

BAB - 3

Fungsi - Fungsi SQL Server 2005

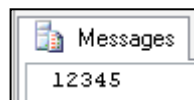
Pada pertemuan ini akan dibahas mengenai fungsi – fungsi SQL Server 2005 dengan mudah dan dapat dipahami baik secara teori maupun praktek.

I. Perintah PRINT

Salah satu perintah dalam SQL Server adalah PRINT, yang berfungsi untuk menampilkan sesuatu dalam bentuk teks biasa dari sebuah pernyataan. Jenis PRINT diikuti dengan apa yang akan ditampilkan.

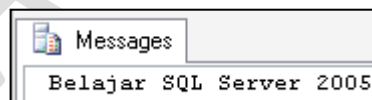
Contoh PRINT :

```
PRINT 12345;  
GO
```



Item yang ditampilkan dapat berupa operasi atau hasil dari operasi. Jika Anda ingin menampilkan karakter, kata, atau kalimat, memberikan tanda petik tunggal diantanya. Jika Anda ingin menyertakan petik tunggal dalam cetakan Anda, yaitu dapat menulis dua kali tanda petik tunggal. Ini adalah suatu contoh:

```
PRINT 'Belajar SQL Server 2005';  
GO
```



```
PRINT 'Hari Jum''at';  
GO
```



Anda juga dapat menampilkan ekspresi sebagai kombinasi angka dengan kalimat.

II. Perintah SELECT

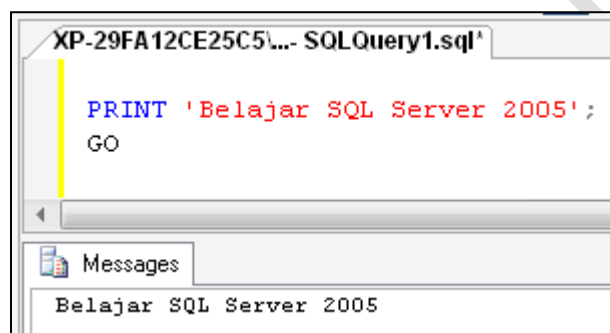
Perintah SELECT dapat digunakan antara lain untuk menampilkan nilai.

Contoh SELECT :

Perintah SELECT diikuti oleh angka, kata, string, atau ekspresi. Untuk menampilkan hal berikut, beberapa aturan yang sama seperti PRINT. Salah satu perbedaan antara PRINT dan SELECT adalah :

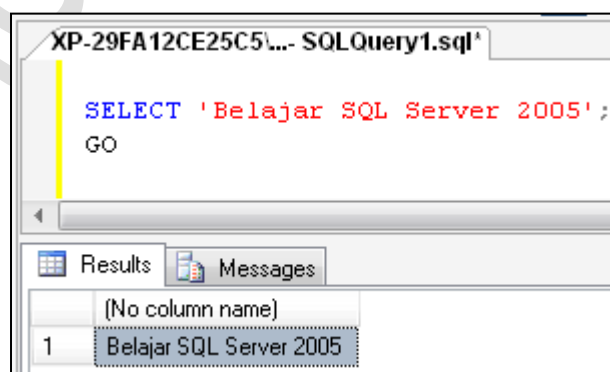
- **PRINT**

Perintah PRINT banyak digunakan untuk menguji nilai yang sederhana, string, atau ekspresi. Oleh karena itu, hasil yang ditampilkan pada tab dibawah berlabel *Messages*. PRINT dapat digunakan hanya dengan satu nilai.

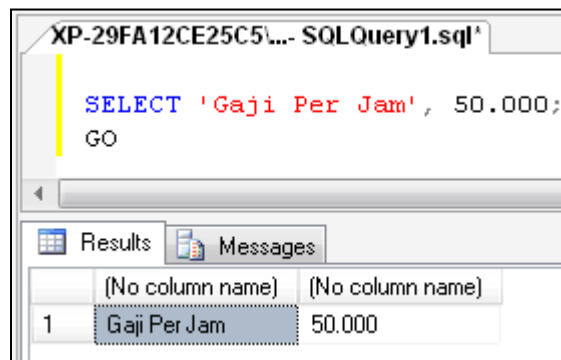


- **SELECT**

Perintah SELECT adalah yang paling sering digunakan operator SQL. Kita akan melihat bahwa hal tersebut digunakan untuk mengambil data dari tabel. Untuk alasan ini, hasil yang ditampilkan oleh SELECT terorganisir terdiri dari kategori yang disebut kolom, dibawah tab berlabel *Results*. SELECT dapat digunakan dengan lebih dari satu nilai.

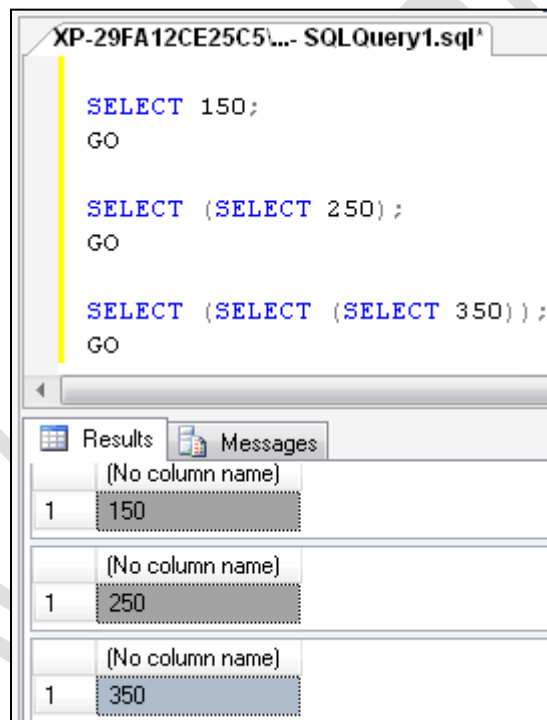


Perintah SELECT dapat digunakan untuk menampilkan lebih dari satu nilai. Nilai-nilai harus dipisahkan oleh koma. Berikut contohnya :



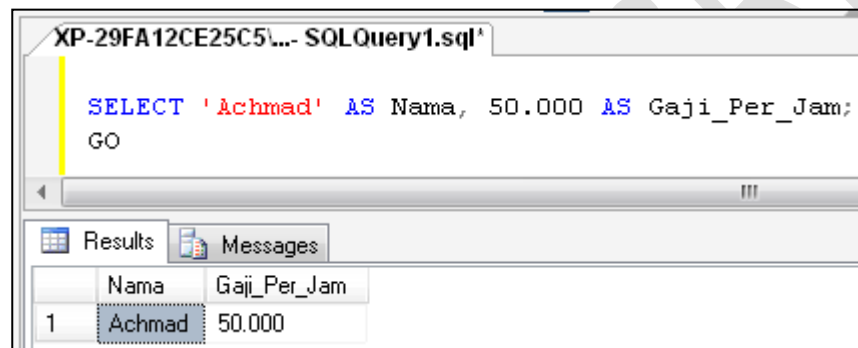
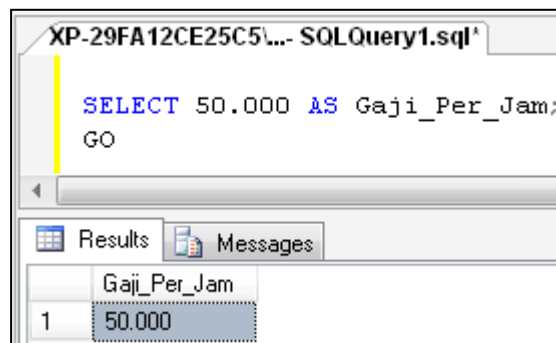
- **Perintah SELECT (Nested) yang bersarang**

Contoh :

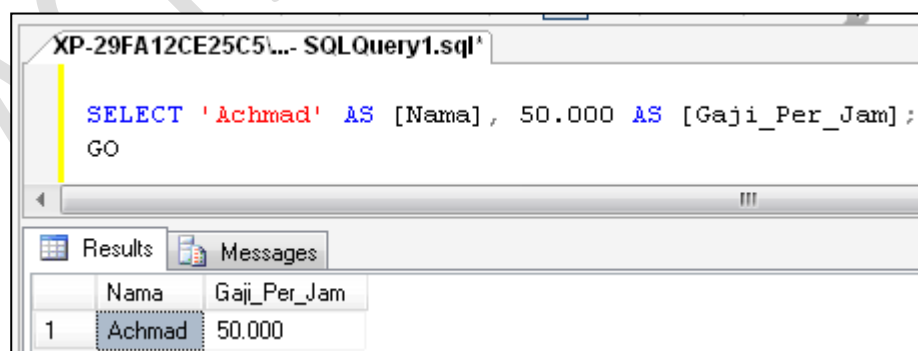


Dalam pengenalan di atas, dijelaskan perintah PRINT atau SELECT untuk menampilkan sesuatu pada jendela query. Salah satu karakteristik dari SELECT adalah dapat menghasilkan segmen pada bagian-bagian yang berbeda. SELECT masing-masing mewakili nilai dalam bagian yang disebut kolom. Setiap kolom diwakili dengan nama juga disebut sebagai keterangan. Secara default, menampilkan judul sebagai " (No column name)". Jika Anda ingin menggunakan keterangan/judul sendiri, disisi kanan dari sebuah ekspresi, ketik kata kunci AS diikuti oleh keterangan/judul yang diinginkan.

Item disisi kanan kata kunci AS harus dianggap sebagai satu kata. Berikut contohnya :



Jika item pada sisi kanan AS dalam kata-kata yang berbeda, Anda harus memasukkannya dalam tanda kutip tunggal atau menempatkannya didalam braket kurung buka persegi "[" dan tutup kurung persegi "]". Berikut contohnya :



III. Operator Aritmatika

Setiap database mendukung operator aritmatika. Berikut ini keterangan operator aritmatika dan derajat prioritas.

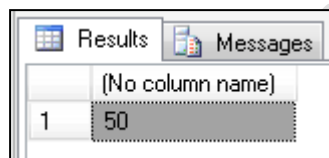
Tabel. Operator Aritmatika dan Derajat Prioritas

OPERATOR	KETERANGAN	PRIORITAS
*	Perkalian	Pertama
/	Pembagian	Kedua
%	Sisa Bagi (Modulus)	Ketiga
+	Penjumlahan	Keempat
-	Pengurangan	Keempat

1. Operator Perkalian

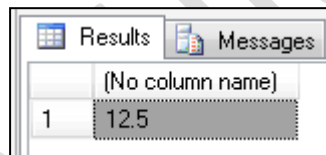
Operator yang digunakan untuk operasi perkalian adalah tanda asterix (*). Dengan menggunakan operator ini, dapat mengalikan dua buah nilai atau lebih. Berikut contoh perintahnya :

```
SELECT 5 * 10
```



(No column name)
50

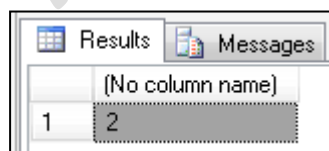
```
SELECT 2.5 * 5
```



(No column name)
12.5

2. Operator Pembagian

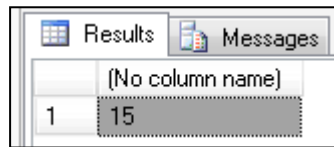
```
SELECT 10 / 5
```



(No column name)
2

3. Operator Penjumlahan

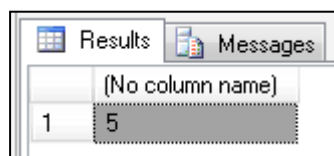
```
SELECT 5 + 10
```



	(No column name)
1	15

4. Operator Pengurangan

```
SELECT 10 - 5
```

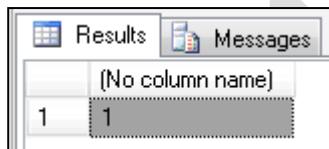


	(No column name)
1	5

5. Operator Modulus

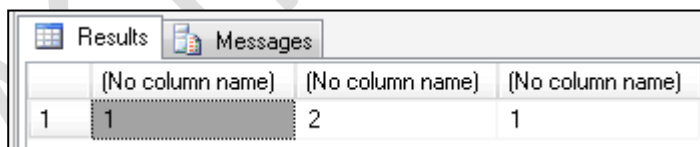
Fungsi modulus (%) digunakan untuk mendapatkan nilai sisa pembagian dari operasi pembagian.

```
SELECT 5 % 2
```



	(No column name)
1	1

```
SELECT 5 % 2, 5 % 3, 5 % 4
```



	(No column name)	(No column name)	(No column name)
1	1	2	1

IV. Fungsi Trigonometri

Dalam fungsi ini membahas tentang penggunaan fungsi-fungsi trigonometri diantaranya yaitu : Sinus, Cosinus, dan Tangen.

Dalam menghitung nilai trigonometri dari sudut yang mempunyai besaran dalam bentuk Radians.

Bekerja dengan fungsi trigonometri untuk T-SQL

Pertama set fungsi yang kita lihat adalah fungsi trigonometri - yang berkaitan dengan sudut, radian, derajat, dan nilai pi (π). Dua fungsi yang termasuk dalam kategori ini adalah SIN dan ASIN. Mengembalikan fungsi SIN yang sinus dari sudut tertentu. Fungsi ASIN (juga disebut sebagai fungsi arcsine) mengembalikan sebuah sinus sudut tertentu.

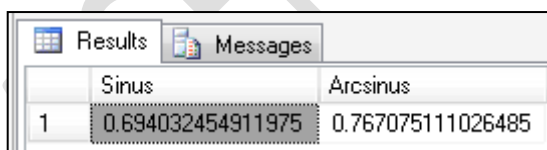
SIN dan ASIN

Sekarang mari kita lihat sebuah contoh bagaimana menggunakan SIN dan fungsi ASIN. Dalam deklarasi berikut menyatakan dan menetapkan dua variabel dan memanggil dengan perintah SELECT:

```
DECLARE @sudut1 float
SET @sudut1 = 52.64

DECLARE @sinus float
SET @sinus = SIN(@sudut1)

SELECT
    @sinus AS Sinus,
    ASIN(@sinus) AS Arcsinus
```



	Sinus	Arcsinus
1	0.694032454911975	0.767075111026485

Penjelasan :

Pertama, kita membuat sebuah variabel yang disebut @sudut1, yang dikonfigurasi dengan tipe data float. Lalu, kita tetapkan variabel sudut1 berisi nilai 52,64.

Selanjutnya, kita membuat sebuah variabel yang disebut @sinus, yang juga merupakan nilai float, dan menetapkan nilai variabel @sinus dengan menggunakan fungsi SIN untuk menentukan sinus dari variabel @sudut1. Fungsi SIN mensyaratkan bahwa ekspresi itu berlalu dalam sebagai argumen adalah nilai float atau nilai numerik yang dapat secara implisit diubah menjadi float.

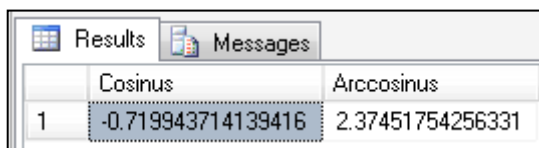
COS dan ACOS

Sebuah set fungsi yang mirip dengan SIN dan ASIN adalah COS dan fungsi ACOS. Mengembalikan fungsi COS dengan cosinus dari sudut tertentu dan fungsi ACOS (disebut sebagai fungsi arccosinus) mengembalikan sebuah kosinus sudut tertentu. Contoh berikut menunjukkan dua fungsi:

```
DECLARE @sudut2 float
SET @sudut2 = 52.64

DECLARE @cosinus float
SET @cosinus = COS(@sudut2)

SELECT
    @cosinus AS Cosinus,
    ACOS(@cosinus) AS Arccosinus
```



	Cosinus	Arccosinus
1	-0.719943714139416	2.37451754256331

Penjelasan :

Pertama, kita deklarasikan bahwa variabel @sudut2 sebagai float dan set nilainya berisi 52,64. Selanjutnya, kita deklarasikan variabel @cosinus, juga sebagai float, dan kemudian menggunakan fungsi COS untuk mengatur nilai @cosinus ke cosinus @sudut2. Akhirnya, kita membuat sebuah pernyataan SELECT yang mengambil nilai @cosinus dan arccosinus-nya. Perhatikan bahwa kita menggunakan fungsi ACOS untuk mengambil arccosinus tersebut. Arccosinus ini ditampilkan dalam radian, seperti ditunjukkan pada hasil diatas.

Hasil menunjukkan bahwa nilai cosinus untuk 52,64 : -0,719943714139416, dan arccosinus untuk nilai cosinus adalah 2,37451754256331.

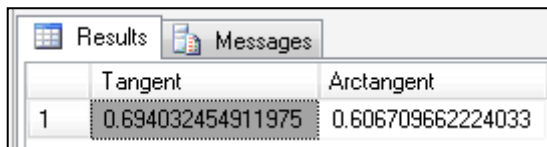
TAN dan ATAN

Dua fungsi lain yang serupa dengan set sebelumnya adalah TAN dan ATAN. Fungsi TAN mengembalikan tangen sebuah ekspresi float, dan fungsi ATAN mengembalikan sudut tangen dalam radian. Contoh berikut ini menunjukkan kedua fungsi tersebut :


```
DECLARE @nilai3 float
SET @nilai3 = 52.64

DECLARE @tangent float
SET @tangent = SIN(@nilai3)

SELECT
    @tangent AS Tangent,
    ATAN(@tangent) AS Arctangent
```



	Tangent	Arctangent
1	0.694032454911975	0.606709662224033

Penjelasan :

Seperti yang Anda lihat, contoh hampir identik dengan kedua fungsi sebelumnya, kecuali bahwa itu mengembalikan tangen dan arctangen.

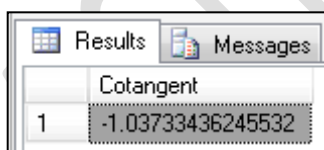
COT (Cotangent)

SQL Server juga mendukung fungsi COT, yang mengembalikan cotangens dari sudut tertentu, seperti ditunjukkan pada contoh berikut :

```
DECLARE @sudut4 float
SET @sudut4 = 52.64

DECLARE @cotangent float
SET @cotangent = COT(@sudut4)

SELECT @cotangent AS Cotangent
```



	Cotangent
1	-1.03733436245532

Penjelasan :

Dalam contoh ini, kita deklarasikan variabel @sudut4 dan set nilainya ke 52,64. Lalu kita deklarasikan variabel @cotangent dan set nilainya dengan menggunakan fungsi COT, untuk mengambil cotangent dari @sudut4. Selanjutnya, kita gunakan perintah SELECT untuk memanggil variabel @cotangent.

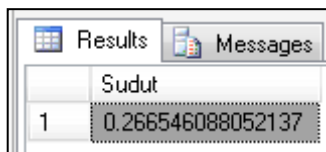
ATN2

Sekarang mari kita lihat fungsi ATN2. Fungsi mengembalikan sudut antara sumbu x positif dan garis yang berjalan dari asal ke titik tertentu yang diwakili oleh nilai x, y. Contoh berikut menunjukkan bagaimana ATN2 bekerja :

```
DECLARE @x float
SET @x = 52.64

DECLARE @y float
SET @y = 192.79

SELECT ATN2 (@x, @y) AS Sudut
```



	Sudut
1	0.266546088052137

Penjelasan :

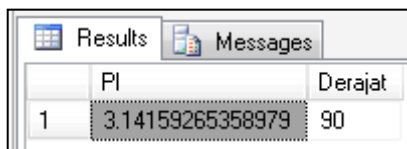
Perhatikan bahwa kita pertama kali mendeklarasikan dan menetapkan variabel @x dan @y dan menggunakan keduanya sebagai argumen dalam fungsi ATN2. Selanjutnya, kita menggunakan fungsi ATN2 dalam sebuah pernyataan SELECT untuk mengambil sudut.

PI

Satu lagi fungsi trigonometri yang ingin ditunjukkan yaitu PI, berikut contohnya :

```
DECLARE @pi float
SET @pi = PI()

SELECT
    @pi AS [PI],
    DEGREES (@pi/2) AS [Derajat]
```



	PI	Derajat
1	3.14159265358979	90

Pertama, kita menggunakan fungsi PI untuk mengatur variabel @pi dengan nilai pi, dan kemudian fungsi tersebut digunakan dalam sebuah pernyataan SELECT dimana kita mengambil nilai pi dan dikonversi @pi/2 ke derajat.

V. Fungsi Matematika

1. ROUND

Fungsi ROUND digunakan untuk membulatkan bilangan desimal menjadi satu nilai diatasnya ataupun nilai dibawahnya. Hal tersebut dapat dilakukan sesuai dengan kondisi nilai desimal yang diberikan. ROUND dapat berupa nilai positif atau negatif terhadap suatu panjang tertentu.

Sintaks :

```
SELECT ROUND (BilanganDesimal, JumlahDesimal)
```

BilanganDesimal : bilangan yang akan dibulatkan

JumlahDesimal : banyak desimal yang akan dihasilkan setelah bilangan aslinya (optional).

Contoh :

```
SELECT ROUND (5.49, 0) ,
        ROUND (5.50, 0) ,
        ROUND (5.51, 0) ,
        ROUND (5.59, 0)
```

Results		Messages		
	(No column name)	(No column name)	(No column name)	(No column name)
1	5.00	6.00	6.00	6.00

```
SELECT ROUND (5.49, 1) ,
        ROUND (5.50, 1) ,
        ROUND (5.51, 1) ,
        ROUND (5.59, 1)
```

Results		Messages		
	(No column name)	(No column name)	(No column name)	(No column name)
1	5.50	5.50	5.50	5.60

```
SELECT ROUND (5.490, 2) ,
        ROUND (5.550, 2) ,
        ROUND (5.551, 2) ,
        ROUND (5.559, 2)
```

Results		Messages		
	(No column name)	(No column name)	(No column name)	(No column name)
1	5.490	5.550	5.550	5.560

```
SELECT ROUND (-1.49, 1),
        ROUND (-1.50, 1),
        ROUND (-1.51, 1),
        ROUND (-1.59, 1)
```

	(No column name)	(No column name)	(No column name)	(No column name)
1	-1.50	-1.50	-1.50	-1.60

2. CEILING

Fungsi CEILING digunakan untuk membulatkan bilangan desimal menjadi satu nilai di atasnya.

```
SELECT CEILING (5.00),
        CEILING (5.01),
        CEILING (5.55),
        CEILING (5.99)
```

	(No column name)	(No column name)	(No column name)	(No column name)
1	5	6	6	6

```
SELECT CEILING (-5.00),
        CEILING (-5.01),
        CEILING (-5.55),
        CEILING (-4.99)
```

	(No column name)	(No column name)	(No column name)	(No column name)
1	-5	-5	-5	-4

3. FLOOR

Fungsi FLOOR merupakan kebalikan dari fungsi ceiling, digunakan untuk membulatkan bilangan desimal menjadi satu nilai dibawahnya.

```
SELECT FLOOR (4.00),
        FLOOR (4.99),
        FLOOR (5.01),
        FLOOR (5.50)
```

	(No column name)	(No column name)	(No column name)	(No column name)
1	4	4	5	5

4. ABS

Fungsi ABS digunakan untuk mendapatkan nilai absolut (nilai positif) dari suatu bilangan.

```
SELECT ABS (5) ,
        ABS (-5) ,
        ABS (-5.5)
```

Results Messages			
	(No column name)	(No column name)	(No column name)
1	5	5	5.5

5. POWER

Fungsi POWER digunakan untuk mencari nilai pangkat dari suatu bilangan.

```
SELECT POWER (2, 2) ,
        POWER (3, 2) ,
        POWER (4, 2)
```

Results Messages			
	(No column name)	(No column name)	(No column name)
1	4	9	16

6. RAND

Fungsi RAND digunakan untuk menghasilkan susunan angka yang bersifat acak (random).

Sintaks :

```
SELECT RAND ()
```

Contoh :

```
SELECT RAND (1) ,
        RAND (2) ,
        RAND (10) ,
        RAND (20)
```

Results Messages				
	(No column name)	(No column name)	(No column name)	(No column name)
1	0.713591993212924	0.713610626184182	0.713759689954247	0.713946019666829

7. SQRT

Fungsi SQRT digunakan untuk mencari akar kuadrat dari suatu bilangan.

Sintaks :

```
SELECT SQRT ( )
```

Contoh :

```
SELECT SQRT (9) ,  
       SQRT (16) ,  
       SQRT (25) ,  
       SQRT (36)
```

	(No column name)	(No column name)	(No column name)	(No column name)
1	3	4	5	6

8. SQUARE

Fungsi SQUARE digunakan untuk meng-kuadrat-kan dari suatu bilangan.

Sintaks :

```
SELECT SQUARE ( )
```

Contoh :

```
SELECT SQUARE (2) ,  
       SQUARE (3) ,  
       SQUARE (4) ,  
       SQUARE (5) )
```

	(No column name)	(No column name)	(No column name)	(No column name)
1	4	9	16	25

9. LOG

Fungsi LOG digunakan untuk mencari nilai logaritma dari suatu bilangan. LOG dapat digunakan berdasarkan basisnya, LOG10 untuk basis-10.

Sintaks :

```
SELECT LOG ( )
```

Contoh :

```
SELECT LOG (3.354) ,  
       LOG10 (100)
```

	(No column name)	(No column name)
1	1.21015366340102	0.525563058270067

10. EXP

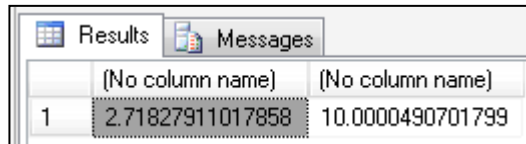
Fungsi eksponen (EXP) adalah kebalikan dari fungsi logaritma.

Sintaks :

```
SELECT EXP ( )
```

Contoh :

```
SELECT EXP (0.999999) ,  
EXP (2.30259)
```



	(No column name)	(No column name)
1	2.71827911017858	10.0000490701799

VI. Fungsi String

Pada pertemuan ini kita akan membahas fungsi-fungsi untuk mengolah data string (kata, kalimat, paragraf, dan teks).

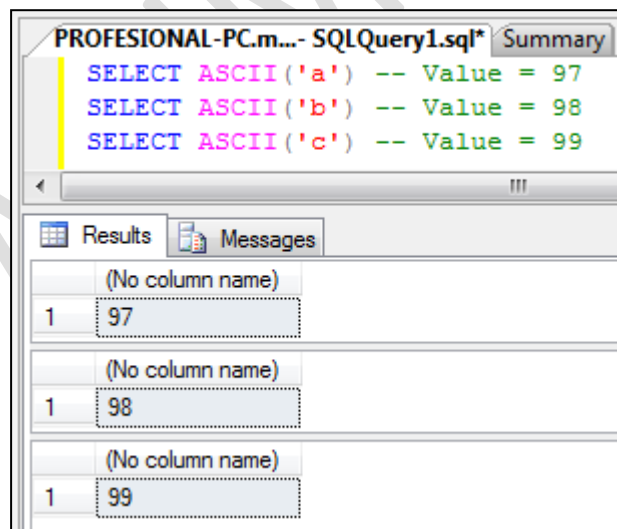
1. ASCII

Fungsi ASCII berguna untuk mengembalikan nilai kode ASCII dari karakter pada tombol keyboard.

Sintaks :

```
SELECT ASCII (Character)
```

Contoh :



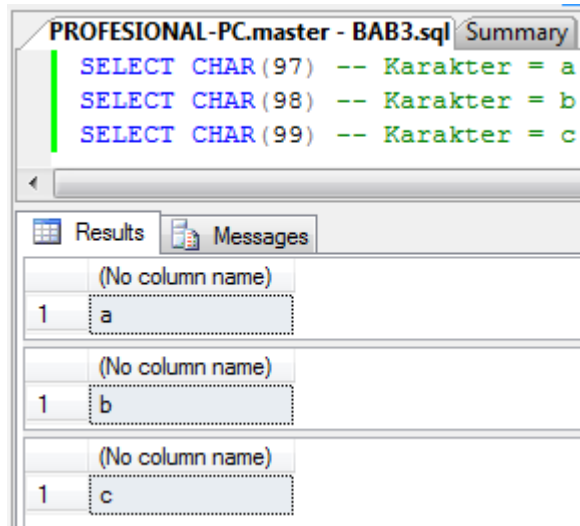
	(No column name)	(No column name)	(No column name)
1	97		
1	98		
1	99		

2. CHAR

Fungsi CHAR digunakan untuk mengambil karakter dari nilai kode ASCII pada tombol keyboard.

Sintaks :

```
SELECT CHAR (Kode ASCII)
```

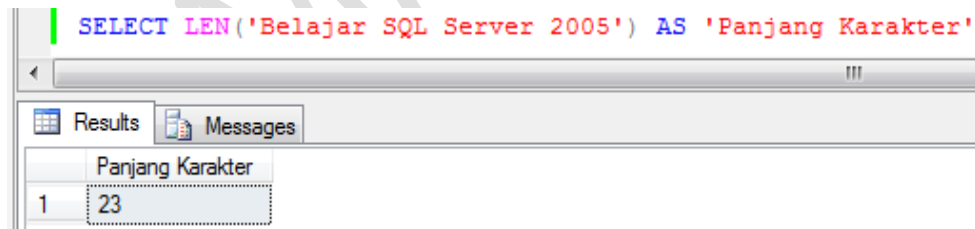


3. LEN

Fungsi LEN digunakan untuk menghitung panjang karakter

Sintaks :

```
SELECT LEN (String)
```



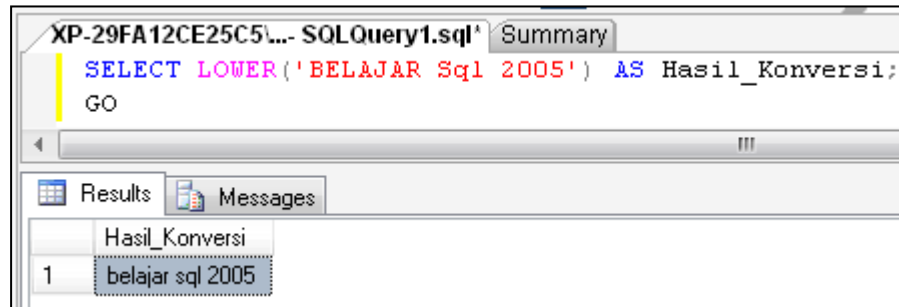
4. LOWER

Fungsi LOWER digunakan untuk mengkonversi dari huruf/string kedalam bentuk huruf kecil.

Sintaks :

```
SELECT LOWER (String)
```


Contoh :



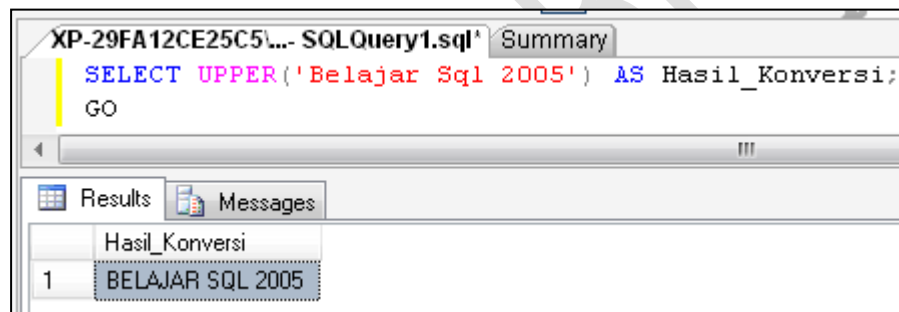
5. UPPER

Fungsi UPPER merupakan kebalikan dari fungsi LOWER, yaitu mengkonversi string menjadi huruf besar.

Sintaks :

```
SELECT UPPER(String)
```

Contoh :

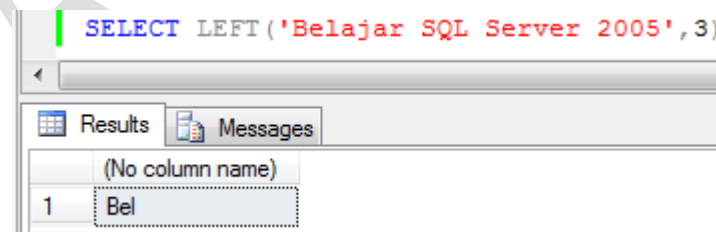


6. LEFT

Fungsi LEFT digunakan untuk mengambil dan menampilkan sejumlah karakter tertentu dari sebelah kiri suatu karakter string.

Sintaks :

```
SELECT LEFT(String, JumlahKarakter)
```

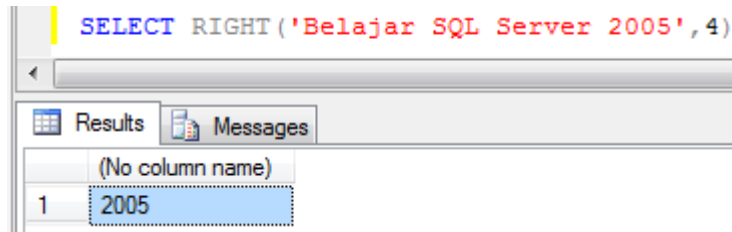


7. RIGHT

Fungsi RIGHT digunakan untuk mengambil dan menampilkan sejumlah karakter tertentu dari sebelah kanan suatu karakter string.

Sintaks :

```
SELECT RIGHT(String, JumlahKarakter)
```



8. SUBSTRING

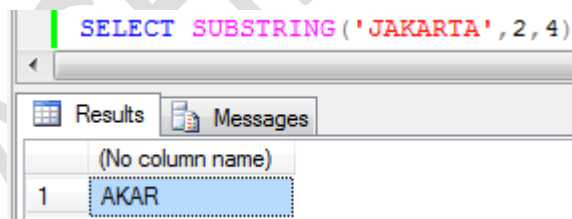
Fungsi SUBSTRING digunakan untuk mengambil sejumlah karakter tertentu dari tengah dari suatu karakter string, baik karakter yang ditengah maupun karakter yang kiri dan kanan.

```
SELECT SUBSTRING(String, NomorKarakterAwal, JumlahKarakter)
```

NomorKarakterAwal : urutan nomor pada susunan karakter string tersebut. Pada nomor pertama dianggap sebagai urutan 1, sedangkan pada urutan nomor terakhir dianggap sebagai nomor urut -1.

JumlahKarakter : jumlah karakter yang akan diambil dari suatu string.f

Misalkan dari kata 'JAKARTA', kita akan mengambil kata AKAR, maka perintahnya sebagai berikut :

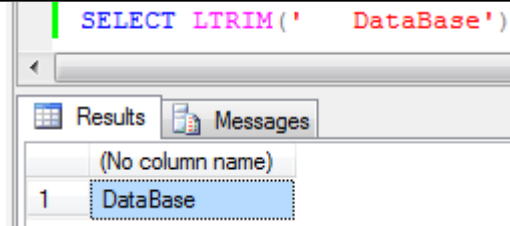


9. LTRIM

Fungsi LTRIM digunakan untuk menghilangkan karakter blank disebelah kiri string

Sintaks :

```
SELECT LTRIM(String)
```

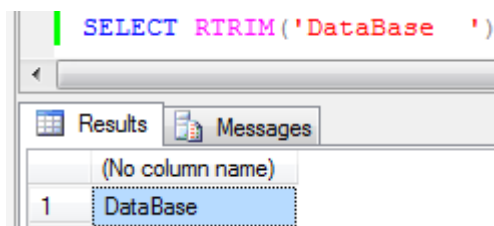


10. RTRIM

Fungsi RTRIM digunakan untuk menghilangkan karakter blank disebelah kanan string

Sintaks :

```
SELECT RTRIM(String)
```

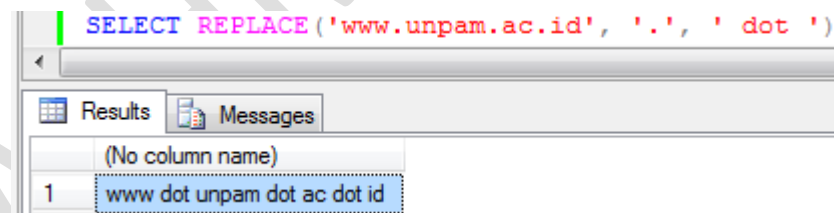


11. REPLACE

Fungsi REPLACE digunakan untuk menggantikan karakter lama dengan karakter baru pada suatu string.

Sintaks :

```
SELECT REPLACE(String, KarakterLama, KarakterBaru)
```

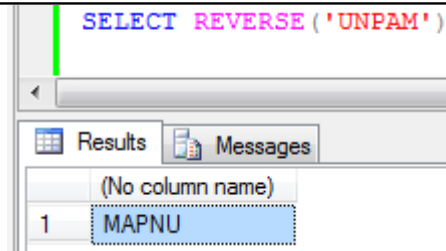


12. REVERSE

Fungsi REVERSE digunakan untuk membalik susunan karakter dari suatu string menjadi susunan yang terbalik.

Sintaks :

```
SELECT REVERSE(String)
```

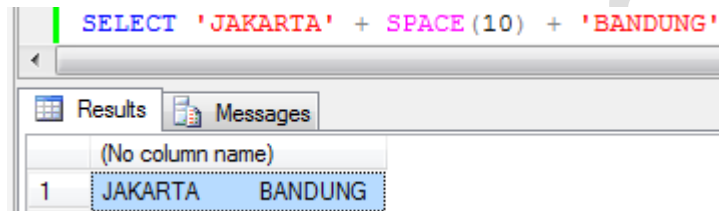


13. SPACE

Fungsi SPACE digunakan untuk membuat spasi kosong sebanyak jumlah spasi yang diinginkan.

Sintaks :

```
SELECT SPACE (JumlahSpasi)
```



VII. Fungsi Tanggal dan Waktu

Fungsi tanggal merupakan fungsi yang diperuntukan 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.

Komponen tanggal yang disebut **DatePart** digunakan untuk menentukan elemen nilai tanggal untuk aritmatika tanggal.

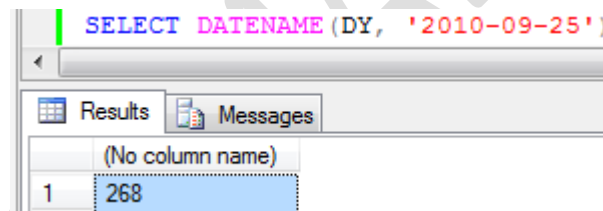
- **DATEADD**
Fungsi DATEADD digunakan untuk menambah datepart ke tanggal sesuai dengan besar number
Sintaks :
- **DATEDIFF**
Fungsi DATEDIFF digunakan untuk mengkalkulasi angka datepart antara dua tanggal
- **DATEPART**
Fungsi DATEPART digunakan untuk mengembalikan datepart dari daftar tanggal sebagai integer

- **DATENAME**
Fungsi DATENAME digunakan untuk mengembalikan datepart dari daftar tanggal sebagai nilai ASCII
- **GetDate**
Fungsi GetDate digunakan untuk mengambil tanggal dan waktu sekarang dari sistem komputer

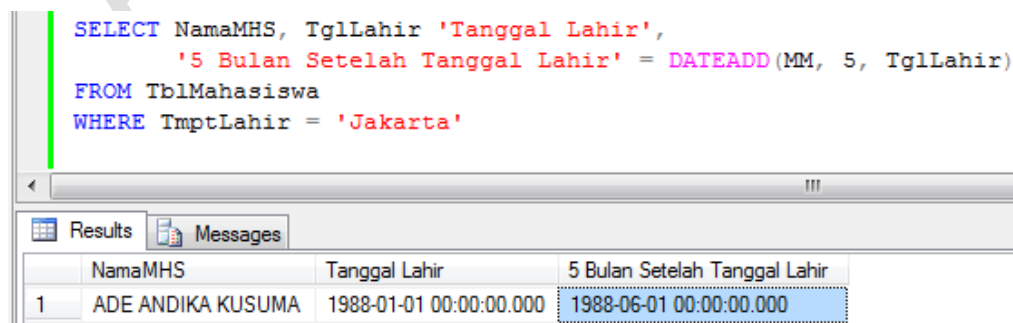
Tabel. Komponen Tanggal (DatePart)

DatePart	Singkatan	Nilai	Keterangan
Year	Tahun	YY	1753-9999
Quarter	Kwartel	QQ	1-4
Month	Bulan	MM	1-12
Day of Year	Hari ke	DY	1-366
Day	Tanggal	DD	1-31
Week	Minggu	WW	0-51
Weekday	Hari	DW	1-7 (1=Sunday)
Hour	Jam	HH	0-23
Minute	Menit	MI	(0-59)
Second	Detik	SS	(0-59)
Millisecond	Milidetik	MS	(0-999)

1. Menampilkan hari ke berapa pada urutan hari dalam setahun



2. Menampilkan nama, tanggal lahir, dan bertambah 5 bulan setelah tanggal lahir yang tempat lahirnya Jakarta, dengan menggunakan fungsi DATEADD.



- Menampilkan nama, tanggal lahir, dan bertambah 30 hari setelah tanggal lahir, dengan menggunakan fungsi DATEADD.

```
SELECT NamaMHS, TglLahir 'Tanggal Lahir',
       '30 Hari Setelah Tanggal Lahir' = DATEADD(DD, 30, TglLahir)
FROM TblMahasiswa
```

	NamaMHS	Tanggal Lahir	30 Hari Setelah Tanggal Lahir
1	ADE CAHYADI	1987-08-17 00:00:00.000	1987-09-16 00:00:00.000
2	ADE ANDIKA KUSUMA	1988-01-01 00:00:00.000	1988-01-31 00:00:00.000
3	ARIZONA RH	1985-10-02 00:00:00.000	1985-11-01 00:00:00.000

- Menampilkan hari dari sebuah tanggal menggunakan fungsi DATENAME.

```
SELECT DATENAME(DW, '2010-09-25')
```

	(No column name)
1	Saturday

- Menampilkan bulan dari sebuah tanggal menggunakan fungsi DATENAME.

```
SELECT DATENAME(MM, '2010-09-25')
```

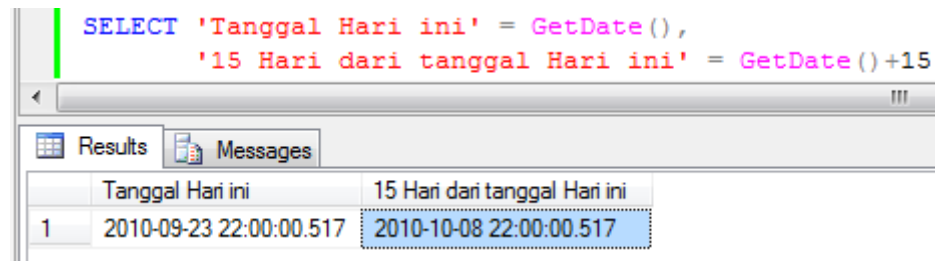
	(No column name)
1	September

- Dari tabel mahasiswa ingin mengetahui hari apa mahasiswa lahir, dengan menggunakan fungsi DATENAME.

```
SELECT NamaMHS, TglLahir 'Tanggal Lahir',
       'Hari Lahir' = DATENAME(DW, TglLahir)
FROM TblMahasiswa
```

	NamaMHS	Tanggal Lahir	Hari Lahir
1	ADE CAHYADI	1987-08-17 00:00:00.000	Monday
2	ADE ANDIKA KUSUMA	1988-01-01 00:00:00.000	Friday
3	ARIZONA RH	1985-10-02 00:00:00.000	Wednesday

7. Selanjutnya ingin mengetahui tanggal pada hari ini dan 15 hari setelah hari ini, dengan menggunakan fungsi GetDate.



The screenshot shows a SQL query window with the following text:

```
SELECT 'Tanggal Hari ini' = GetDate(),  
       '15 Hari dari tanggal Hari ini' = GetDate()+15
```

Below the query window, the 'Results' tab is active, displaying a table with two columns: 'Tanggal Hari ini' and '15 Hari dari tanggal Hari ini'. The first row shows the current date and time (2010-09-23 22:00:00.517) and the date 15 days later (2010-10-08 22:00:00.517).

	Tanggal Hari ini	15 Hari dari tanggal Hari ini
1	2010-09-23 22:00:00.517	2010-10-08 22:00:00.517