

## PEMROGRAMAN BASIS DATA

### Function

**Kamarudin, M.Kom**

[kamarudin@amikom.ac.id](mailto:kamarudin@amikom.ac.id)

<http://coding4ever.net/>

<https://github.com/rudi-krsoftware/open-retail>

# Materi

- ✓ Definisi Function
- ✓ Manfaat Function
- ✓ Jenis-jenis Function
  1. Build-in Function
  2. User Defined Function (UDF)
    - Scalar Function
    - Table Value Function
      - Inline Table-valued Function
      - Multi-statement Table-valued Function

# Function

Berfungsi seperti store procedure dengan sedikit perbedaan yaitu store procedure *tidak mengembalikan nilai* sedang function *bisa mengembalikan sebuah nilai tunggal (scalar) atau tabel*.

# Jenis-jenis Function

- ✓ Built in function.  
Fungsi yang sudah disediakan oleh DBMS. Contoh :
  1. Fungsi agregasi : SUM, MIN, MAX
  2. Fungsi string : LEFT, RIGHT, SUBSTRING
- ✓ User defined function (UDF).  
Fungsi yang dibuat oleh user

# User Defined Function

Seperti halnya method/function dalam bahasa pemrograman, user defined function berisi sekumpulan perintah (T-SQL) yang menerima parameter, melakukan aksi tertentu seperti perhitungan sederhana/komplek dan mengembalikan hasilnya.

User defined function mempunyai beberapa keuntungan:

- ✓ Penggunaan ulang kode yang sama
- ✓ Mengurangi lalu lintas jaringan.
- ✓ Eksekusi program menjadi lebih cepat.
- ✓ Lebih mudah dimaintenance
- ✓ Dapat digunakan untuk mekanisme keamanan.

# User Defined Function (Lanjutan)

Terbagi menjadi dua yaitu :

1. Scalar function

Mengembalikan nilai dengan type data scalar (tipe standar SQL Server seperti Varchar, char, int dan lain-lain).

2. Table value function

Jika scalar function hanya mengembalikan sebuah nilai dengan tipe data tertentu, maka table value function dapat mengembalikan nilai berupa table.

Ada dua jenis yaitu :

- ✓ Inline Table-valued Functions
- ✓ Multi-statement Table-valued Functions

# Scalar Function

Perintah yang digunakan :

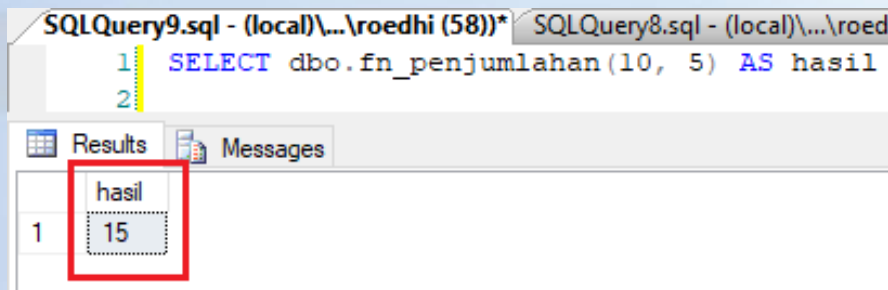
```
CREATE FUNCTION nama_func (  
    @parameter_1 TipeData_1, ...  
)  
RETURNS scalar_data_type  
AS  
BEGIN  
    perintah T-SQL  
  
    RETURN scalar_expression  
END
```

# Contoh #1

Buat sebuah fungsi untuk menjumlahkan 2 buah bilangan.

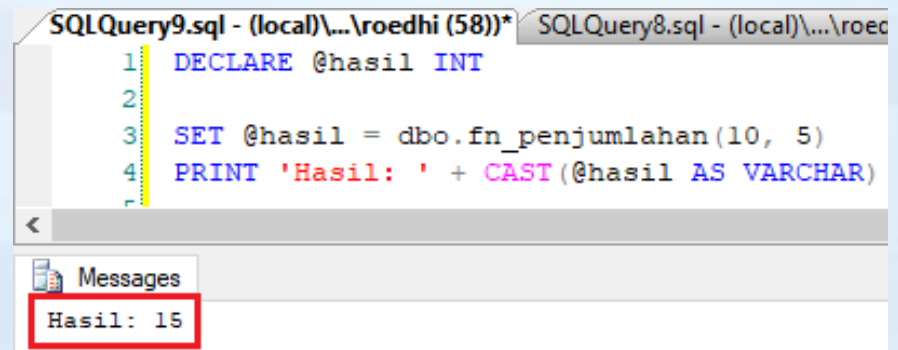
```
1 CREATE FUNCTION fn_penjumlahan (  
2     @bilangan1 INT, @bilangan2 INT  
3 )  
4 RETURNS INT  
5 AS  
6 BEGIN  
7     DECLARE @hasil INT  
8  
9     SET @hasil = @bilangan1 + @bilangan2  
10  
11     RETURN @hasil  
12 END
```

Contoh pemanggilan :



The screenshot shows a SQL Server Enterprise Manager window with a query executed. The query is `SELECT dbo.fn_penjumlahan(10, 5) AS hasil`. The Results tab is active, displaying a single row with the value 15 under the column header 'hasil'.

	hasil
1	15



The screenshot shows a SQL Server Enterprise Manager window with a query executed. The query is `DECLARE @hasil INT; SET @hasil = dbo.fn_penjumlahan(10, 5); PRINT 'Hasil: ' + CAST(@hasil AS VARCHAR)`. The Messages tab is active, displaying the output 'Hasil: 15'.

```
1 DECLARE @hasil INT  
2  
3 SET @hasil = dbo.fn_penjumlahan(10, 5)  
4 PRINT 'Hasil: ' + CAST(@hasil AS VARCHAR)
```

Hasil: 15



# Contoh #2

Buat sebuah fungsi untuk mendapatkan bobot nilai (angka) dari nilai huruf.

```
1 CREATE FUNCTION fn_get_bobot_nilai (  
2     @nilai_huruf CHAR(1)  
3 )  
4 RETURNS INT  
5 AS  
6 BEGIN  
7     DECLARE @nilai_angka INT  
8  
9     IF @nilai_huruf = 'A'  
10        SET @nilai_angka = 4  
11     ELSE IF @nilai_huruf = 'B'  
12        SET @nilai_angka = 3  
13     ELSE IF @nilai_huruf = 'C'  
14        SET @nilai_angka = 2  
15     ELSE IF @nilai_huruf = 'D'  
16        SET @nilai_angka = 1  
17     ELSE  
18        SET @nilai_angka = 0  
19  
20     RETURN @nilai_angka  
21 END
```



Contoh pemanggilan :

```
SQLQuery2.sql - (local)\...roedhi (52))* SQLQuery1.sql - (local)\...roedhi (59))*  
1 DECLARE @nilai_huruf CHAR(1)  
2 DECLARE @nilai_angka INT  
3  
4 SET @nilai_huruf = 'A'  
5 SET @nilai_angka = dbo.fn_get_bobot_nilai(@nilai_huruf)  
6  
7 PRINT 'Bobot nilai ' + @nilai_huruf + ' = ' +  
8     CAST(@nilai_angka AS VARCHAR)  
9  
<  
Messages  
Bobot nilai A = 4
```

# Contoh #3

Buat sebuah fungsi untuk mendapatkan nama hari berdasarkan tanggal tertentu.

```
1 CREATE FUNCTION fn_get_nama_hari (  
2     @tanggal DATETIME  
3 )  
4 RETURNS VARCHAR(7)  
5 AS  
6 BEGIN  
7     DECLARE @hari VARCHAR(7)  
8     DECLARE @nomor_hari INT  
9  
10    SET @nomor_hari = DATEPART(WEEKDAY, @tanggal)  
11  
12    IF @nomor_hari = 1  
13        SET @hari = 'Minggu'  
14    ELSE IF @nomor_hari = 2  
15        SET @hari = 'Senin'  
16    ELSE IF @nomor_hari = 3  
17        SET @hari = 'Selasa'  
18    ELSE IF @nomor_hari = 4  
19        SET @hari = 'Rabu'  
20    ELSE IF @nomor_hari = 5  
21        SET @hari = 'Kamis'  
22    ELSE IF @nomor_hari = 6  
23        SET @hari = 'Jumat'  
24    ELSE  
25        SET @hari = 'Sabtu'  
26  
27    RETURN @hari  
28 END
```



Contoh pemanggilan :

```
SQLQuery2.sql - (local)\...\\roedhi (52))* SQLQuery1.sql - (local)\...\\roedhi  
1 DECLARE @tanggal DATETIME  
2 DECLARE @hari VARCHAR(7)  
3  
4 SET @tanggal = '2019-06-25'  
5 SET @hari = dbo.fn_get_nama_hari(@tanggal)  
6  
7 PRINT 'Hari: ' + CAST(@hari AS VARCHAR)
```

Messages  
Hari: Selasa

# Contoh #4

Buat sebuah fungsi untuk mendapatkan informasi stok product berdasarkan product id.

Products	
product_id	
name	
stok	
purchasing_price	
selling_price	
category_id	
supplier_id	



```
1 CREATE FUNCTION fn_get_stok_product (  
2     @product_id CHAR(15)  
3 )  
4 RETURNS INT  
5 AS  
6 BEGIN  
7     DECLARE @stok INT  
8  
9     SELECT @stok = stok FROM products  
10    WHERE product_id = @product_id  
11  
12    RETURN @stok  
13 END
```

Contoh pemanggilan :

```
SQLQuery9.sql - (local)\... \roedhi (58)* SQLQuery8.sql - (local)\... \roedhi (5  
1 DECLARE @stok INT  
2  
3 SET @stok = dbo.fn_get_stok_product('12345')  
4 PRINT 'Stok: ' + CAST(@stok AS VARCHAR)  
5  
6  
7  
8  
9  
10  
11  
12  
13
```

Messages

Stok: 35

# Contoh #5

Fungsi juga bisa dipanggil sebagai *column expression*.

SQLQuery11.sql - ...DJ7G\roedhi (60))\* SQLQuery10.sql - ...DJ7G\roedhi (58)) SQLC

```
1 select npm, nama, (dbo.fn_get_ipk(mhs.npm)) AS ipk
2 from mhs
3 where npm like '14.11.%'
4
```

Results Messages

	npm	nama	ipk
1	14.11.7616	KRISNAMURTIAJI	3.55072463768116
2	14.11.7617	FAUZAN AHAD NURFALAH	3.25
3	14.11.7618	MARTINUS YUDI PURWONO	3.44444444444444
4	14.11.7619	INDRA SETIAWAN	3.81944444444444
5	14.11.7620	YUSUP DARMAN JATI	3.07246376811594
6	14.11.7621	DWIKI FATAN AZIZI	3.43055555555556
7	14.11.7622	ERVIEEN ISMUNANDAR SUSILA	3.38888888888889
8	14.11.7624	DICKY HARDIKA DIMAS NUGROHO	3.5
9	14.11.7625	RENKARISMADYA	3.68055555555556
10	14.11.7626	RADEN RISKY SAPUTRO	2.92753623188406
11	14.11.7627	RINDA AZARICA NASFADIASY	2.78260869565217
12	14.11.7628	HARRY VISWANTO	3.38888888888889
13	14.11.7630	MUHAMMAD MAHFUD ASHARI	3.375
14	14.11.7631	HFI MI FAKHRIYANTO	3.47826086956522

# Inline Table-Valued Functions

Inline Table-valued Functions hanya bisa menampung satu pernyataan SELECT.

Perintah yang digunakan :

```
CREATE FUNCTION nama_func (  
    @parameter_1 TipeData_1, ...  
)  
RETURNS TABLE  
AS  
    RETURN (perintah-select)
```

# Contoh #1

```
1 CREATE FUNCTION fn_get_supplier (  
2     @supplier_id CHAR(10)  
3 )  
4 RETURNS TABLE  
5 AS  
6     RETURN (select supplier_id, name, address  
7             from suppliers where supplier_id = @supplier_id)  
8
```

Contoh pemanggilan :

SQLQuery9.sql - (local)\...\roedhi (54))\* SQLQuery8.sql - (local)\...\roedhi (52))\*

```
1 select * from dbo.fn_get_supplier('SUP-000001')  
2
```

<

Results Messages

	supplier_id	name	address
1	SUP-000001	G&R Collection	Semarang

# Contoh #2

```
1 CREATE FUNCTION fn_get_product (  
2     @category_id CHAR(6)  
3 )  
4 RETURNS TABLE  
5 AS  
6 RETURN (select categories.category_id, categories.description as category,  
7     products.product_id, products.name as product, products.stok,  
8     products.purchasing_price, products.selling_price  
9     from categories inner join products on categories.category_id = products.category_id  
10    where categories.category_id = @category_id)  
11
```

Contoh pemanggilan :

SQLQuery9.sql - (local)\...roedhi (54))\* SQLQuery8.sql - (local)\...roedhi (52))\*

```
1 select * from dbo.fn_get_product('CG-001')  
2
```

	category_id	category	product_id	product	stok	purchasing_price	selling_price
1	CG-001	Fashions & Accessories	CG-001-IT000001	Snapback Hat NY	50	135000.00	140000.00
2	CG-001	Fashions & Accessories	CG-001-IT000002	Roberto Cavalli Sunglass	100	160000.00	165000.00
3	CG-001	Fashions & Accessories	CG-001-IT000003	Porsche Design Sunglass	25	265000.00	275000.00
4	CG-001	Fashions & Accessories	CG-001-IT000004	Giorgio Armani Wallet	50	145000.00	150000.00
5	CG-001	Fashions & Accessories	CG-001-IT000005	Levi's Design Wallet	50	160000.00	175000.00

# Multi-statement Table-valued Functions

Multi-statement Table-valued Functions bisa menampung lebih dari satu pernyataan SELECT (query). Fungsi ini sangat berguna jika ingin menjalankan banyak query, dan hasilnya di kembalikan dalam bentuk tabel.

Perintah yang digunakan :

```
CREATE FUNCTION nama_func (  
    @parameter_1 TipeData_1, ...  
)  
RETURNS @return_variable TABLE (table_type_definition)  
AS  
BEGIN  
  
    perintah T-SQL  
  
    RETURN  
END
```



# Contoh

```
1 CREATE FUNCTION fn_get_contacts()  
2 RETURNS  
3     @contacts TABLE (  
4         name VARCHAR(30),  
5         address VARCHAR(50),  
6         contact_type VARCHAR(20)  
7     )  
8 AS  
9 BEGIN  
10     INSERT INTO @contacts  
11     SELECT name, address, 'supplier' FROM suppliers  
12  
13     INSERT INTO @contacts  
14     SELECT name, address, 'customer' FROM customers  
15  
16     RETURN  
17 END
```

Contoh pemanggilan :

SQLQuery9.sql - (local)\...roedhi (54))\* SQLQuery8.sql - (local)

```
1 SELECT * FROM dbo.fn_get_contacts()  
2
```

	name	address	contact_type
1	G&R Collection	Semarang	supplier
2	Mollersz	Jakarta	supplier
3	Tamza Accessories	Surabaya	supplier
4	Bagus Nur Hakim	Jakarta	customer
5	Putri Kharisma	Yogyakarta	customer
6	Aliany Milawaty	Yogyakarta	customer
7	Yudha Adisaputra	Solo	customer