

## Cara Membuat Function di SQL

Dalam pengembangan database dan pemrograman SQL, seringkali kita perlu membuat dan menggunakan function. Function adalah objek SQL yang digunakan untuk melakukan tugas-tugas tertentu dan mengembalikan hasil yang diinginkan. Dengan menggunakan function, kita dapat mengelompokkan logika dan operasi-operasi yang sering digunakan menjadi satu unit yang dapat dipanggil dari dalam pernyataan SQL lainnya.

Artikel ini akan membahas secara rinci tentang cara membuat function di SQL dan memberikan contoh-contoh yang berguna. Kami akan meliputi sintaks dasar, tipe-tipe function yang tersedia, dan langkah-langkah untuk membuat dan menggunakan function di SQL.

### Mengenal Sintaks Dasar Function di SQL

Sebelum kita membahas cara membuat function di SQL, penting untuk memahami sintaks dasar function di SQL. Berikut adalah sintaks dasar untuk membuat function di SQL:

```
CREATE FUNCTION nama_function ([parameter1 tipe_data1, parameter2
tipe_data2, ...])
    RETURNS tipe_data
    [LANGUAGE {SQL | bahasa}]
    [DETERMINISTIC | NOT DETERMINISTIC]
    [SQL DATA ACCESS {CONTAINS SQL | NO SQL | READS SQL DATA | MODIFIES
SQL DATA}]
    [COMMENT 'komentar']
    BEGIN
        -- Blok kode SQL di sini
    END;
```

Mari kita jelajahi setiap bagian dari sintaks di atas:

- **nama\_function**: Nama function yang ingin Anda buat.
- **[parameter1 tipe\_data1, parameter2 tipe\_data2, ...]**: Parameter opsional yang dapat diteruskan ke function.
- **RETURNS tipe\_data**: Jenis data yang akan dikembalikan oleh function.

- `[LANGUAGE {SQL | bahasa}]`: Bahasa yang digunakan dalam function. Biasanya SQL.
- `[DETERMINISTIC | NOT DETERMINISTIC]`: Opsi untuk menunjukkan apakah function deterministik atau tidak. Jika deterministik, maka function akan mengembalikan hasil yang sama untuk input yang sama.
- `[SQL DATA ACCESS {CONTAINS SQL | NO SQL | READS SQL DATA | MODIFIES SQL DATA}]`: Menentukan akses data SQL yang digunakan oleh function.
- `[COMMENT 'komentar']`: Opsi untuk memberikan komentar deskriptif tentang function.
- `BEGIN` dan `END`: Menandai awal dan akhir blok kode SQL dalam function.

Sekarang setelah kita mengenal sintaks dasar function di SQL, mari kita jelajahi langkah-langkah untuk membuat function di SQL.

### Langkah-langkah Membuat Function di SQL

Berikut adalah langkah-langkah untuk membuat function di SQL:

#### Langkah 1: Menentukan Nama dan Parameter Function

Pertama, kita perlu menentukan nama function yang ingin kita buat. Pastikan nama function tersebut deskriptif dan menggambarkan tugas atau hasil yang akan diberikan. Selanjutnya, kita dapat menentukan parameter-function yang diterima oleh function tersebut. Parameter-function adalah nilai yang dapat diteruskan ke function saat pemanggilan.

#### Langkah 2: Menentukan Jenis Data yang Dikembalikan

Setelah menentukan nama dan parameter-function, kita perlu menentukan jenis data yang akan dikembalikan oleh function. Misalnya, jika function akan menghitung total penjualan, jenis data yang dikembalikan mungkin berupa angka desimal.

### Langkah 3: Menentukan Logika dan Operasi Function

Berikutnya, kita perlu menentukan logika dan operasi yang ingin dilakukan oleh function. Ini bisa berupa serangkaian pernyataan SQL yang menjalankan operasi yang diinginkan dan mengembalikan hasil yang diharapkan.

Pastikan untuk memperhatikan sintaks SQL yang benar dan memeriksa kecocokan tipe data dalam operasi yang dilakukan.

### Langkah 4: Membuat Function dengan Sintaks CREATE FUNCTION

Setelah kita menentukan semua parameter, jenis data yang dikembalikan, dan logika function, saatnya membuat function dengan menggunakan sintaks **CREATE FUNCTION**. Gunakan sintaks yang sudah dijelaskan sebelumnya dan masukkan logika dan operasi yang telah ditentukan ke dalam blok kode SQL function.

### Langkah 5: Menggunakan Function di SQL

Setelah function dibuat, kita dapat menggunakannya dalam pernyataan SQL lainnya. Cara menggunakan function sama seperti menggunakan fungsi built-in SQL lainnya. Kita dapat memanggil function dengan memberikan argumen yang sesuai dan menggunakan hasil yang dikembalikan sesuai kebutuhan.

### Contoh Penggunaan Function di SQL

Mari kita lihat contoh penggunaan function di SQL dengan membuat function sederhana yang mengembalikan total harga berdasarkan harga satuan dan jumlah item.

```
CREATE FUNCTION hitung_total_harga (harga_satuan DECIMAL(10, 2),  
jumlah_item INT)  
RETURNS DECIMAL(10, 2)  
BEGIN  
    DECLARE total DECIMAL(10, 2);  
    SET total = harga_satuan * jumlah_item;  
    RETURN total;  
END;
```

Setelah function di atas dibuat, kita dapat menggunakannya dalam pernyataan SQL lainnya seperti ini:

```
SELECT hitung_total_harga(10.50, 5) AS total;
```

Hasilnya akan menjadi:

```
total  
-----  
52.50
```

Dalam contoh di atas, function `hitung_total_harga` menerima dua argumen: `harga_satuan` dan `jumlah_item`. Function tersebut mengalikan kedua argumen tersebut dan mengembalikan hasilnya sebagai total harga.

### Kesimpulan

Function adalah objek SQL yang berguna untuk mengelompokkan logika dan operasi-operasi yang sering digunakan menjadi satu unit yang dapat dipanggil dari dalam pernyataan SQL lainnya. Dengan menggunakan function, kita dapat meningkatkan kinerja dan efisiensi database kita.

Selanjutnya kita akan belajar mengenai aturan penulisan function tersebut beserta contoh sederhannya. Untuk itu akan dibuat 2 point, yaitu a dan b seperti di bawah ini:

#### **a. Aturan penulisan function mysql**

Aturan penulisan syntax function dapat dilihat pada block tulisan di bawah ini:

```
DELIMITER $$
USE [nama database] $$
DROP FUNCTION IF EXISTS [nama function]
CREATE FUNCTION [nama function]([parameter]) RETURNS [tipe data
return] CHARSET latin1
    DETERMINISTIC
BEGIN
    [bagian block logic function]
    RETURN [varibel sebagai output function]
END$$
DELIMITER ;
```

Di atas merupakan aturan penulisan function dengan kata yang dikurungi [] merupakan inputan yang harus diisi dari programmer. Penjelasannya adalah sebagai berikut:

- Use [nama database] = perintah function ini dibuat didatabase dengan nama apa, contoh use coba\_pertama berarti kita memakai database coba\_pertama untuk membuat function.
- Create function [nama function] ( [parameter] ) returns [output tipe data] = perintah penulisan bagian head (kepala) yang berisi nama function , parameter berupa nama variabel beserta tipe datanya, sedangkan returns [output tipe data] merupakan tipe data output dari function itu (function pasti menghasilkan sebuah output, jika tidak ada return akan error tidak bisa diexecute).

- [bagian block logic function] = bagian kita menaruh perintah-perintah seperti declare variabel, operasi aritmatika, dll (nanti kita bisa lihat contohnya).
- RETURN [variabel sebagai output] = variabel sebagai hasil akhir dari sebuah function mysql

Penjelasan di atas tentunya belum afdol jika tidak disertai contohnya. Oh ya, tipe data di mysql terdiri dari banyak tipe data, yaitu: Int, bigint, smallint, mediumint, tinyint, float, double, decimal, numeric, date, varchar, char, dll. Itu semua dapat kamu cari penjelasannya di mbah google karena kalau saya jelaskan tiap tipe akan melebar kemana-mana. Biasanya saya memakai decimal dan varchar sebagai presentasi output hitungan uang dan text.

#### **b. Contoh pembuatan sebuah function**

Untuk lebih memahami penulisan function di mysql, admin akan memberi satu contoh dengan skenario daftar nilai siswa dan pengkategorian kelulusan.

Pertama-tama tabel yang kita gunakan bernama daftarnilai dan berada di database coba\_pertama. Daftar nilai memiliki data sebagai berikut:

	nama	kelas	nilai
<input type="checkbox"/>	budi	ipa 1	90
<input type="checkbox"/>	david	ipa 1	78
<input type="checkbox"/>	dodi	ipa 1	56
<input type="checkbox"/>	joko	ipa 1	85

kita ingin membuat pengkategorian yang mana lulus dan yang mana tidak. maka dari itu berikut adalah perintah pembuatan function dan cara pemanggilan function tersebut dalam query, dengan kode function berikut:

```
DELIMITER $$

USE `coba_pertama`$$

DROP FUNCTION IF EXISTS `kategori_lulus`$$
```

```

CREATE FUNCTION `kategori_lulus`(xnilai INT) RETURNS VARCHAR(155) CHARSET
latin1
  DETERMINISTIC
BEGIN
  DECLARE klulus VARCHAR(155);
  IF xnilai >=60 THEN
    SET klulus:='Lulus';
  ELSE
    SET klulus:='Tidak lulus';
  END IF;

  RETURN klulus;
END$$

DELIMITER ;

```

Pada kode di atas admin membuat function dengan nama kategori\_lulus dengan parameter inputan variabel xnilai yang didapat waktu kita memanggil function tersebut. Untuk output function kita memakai variabel klulus dengan tipe data varchar panjang 155 kita sesuaikan dengan header output function, yaitu varchar (155). Selanjutnya, kita akan menulis logika dimana jika memiliki nilai lebih sama dengan 60 maka dia lulus dan jika di bawah itu akan dilabeli tidak lulus, hal ini dipresentasikan dengan perintah if xnilai >= 60 else ... end if dan label tersebut disimpan divariabel klulus yang sudah kita declare di atas.

Untuk proses pemanggilan function kategori\_lulus tadi dapat kita simak dengan kode di bawah ini:

```

SELECT a.*,kategori_lulus(a.nilai) AS kategori FROM daftarnilai a

```

dan hasil output akan seperti ini:

	nama	kelas	nilai	kategori
<input type="checkbox"/>	budi	ipa 1	90	Lulus
<input type="checkbox"/>	david	ipa 1	78	Lulus
<input type="checkbox"/>	dodi	ipa 1	56	Tidak lulus
<input type="checkbox"/>	joko	ipa 1	85	Lulus

Melalui query `select a.*, kategori_lulus(a.nilai) ...` di atas pemanggilan function berada dibagian kolom select sehingga dapat menghasilkan output label lulus/tidak lulus.