dimana pada kolom jenjang berisi NULL, maka perintah nya adalah:



MENGUJI NILAI NULL LANJUTAN...

```
SELECT  nama_m 'Nama Mahasiswa',j_kelamin  
         'Jenis Kelamin',nama_jur  
         'Jurusan',jenjang  
  
FROM    Mahasiswa INNER JOIN Jurusan ON  
         Mahasiswa.kode_jur=Jurusan.kode_jur  
  
WHERE    jenjang IS NULL
```

Nama Mahasiswa	Jenis Kelamin	Jurusan	jenjang
----------------	---------------	---------	---------



FUNGSI AGREGATE

Terdapat beberapa fungsi agregate atau fungsi untuk melakukan suatu operasi terhadap himpunan data yang dapat digunakan dalam perintah query. Hasil dari penggunaan fungsi ini merupakan nilai perolehan yang dapat diolah lebih lanjut.



FUNGSI AGREGATE

Fungsi	Penjelasan
SUM	Menjumlahkan Nilai
MIN	Mencari nilai minimum
MAX	Mencari Nilai Maximum
AVG	Mencari Nilai Rata rata
COUNT	Menghitung jumlah data

Fungsi AVG, SUM, MIN, MAX dan COUNT meng *Ignore* nilai *Null* sedangkan fungsi COUNT() menghitung nilai Null



FUNGSI AGREGATE - SUM

Fungsi ini mempunyai bentuk SUM(X), digunakan untuk menghasilkan nilai penjumlahan dari suatu data bilangan, dimana X adalah nilai numerik atau nama field yang memiliki nilai numerik atau rumus yang menghasilkan nilai numerik.

Tampilkan kolom 'Total SKS Semester 1' dari table Matakuliah dan hitung jumlah sks nya pada semester 1, maka perintah nya adalah:

SELECT SUM(sks) 'Total SKS Semester 1'

FROM Matakuliah

WHERE semester=1

	Total SKS Semester 1
1	20



FUNGSI AGREGATE - MIN

Fungsi ini mempunyai bentuk MIN(X), digunakan untuk menghasilkan nilai terkecil dari suatu data bilangan, dimana X adalah nilai numerik atau nama field yang memiliki nilai numerik atau rumus yang menghasilkan nilai numerik.

Tampilkan kolom 'SKS Terkecil Semester 2' dari table Matakuliah pada semester 2, maka perintah nya adalah:

```
SELECT MIN(sks) 'SKS Terkecil Semester 2'  
  
FROM Matakuliah  
  
WHERE semester=2
```

	SKS Terkecil Semester 2
1	1



FUNGSI COUNT

Fungsi ini mempunyai bentuk COUNT(field), digunakan untuk menghitung banyaknya data dalam suatu table dari hasil query. Dengan fungsi ini dapat dihitung jumlah data yang diperoleh atas dasar field tertentu atau seluruh field pada query. Dengan menyertakan nama field, nilai NULL pada field yang disebutkan tidak akan dimasukkan dalam kalkulasi.

Tampilkan kolom 'Jumlah Seluruh Mahasiswa' dari table Mahasiswa, maka perintah nya adalah:

SELECT 'Jumlah Seluruh Mahasiswa'=count(nim)

FROM Matakuliah

	Jumlah Seluruh Mahasiswa
1	16



FUNGSI TANGGAL dan WAKTU

Fungsi tanggal merupakan fungsi yang diperuntukkan bagi data, variable atau field yang berkaitan dengan data tanggal. Demikian pula untuk fungsi waktu yaitu fungsi yang diperuntukkan bagi data, variable atau field yang berkaitan dengan data waktu. Fungsi tanggal dan waktu dapat pula diterapkan untuk data dari tanggal sistem komputer yang sedang aktif.



FUNGSI TANGGAL dan WAKTU LANJUTAN...

Fungsi	Sintaks	Keterangan
DATEADD	(Datepart,number, Date)	Menambah datepart ke tanggal sesuai dengan besar number
DATEDIFF	(Datepart, date, date2)	Mengkalkulasi angka datepart antara dua tanggal
DATEPART	(Datepart, date)	Mengembalikan datepart dari daftar tanggal sebagai integer
DATENAME	(Datepart, date)	Mengembalikan datepart dari daftar tanggal sebagai nilai Ascii (contoh October)
GetDate	()	Tanggal dan Waktu sekarang



Komponen tanggal yang disebut ***Datepart*** digunakan untuk menentukan elemen nilai tanggal untuk Aritmatika tanggal. Datepart dapat berupa :

Datepart		Singkatan	Nilai	Keteangan
Year	Tahun	YY	1753-9999	8246 Thn
Quarter	Kwartel	QQ	1-4	4 Kwartal 1thn
Month	Bulan	MM	1-12	12 bln setahun
Day Of Year	Hari ke	DY	1-366	366 hari setahun
Day	Tanggal	DD	1-31	31 hari sebulan
Week	Minggu	WW	0-51	52 mingg se thn
Weekday	Hari	DW	1-7 (1=Sunday)	7 hari seminggu



FUNGSI TANGGAL dan WAKTU LANJUTAN...

Datepart		Singkatan	Nilai	Keteangan
Hour	Jam	HH	0-23	24 jam sehari
Minute	Menit	MI	(0-59)	60 mnt sejam
Second	Detik	SS	(0-59)	60 detik semenit
Milisecond	Milidetik	MS	(0-999)	1000 mili sedetik



FUNGSI DATEADD

Fungsi ini mempunyai bentuk `DateAdd(DatePart,Number,Date)`, merupakan fungsi yang menambah datepart ke tanggal sesuai dengan besar number yang diberikan (DD,MM,YY).

1. DD

Tampilkan kolom `nama_m`, `tgl_lhr_m`, dan kolom '30 Hari Setelah Tanggal Lahir' dari table Mahasiswa dimana mahasiswa yang tempat lahirnya 'Bogor', maka perintah nya adalah:



FUNGSI DATEADD

```
SELECT  nama_m, tgl_lhr_m, "30 Hari Setelah  
         Tanggal Lahir"=DATEADD  
         dd,30,tgl_lhr_m)  
  
FROM    Mahasiswa  
  
WHERE    tpt_lhr_m='Bogor'
```

	nama_m	tgl_lhr_m	30 Hari Setelah Tanggal Lahir
1	Lestary Ramiati	1976-01-10 ...	1976-02-09 00:00:00.000
2	Rini Hapsari	1981-08-18 ...	1981-09-17 00:00:00.000



2. MM

Tampilkan kolom nama_m, tgl_lhr_m, dan kolom '5 Bulan Setelah Tanggal Lahir' dari table Mahasiswa dimana mahasiswa yang tempat lahirnya 'Jakarta', maka perintah nya adalah:

```
SELECT  nama_m, tgl_lhr_m, "5 Bulan Setelah  
          TanggalLahir"=DATEADD  
          mm, 5, tgl_lhr_m)  
  
FROM    Mahasiswa  
  
WHERE    tpt_lhr_m='Jakarta'
```



FUNGSI DATEADD LANJUTAN...

	nama_m	tgl_lhr_m	5 Bulan Setelah Tanggal Lahir
1	Deni Hermawan	1980-01-17 00:00:00.000	1980-06-17 00:00:00.000
2	Muhammad Fahrurozi	1980-10-13 00:00:00.000	1981-03-13 00:00:00.000
3	Susilo Wahono	1979-12-21 00:00:00.000	1980-05-21 00:00:00.000
4	Yuni Nurwati	1978-11-14 00:00:00.000	1979-04-14 00:00:00.000



3. YY

Tampilkan kolom nama_m, tgl_lhr_m, dan kolom '5 Tahun Setelah Tanggal Lahir' dari table Mahasiswa dimana mahasiswa yang tempat lahirnya 'Jakarta' dan 'Bekasi', maka perintah nya adalah:

```
SELECT  nama_m, tgl_lhr_m, "5 Tahun Setelah  
        Tanggal  
        Lahir"=DATEADD(yy, 5, tgl_lhr_m)  
  
FROM    Mahasiswa  
  
WHERE    tpt_lhr_m in ('Jakarta', 'Bekasi')
```




FUNGSI DATEADD LANJUTAN...

	nama_m	tgl_lhr_m	5 Tahun Setelah Tanggal Lahir
1	Dewi Nurbaini	1987-12-10 00:00:00.000	1992-12-10 00:00:00.000
2	Deni Hermawan	1980-01-17 00:00:00.000	1985-01-17 00:00:00.000
3	Fitria Choirunissa	1979-01-21 00:00:00.000	1984-01-21 00:00:00.000
4	Ira Sulistyaningsih	1979-12-01 00:00:00.000	1984-12-01 00:00:00.000
5	Muhammad Fahrurozi	1980-10-13 00:00:00.000	1985-10-13 00:00:00.000
6	Susilo Wahono	1979-12-21 00:00:00.000	1984-12-21 00:00:00.000
7	Yonita Veronika	1980-10-24 00:00:00.000	1985-10-24 00:00:00.000
8	Yuni Nurwati	1978-11-14 00:00:00.000	1983-11-14 00:00:00.000



FUNGSI DATEIIF

Fungsi ini mempunyai bentuk DateDIFF(DatePart,Date,Date2), merupakan fungsi yang melakukan kalkulasi angka datepart antara dua tanggal yang diberikaan (DD,MM,YY).

1. DD

Tampilkan kolom nama_m,tgl_lhr_m, 'Tanggal dan Waktu Sekarang', 'Jumlah Hari' dari table Mahasiswa dimana mahasiswa yang tempat lahirnya 'Jakarta', maka perintah nya adalah:



FUNGSI DATEIIF LANJUTAN...

```
SELECT nama_m, tgl_lhr_m, getdate() 'Tanggal  
dan Waktu Sekarang', "Jumlah Hari"  
=DATEDIFF(dd, tgl_lhr_m, getdate())  
  
FROM Mahasiswa  
  
WHERE tpt_lhr_m='Jakarta'
```

	nama_m	tgl_lhr_m	Tanggal dan Waktu Sekarang	Jumlah Hari
1	Deni Hermawan	1980-01-17 00:00:00.000	2006-06-07 12:59:16.043	9638
2	Muhammad Fahrurozi	1980-10-13 00:00:00.000	2006-06-07 12:59:16.043	9368
3	Susilo Wahono	1979-12-21 00:00:00.000	2006-06-07 12:59:16.043	9665
4	Yuni Nurwati	1978-11-14 00:00:00.000	2006-06-07 12:59:16.043	10067



2. MM

Tampilkan kolom nama_m, tgl_lhr_m, 'Tanggal dan Waktu Sekarang', 'Jumlah Bulan' dari table Mahasiswa dimana mahasiswa yang tempat lahirnya 'Jakarta', maka perintah nya adalah:

```
SELECT  nama_m, tgl_lhr_m, getdate() 'Tanggal  
        dan Waktu Sekarang', "Jumlah Bulan"  
        =DATEDIFF (mm, tgl_lhr_m, getdate())  
  
FROM    Mahasiswa  
  
WHERE    tpt_lhr_m='Jakarta'
```



FUNGSI DATEIF LANJUTAN...

	nama_m	tgl_lhr_m	Tanggal dan Waktu Sekarang	Jumlah Bulan
1	Deni Hermawan	1980-01-17 00:00:00.000	2006-06-07 13:03:33.203	317
2	Muhammad Fahrurrozi	1980-10-13 00:00:00.000	2006-06-07 13:03:33.203	308
3	Susilo Wahono	1979-12-21 00:00:00.000	2006-06-07 13:03:33.203	318
4	Yuni Nurwati	1978-11-14 00:00:00.000	2006-06-07 13:03:33.203	331



3. YY

Tampilkan kolom nama_m, tgl_lhr_m, 'Tanggal dan Waktu Sekarang', 'Jumlah Tahun' dari table Mahasiswa dimana mahasiswa yang tempat lahirnya 'Jakarta', maka perintah nya adalah:

```
SELECT  nama_m, tgl_lhr_m, getdate() 'Tanggal
          dan Waktu Sekarang', "Jumlah Tahun"
          =DATEDIFF (yy, tgl_lhr_m, getdate())

FROM    Mahasiswa

WHERE    tpt_lhr_m='Jakarta'
```



FUNGSI DATEIF LANJUTAN...

	nama_m	tgl_lhr_m	Tanggal dan Waktu Sekarang	Jumlah Tahun
1	Deni Hermawan	1980-01-17 00:00:00.000	2006-06-07 13:09:40.323	26
2	Muhammad Fahrurozi	1980-10-13 00:00:00.000	2006-06-07 13:09:40.323	26
3	Susilo Wahono	1979-12-21 00:00:00.000	2006-06-07 13:09:40.323	27
4	Yuni Nurwati	1978-11-14 00:00:00.000	2006-06-07 13:09:40.323	28

Fungsi ini mempunyai bentuk `DatePart(DatePart,Date)`, merupakan fungsi yang mengembalikan datepart dari daftar tanggal sebagai integer sesuai tanggal yang diberikaan (DD,MM,YY).

1. DD

Tampilkan kolom `nama_m,tgl_lhr_m`, 'Tanggal Lahir' dari table Mahasiswa dimana mahasiswa yang tempat lahirnya 'Bogor', maka perintah nya adalah:



FUNGSI DATEPART LANJUTAN...

```
SELECT  nama_m, tgl_lhr_m, "Tanggal  
         Lahir"=DATEPART(dd,tgl_lhr_m)  
  
FROM    Mahasiswa  
  
WHERE    tpt_lhr_m='Bogor'
```

	nama_m	tgl_lhr_m	Tanggal Lahir
1	Lestary Ramianti	1976-01-10 00:00:00.000	10
2	Rini Hapsari	1981-08-18 00:00:00.000	18



2. MM

Tampilkan kolom nama_m, tgl_lhr_m, 'Bulan Lahir' dari table Mahasiswa dimana mahasiswa yang tempat lahirnya 'Bogor', maka perintah nya adalah:

```
SELECT    nama_m, tgl_lhr_m, "Bulan  
            Lahir"=DATEPART(mm, tgl_lhr_m)  
  
FROM      Mahasiswa  
  
WHERE      tpt_lhr_m='Bogor'
```



FUNGSI DATEPART LANJUTAN...

	nama_m	tgl_lhr_m	Bulan Lahir
1	Lestary Ramiati	1976-01-10 00:00:00.000	1
2	Rini Hapsari	1981-08-18 00:00:00.000	8



3. YY

Tampilkan kolom nama_m, tgl_lhr_m, 'Tahun Lahir' dari table Mahasiswa dimana mahasiswa yang tempat lahirnya 'Bogor', maka perintah nya adalah:

```
SELECT    nama_m, tgl_lhr_m, "Tahun  
            Lahir"=DATEPART(yy, tgl_lhr_m)  
  
FROM      Mahasiswa  
  
WHERE     tpt_lhr_m='Bogor'
```



FUNGSI DATEPART LANJUTAN...

	nama_m	tgl_lhr_m	Tahun Lahir
1	Lestary Ramiati	1976-01-10 00:00:00.000	1976
2	Rini Hapsari	1981-08-18 00:00:00.000	1981

FUNGSI DATENAME

Fungsi ini mempunyai bentuk `DateName(DatePart,Date)`, merupakan fungsi yang mengembalikan datepart dari daftar tanggal sebagai nilai ASCII (misal October), fungsi ini hanya berfungsi untuk bulan (MM).

Tampilkan kolom `nama_m,tgl_lhr_m`, 'Nama Bulan Lahir' dari table Mahasiswa dimana mahasiswa yang tempat lahirnya 'Jakarta', maka perintah nya adalah:



FUNGSI DATENAME LANJUTAN....

```
SELECT  nama_m, tgl_lhr_m, "Nama Bulan  
Lahir"=DATENAME(mm,tgl_lhr_m)  
  
FROM    Mahasiswa  
  
WHERE    tpt_lhr_m='Jakarta'
```

	nama_m	tgl_lhr_m	Nama Bulan Lahir
1	Deni Hermawan	1980-01-17 00:00:00.000	January
2	Muhammad Fahrurozi	1980-10-13 00:00:00.000	October
3	Susilo Wahono	1979-12-21 00:00:00.000	December
4	Yuni Nurwati	1978-11-14 00:00:00.000	November



FUNGSI GETDATE

Fungsi ini mempunyai bentuk GetDate(), merupakan fungsi yang digunakan untuk mengetahui tanggal dan waktu sekarang.

Tampilkan kolom 'Tanggal Hari Ini' dan '15 Hari Dari Sekarang', maka perintah nya adalah:

```
SELECT  "Tanggal Hari Ini"=GetDate(), "15  
        Hari Dari Sekarang"=Getdate()+5
```

	Tanggal Hari Ini	15 Hari Dari Sekarang
1	2006-06-07 13:54:13.203	2006-06-12 13:54:13.203



Key Word Distinct menghilangkan duplikasi pada baris baris sebagai Default, Query menampilkan semua baris termasuk duplikasinya. Baris-baris yang memiliki duplikasi dapat dieleminasi dengan menggunakan key word Distinct dalam pernyataan Select .

Sintaks dalam pernyataan ini adalah:

```
SELECT [ All | Distinct ]  nama_kolom  
FROM                      Nama_tabel  
WHERE                     kondisi
```



DISTINC LANJUTAN....

Tampilkan kolom kota_m dari table Mahasiswa, dimana kolom kota_m diganti menjadi 'Kota Alamat Mahasiswa' dengan Distinct, maka perintah nya adalah:

```
SELECT Distinct      kota_m 'Kota Alamat  
                        Mahasiswa`  
  
FROM                Mahasiswa
```

	Kota Alamat Mahasiswa
1	Bekasi Barat
2	Bekasi Timur
3	Bekasi Utara
4	Bogor
5	Cibinong
6	Cibitung
7	Cikarang
8	Jakarta Timur
9	Tambun

Predikat ini digunakan untuk mengambil sejumlah record yang berada pada jangkauan atas atau bawah dari seluruh data yang diperoleh. Argumen numerik yang diberikan pada predikat Top defaultnya dalam bentuk jumlah record pada jangkauan atas yang ditampilkan. Sedang untuk mengambil n percent dari sejumlah record yang didapat, gunakan argumen Percent.

Tampilkan kolom nim,nama_m,telpon_m dari table Mahasiswa, dimana kolom kota_m diganti menjadi 'Kota Alamat Mahasiswa' dengan data yang ditampilkan 7 record pertama, maka perintah nya adalah:



TOP LANJUTAN....

```
SELECT TOP 7 nim,nama_m 'Nama Mahasiswa`  
            ,telpon_m FROM Mahasiswa  
  
FROM        Mahasiswa
```

	nim	Nama Mahasiswa	telpon_m
1	M01031001	Lestary Ramiati	0251-456767
2	K01031001	Ahmad Sofyan	021-8789991
3	M01031002	Ani Lusiamah	0251-343311
4	K01031002	Daffa Alrozak	021-8781199
5	T01031001	Chandra Khirana	021-8712343
6	I01031001	Dewi Nurbaini	021-8791290
7	I01031002	Deni Hermawan	021-9897119



OPERATOR JOIN

SQL memiliki metode untuk mengambil data dari suatu tabel dengan menggunakan Join. Join dipakai untuk mencari data dari beberapa tabel berdasarkan hubungan yang logis dari tabel-tabel tersebut. Join menyatakan cara SQL memakai data dari sebuah tabel untuk memilih data dari tabel lain. Sintaks untuk melakukan operasi join ini adalah:

SELECT nama_kolom, nama_kolom, [nama_kolom]

FROM nama_tabel, [**Cross / Inner / [Left**
 / Right] Outer]

JOIN nama_tabel On
 nama_tabel.Ref_nama_kolom

OPERATOR JOIN Nama_tabel.Ref_nama_kolom



OPERATOR JOIN LANJUTAN...

Keterangan :

- Nama Kolom** : menspesidikasikan nama kolom dari satu atau beberapa tabel yang ditampilkan.
- Nama Tabel** : adalah nama tabel dari tabel yang diambil
- Ref.nama_kolom** : adalah nama_kolom yang digunakan menggabungkandua tabel dengan menggunakan kunci yang umum. Operator Join menspesifikasikan operator yang digunakan untuk menggabungkan tabel.

Join dapat dikelompokkan menjadi Inner Join danOuter Join.

INNER JOIN

Inner Join adalah sebuah Join yang menghasilkan baris-baris, minimal ada sebuah baris di kedua tabel yang sesuai dengan kondisi Join. Baris-baris yang tidak sesuai dengan sebuah baris dari tabel lain tidak ditampilkan, defaultnya adalah Inner Join.

Tampilkan nama_m (dari table mahasiswa),nama_mk,semester (dari table Matakuliah),nilai_uts,nilai_uas(dari table Nilai), dimana mahasiswa yang kode_jur ='MI' dan nilainya pada semester=1, maka perintahnya adalah:



INNER JOIN LANJUTAN...

```
SELECT      nama_m,nama_mk,semester,nilai_uts  
              ,nilai_uas  
  
FROM        Nilai INNER JOIN Mahasiswa ON  
              Nilai.nim=Mahasiswa.nim  
  
INNER JOIN  Matakuliah ON Nilai.kdmk  
              =Matakuliah.kdmk  
  
WHERE       Where semester=1 and  
              kode_jur='MI'
```

	nama_m	nama_mk	semester	nilai_uts	nilai_uas
1	Lestary Ramiati	PENDIDIKAN AGAMA I	1	89	55
2	Lestary Ramiati	PENDIDIKAN PANCASILA	1	87	76
3	Ani Lusiamah	PENDIDIKAN AGAMA I	1	78	87
4	Ani Lusiamah	PENDIDIKAN PANCASILA	1	90	90



OUTER JOIN

Outer Join akan menghasilkan semua data dari sebuah tabel dan membatasi data dari tabel lain. SQL mempunyai tiga buah tipe Outer Join yaitu Left, Right dan Full. Semua baris dari tabel sebelah kiri diacu oleh sebuah left outer join, dan sebuah baris dari tabel kanan diacu dengan Right outer join. Kegunaan dari outer join adalah mencari record-record piatu, artinya record yang ada disebuah tabel tetapi tidak ada pasangannya pada pada tabel lain.



LEFT OUTER JOIN

Tampilkan nama_m (dari table mahasiswa),nama_mk,semester (dari table Matakuliah),nilai_uts,nilai_uas(dari table Nilai), dimana mahasiswa yang kode_jur ='MI' dan nilainya pada semester=1, maka perintahnya adalah:

```
SELECT      nama_mk, semester, sks, nilai_uts, nilai_uas

FROM        Matakuliah LEFT OUTER JOIN Nilai
              ON Matakuliah.kdmk=Nilai.kdmk

WHERE       semester=2
```



LEFT OUTER JOIN LANJUTAN...

	nama_mk	semester	sks	nilai_uts	nilai_uas
1	PERANCANGAN BASIS DATA	2	4	NULL	NULL
2	PAKET PROGRAM APLIKASI II	2	1	NULL	NULL
3	ALGORITMA PEMROGRAMAN I (PASCAL)	2	3	NULL	NULL
4	PRAK. ALGORITMA PEMROGRAMAN I (PASCAL)	2	1	NULL	NULL
5	DASAR MANAJEMEN DAN BISNIS II	2	2	NULL	NULL
6	MATEMATIKA DISKRIT II	2	2	NULL	NULL
7	AKUTANSI DASAR II	2	2	NULL	NULL
8	PENDIDIKAN KEWARGANEGARAAN	2	2	78	89
9	PENDIDIKAN KEWARGANEGARAAN	2	2	66	89
10	PENDIDIKAN KEWARGANEGARAAN	2	2	78	67
11	PENDIDIKAN KEWARGANEGARAAN	2	2	90	76
12	PENDIDIKAN KEWARGANEGARAAN	2	2	57	80
13	BAHASA INGGRIS II	2	2	NULL	NULL



RIGHT OUTER JOIN

Tampilkan nama_mk, semester,sks (dari table Matakuliah), nilai_uts, nilai_uas (dari table Nilai), dimana sks matakuliahnya tidak sama dengan 2 (<>2), maka perintahnya adalah:

```
SELECT      MK.nama_mk,MK.semester,MK.sks,N.n  
              ilai_uts, N.nilai_uas  
  
FROM        Nilai  N      RIGHT  OUTER  JOIN  
              Matakuliah MK ON N.kdmk=MK.kdmk  
  
WHERE       sks<>2
```



RIGHT OUTER JOIN LANJUTAN...

	nama_mk	semester	sks	nilai_uts	nilai_uas
1	PENGANTAR TEKNOLOGI INFORMASI	1	3	NULL	NULL
2	PRAKTIKUM PAKET PROGRAM APLIKASI I	1	1	NULL	NULL
3	PERANCANGAN BASIS DATA	2	4	NULL	NULL
4	PAKET PROGRAM APLIKASI II	2	1	NULL	NULL
5	ALGORITMA PENROGRAMAN I (PASCAL)	2	3	NULL	NULL
6	PRAK. ALGORITMA PENROGRAMAN I (PASCAL)	2	1	NULL	NULL



FULL OUTER JOIN

Tampilkan nama_mk, semester, sks (dari Table Matakulliah(, nilai_uts, nilai_uas (dari table Nilai), dimana matakuliah yang berada pada semester 2, maka perintahnya adalah:

```
SELECT      MK.nama_mk,MK.semester,MK.sks,  
              N.nilai_uts,N.nilai_uas  
  
FROM        Nilai N  FULL OUTER JOIN  
              Matakuliah MK ON N.kdmk=MK.kdmk  
  
WHERE       semester=2
```



FULL OUTER JOIN LANJUTAN...

	nama_mk	semester	sks	nilai_uts	nilai_uas
1	PERANCANGAN BASIS DATA	2	4	NULL	NULL
2	PAKET PROGRAM APLIKASI II	2	1	NULL	NULL
3	ALGORITMA PEMROGRAMAN I (PASCAL)	2	3	NULL	NULL
4	PRAK. ALGORITMA PEMROGRAMAN I (PASCAL)	2	1	NULL	NULL
5	DASAR MANAJEMEN DAN BISNIS II	2	2	NULL	NULL
6	MATEMATIKA DISKRIT II	2	2	NULL	NULL
7	AKUTANSI DASAR II	2	2	NULL	NULL
8	PENDIDIKAN KEWARGANEGARAAN	2	2	78	89
9	PENDIDIKAN KEWARGANEGARAAN	2	2	66	89
10	PENDIDIKAN KEWARGANEGARAAN	2	2	78	67
11	PENDIDIKAN KEWARGANEGARAAN	2	2	90	76
12	PENDIDIKAN KEWARGANEGARAAN	2	2	57	80
13	BAHASA INGGRIS II	2	2	NULL	NULL

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.



PENGELOMPOKAN DATA ORDER BY

Tampilkan nim,nama_m (dari table mahasiswa),nama_jur,jenjang (dari table Jurusan), dimana nama jurusanannya bukan 'Teknik Komputer dan diurutkan berdasarkan nama mahasiswa secara Descending, maka perintahnya adalah:

```
SELECT      nim,nama_m,nama_jur,jenjang

FROM        Mahasiswa INNER JOIN Jurusan ON
              Mahasiswa.kode_jur
              =Jurusan.kode_jur

WHERE       nama_jur <>'Teknik Komputer'
              ORDER BY nama_m DESC
```



PENGELOMPOKAN DATA ORDER BY (LANJUTAN..)

	nim	nama_m	nama_jur	jenjang
1	S01031002	Yudi Ridwan	Sistem Informasi	Strata 1
2	I01031006	Yonita Veronika	Teknik Informatika	Strata 1
3	I01031004	Sylvia Dwita Ningrum	Teknik Informatika	Strata 1
4	I01031005	Susilo Wahono	Teknik Informatika	Strata 1
5	I01031003	Rini Hapsari	Teknik Informatika	Strata 1
6	M01031001	Lestary Ramiati	Manajemen Informatika	Diploma 3
7	S01031001	Ira Sulistyaningsih	Sistem Informasi	Strata 1
8	I01031001	Dewi Nurbaini	Teknik Informatika	Strata 1
9	I01031002	Deni Hermawan	Teknik Informatika	Strata 1
10	K01031002	Daffa Alrozak	Komputerisasi Akuntansi	Diploma 3
11	M01031002	Ani Lusiamah	Manajemen Informatika	Diploma 3
12	K01031001	Ahmad Sofyan	Komputerisasi Akuntansi	Diploma 3



PENGELOMPOKAN DATA COMPUTE BY

Tampilkan nama_m (dari table Mahasiswa), nama_mk, semester, sks (dari table Jurusan), jumlahkan tiap semester jumlah sksnya dan diurutkan berdasarkan semester, maka perintahnya adalah:

```
SELECT      M.nama_m,MK.nama_mk,MK.semester,M
              K.sks, N.nilai_uts,N.nilai_uas

FROM        Nilai N INNER JOIN Mahasiswa M ON
              N.nim=M.nim
              INNER JOIN Matakuliah MK ON
              N.kdmk=MK.kdmk

ORDER BY    Semester

COMPUTE     SUM(sks) BY Semester
```



PENGELOMPOKAN DATA COMPUTE BY (LANJUTAN..)

	nama_m	nama_mk	semester	sks	nilai_uts	nilai_uas
1	Lestary Ramiati	PENDIDIKAN AGAMA I	1	2	89	55
2	Lestary Ramiati	PENDIDIKAN PANCASILA	1	2	87	76
3	Ani Lusiamah	PENDIDIKAN AGAMA I	1	2	78	87
4	Ani Lusiamah	PENDIDIKAN PANCASILA	1	2	90	90
5	Daffa Alrozak	PENDIDIKAN AGAMA I	1	2	56	56
6	Daffa Alrozak	PENDIDIKAN PANCASILA	1	2	78	78
7	Chandra Khirana	PENDIDIKAN AGAMA I	1	2	98	89
8	Chandra Khirana	PENDIDIKAN PANCASILA	1	2	88	78
9	Dewi Nurbaini	PENDIDIKAN AGAMA I	1	2	78	89
10	Dewi Nurbaini	PENDIDIKAN PANCASILA	1	2	67	78
11	Rini Hapsari	PENDIDIKAN AGAMA I	1	2	56	67
12	Rini Hapsari	PENDIDIKAN PANCASILA	1	2	45	78
13	Ira Sulistyaningsih	PENDIDIKAN AGAMA I	1	2	54	78
14	Ira Sulistyaningsih	PENDIDIKAN PANCASILA	1	2	67	88
	sum					
1	28					

	nama_m	nama_mk	semester	sks	nilai_uts	nilai_uas
1	Ira Sulistyaningsih	PENDIDIKAN KEWARGANEGARAAN	2	2	57	80
2	Rini Hapsari	PENDIDIKAN KEWARGANEGARAAN	2	2	78	89
3	Daffa Alrozak	PENDIDIKAN KEWARGANEGARAAN	2	2	66	89
4	Ani Lusiamah	PENDIDIKAN KEWARGANEGARAAN	2	2	90	76
5	Lestary Ramiati	PENDIDIKAN KEWARGANEGARAAN	2	2	78	67
	sum					
1	10					



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.

SUB QUERY -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        Where M.kdmk IN  
              (SELECT N.kdmk From Nilai N)
```

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



SUB QUERY – 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)
```



SUB QUERY – NOT IN LANJUTAN...

	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



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.

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)
```



ANY LANJUTAN....

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

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)
```



ALL LANJUTAN...

	Kode Matakuliah	Matakuliah	sks
1	MKB331205	PERANCANGAN BASIS DATA	4

