A close-up photograph of a person's hands interacting with a smartphone. One hand holds the phone steady while the other points its index finger towards the screen. The background is dark and out of focus.

CHAPTER 22



Menggunakan Function Dalam SQL Statements



Function adalah suatu sub program di dalam MySQL yang diperuntukkan untuk mengerjakan tugas tertentu.



INTRODUKSI



Pengertian Fungsi (function)

Dalam mengerjakan suatu perintah tertentu di dalam MySQL terdapat suatu sub program yang disebut function atau method yang diperuntukkan untuk mengerjakan tugas tertentu, seperti melakukan penghitungan nilai rata-rata, nilai minimal atau melakukan penjumlahan atas suatu nilai. Fungsi bisa memudahkan pengguna dalam melakukan pengolahan dan analisis data. Selain itu menguasai function juga menjadi dasar dalam pengembangan aplikasi situs web dengan memakai database MySQL.

Function di MySQL

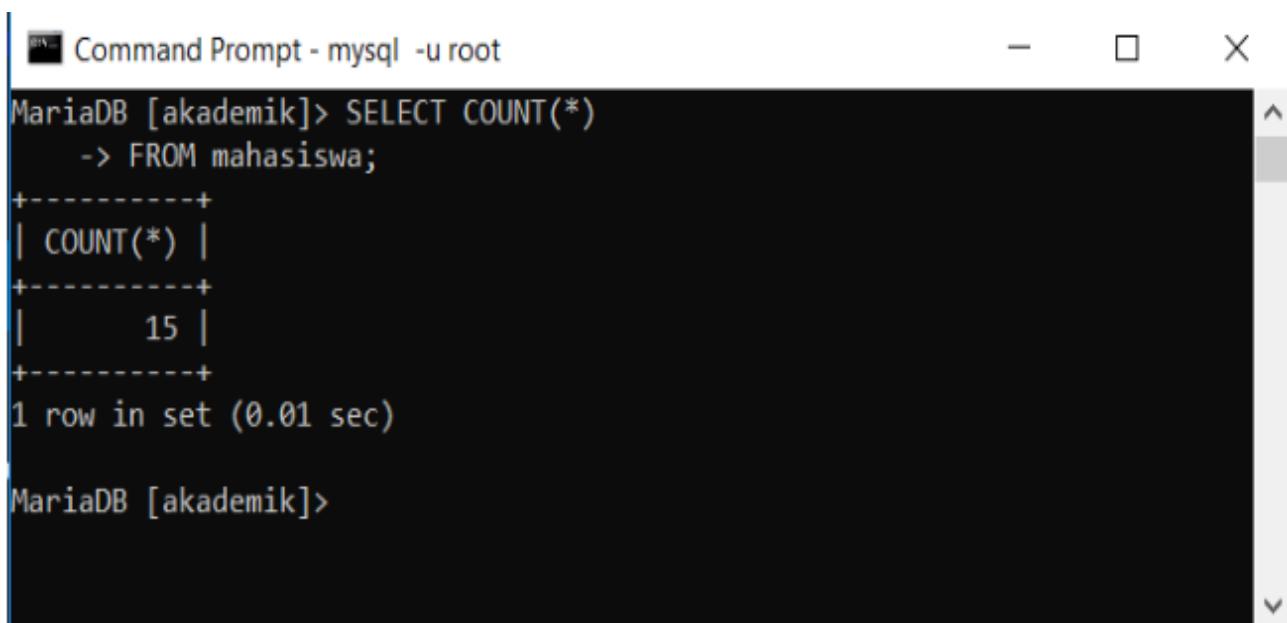
Berikut ialah penjelasan mengenai sejumlah fungsi di MySQL yang relatif sering digunakan:

Fungsi Count()

Berfungsi untuk menghitung jumlah seluruh baris query pada suatu tabel atau cuma pada baris yang memenuhi kondisi tertentu. Integer ialah return value (nilai balik) dari fungsi count.

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

Sintak di atas melakukan penghitungan jumlah baris pada tabel mahasiswa



```
Command Prompt - mysql -u root  
MariaDB [akademik]> SELECT COUNT(*)  
    -> FROM mahasiswa;  
+-----+  
| COUNT(*) |  
+-----+  
|      15 |  
+-----+  
1 row in set (0.01 sec)  
  
MariaDB [akademik]>
```



Fungsi avg()

Ialah fungsi untuk menghitung nilai rata-rata pada suatu kolom numerik. Misal jika terdapat bilangan 7; 6; 4; 2; 8; 5 maka dari ke-6 bilangan tersebut memiliki nilai rata-rata 5,3. Lihat ilustrasi di bawah, yang berisi daftar umur mahasiswa.

```
Command Prompt - mysql -u root
MariaDB [akademik]> SELECT umur
    -> FROM mahasiswa;
+-----+
| umur |
+-----+
| 20   |
| 22   |
| 22   |
| 23   |
| 23   |
| 20   |
| 23   |
| 23   |
| 22   |
| 19   |
| 20   |
| 25   |
| 21   |
| 23   |
| 24   |
+-----+
15 rows in set (0.00 sec)
```

Maka, nilai rata-rata bisa diperoleh dengan Sintak:

```
SELECT avg(umur)
FROM mahasiswa;
```

Perintah kueri diatas menghasilkan output 22. Artinya usia mahasiswa mempunyai nilai rata-rata 22 tahun.



```
Command Prompt - mysql -u root
MariaDB [akademik]> SELECT avg(umur)
    -> FROM mahasiswa;
+-----+
| avg(umur) |
+-----+
| 22.0000 |
+-----+
1 row in set (0.01 sec)
```

Fungsi Sum()

Dipakai untuk melakukan penjumlahan nilai pada suatu field. Perbedaan dengan fungsi count() ialah fungsi sum() digunakan untuk melakukan penjumlahan nilai pada suatu kolom di dalam tabel bukan baris. Misal akan melakukan penjumlahan seluruh umur mahasiswa

```
SELECT sum(umur)
FROM mahasiswa;
```

```
Command Prompt - mysql -u root
MariaDB [akademik]> SELECT sum(umur)
    -> FROM mahasiswa;
+-----+
| sum(umur) |
+-----+
| 330 |
+-----+
1 row in set (0.00 sec)

MariaDB [akademik]>
```

Nilai sum menunjukkan 330, artinya total usia dari 15 mahasiswa jika dijumlahkan dengan fungsi sum() ialah 330.



Fungsi min()

Ialah fungsi untuk mencari nilai terkecil pada field numerik. Misalnya saat melakukan pencarian data mahasiswa yang memiliki usia termuda.

SELECT min(umur)
FROM mahasiswa;



```
Command Prompt - mysql -u root
MariaDB [akademik]> SELECT min(umur)
    -> FROM mahasiswa;
+-----+
| min(umur) |
+-----+
|      19   |
+-----+
1 row in set (0.03 sec)

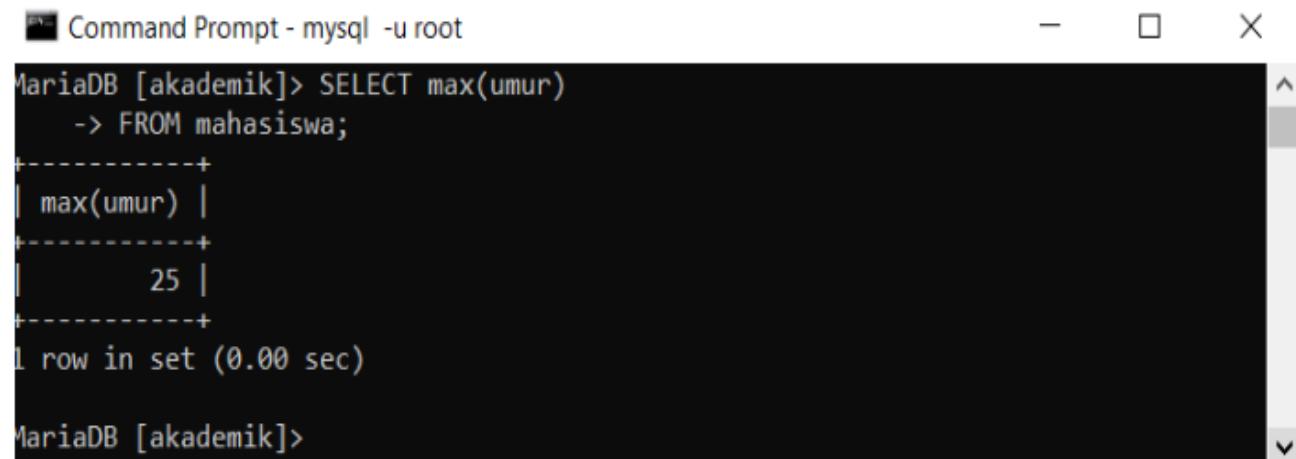
MariaDB [akademik]>
```

Terlihat dari output tersebut kalau usia termuda mahasiswa dari data yang ada ialah 19 tahun.

Fungsi Max()

Dipakai untuk melakukan pencarian nilai terbesar pada suatu field numerik. Misalnya saat ingin mencari usia tertua.

SELECT max(umur)
FROM mahasiswa;



```
Command Prompt - mysql -u root
MariaDB [akademik]> SELECT max(umur)
    -> FROM mahasiswa;
+-----+
| max(umur) |
+-----+
|      25   |
+-----+
1 row in set (0.00 sec)

MariaDB [akademik]>
```



Hasil output menunjukkan usia max atau tertua dalam data tersebut ialah 25 tahun.

Pada modul pembelajaran kali ini kita akan membahas mengenai Menggunakan Function dalam SQL statement. Setelah mengikuti proses pembelajaran pada modul ini, Anda diharapkan dapat memahami dan menggunakan function pada MySQL.





Dalam MySQL, 2 fitur utama terpenting ialah Function dan Stored Procedure, yakni suatu kumpulan perintah atau statement yang disimpan serta dijalankan di server database MySQL. Melalui SP (Stored Procedure), program sederhana berbasis sintaks SQL dapat disusun untuk mengoperasikan fungsi tertentu, sehingga aplikasi bisa lebih efisien dan efektif.

Kelebihan dari penggunaan SP ialah:

Lebih cepat

Tidak seperti ketika dijalankan secara sekuensial di bahasa pemrograman yang harus "bolak-balik" client-server, dengan SP, kumpulan perintah kueri langsung dijalankan di server.

Pemeliharaan yang mudah dengan menghilangkan duplikasi proses.

Operasi yang berjalan di dalam suatu aplikasi terhadap database pada dasarnya sama. Umumnya, terdapat operasi-operasi seperti validasi data yang diinput, melakukan penambahan, perubahan dan penghapusan record, dan lain-lain. Melalui SP, duplikasi proses yang identik fungsinya bisa dihindari yang membuat pemeliharaannya menjadi lebih mudah.

Peningkatan keamanan database.

Database menjadi lebih aman dengan adanya SP, karena aplikasi yang memanggil SP tidak perlu mengetahui isi di dalamnya. Misalnya, dalam proses penambahan data (*insert*), bisa dibuat suatu SP khusus. Sehingga, ketika client atau aplikasi akan melakukan penambahan data (*insert*), nama tabelnya tidak perlu diketahui, cukup memanggil SP tersebut dengan mengirimkan parameter yang diinginkan.

Selanjutnya, SP dari segi bentuk dan sifatnya dibagi menjadi 2: FUNCTION dan PROCEDURE. Perbedaan utama dari keduanya terletak pada nilai yang di-return (dikembalikan). Procedure tidak mempunyai nilai yang di-return, sementara Function punya.

Dalam contoh dibawah ini kasus tentang sistem informasi sekolah, yang mana function yang dibuat akan mengembalikan nilai berupa jumlah siswa dari setiap kelas.



Struktur tabel siswa:

```
CREATE TABLE tb1_siswa (
`nis` char(10) NOT NULL,
`nama` varchar(255) NOT NULL,
`kelas` int(11) NOT NULL,
umur int(5) NOT NULL,
) ENGINE=MyISAM DEFAULT CHARSET=latin1
```

Function untuk mengembalikan jumlah data dari setiap kelas

DELIMITER \$\$

```
CREATE FUNCTION sf_tampil_siswa_kelas (p_kelas int) RETURNS INT DETERMINISTIC
```

BEGIN

```
DECLARE jml INT;
```

```
SELECT COUNT(*) AS jml_kelas INTO jml FROM tb_siswa WHERE kelas = p_kelas;
```

```
RETURN jml;
```

```
END$$
```

DELIMITER [p]

Penjelasannya sebagai berikut:

DELIMITER:

ialah untuk memberitahu ke MySQL soal delimiter yang dipakai, secara default memakai ; jadi bila ada tanda ; mysql akan mengartikan hal tersebut sebagai akhir dari suatu statement, pada contoh diatas delimiter yang digunakan \$\$ jadi akhir statementnya ialah \$\$

CREATE FUNCTION:

ialah header untuk membuat function

RETURNS:

ialah untuk menentukan tipe data yang di return-kan oleh function

DETERMINISTIC/NOT DETERMINISTIC:

ialah untuk menentukan yang bisa memakai function. bisa siapapun (not deterministic) atau hanya user pembuatnya (deterministic).

BEGIN END:

ialah body dari function, sehingga seluruh SQL nya ditulis disini.



Contoh pemanggilannya:

```
select sf_tampil_siswa_kelas("2");
```

Sebuah function tidak bisa memberikan return dalam bentuk resultset dan cuma bisa berupa nilai saja.

untuk penulisan DETERMINISTIC bisa ditulis secara implisit dengan memberikan setting global pada mysql dan secara default bernilai NOT DETERMINISTIC , berikut caranya:

```
SET GLOBAL log_bin_trust_function_creators = 1;
```





Membuat Function sederhana di Mysql

Function ialah salah satu fitur di Mysql, berupa kumpulan SQL yang disimpan dalam database mysql server, umumnya function ini dapat dikombinasikan dengan store procedure.

Dalam contoh pembuatan function dibawah ini akan dibuatkan studi kasus sistem informasi sekolah, dimana function yang dibuat akan mengembalikan nilai berupa jumlah siswa dari setiap kelas.

Berikut langkah-langkahnya :

1. Membuat database

```
create database sekolah1;
```

2. Membuat Tabel

```
CREATE TABLE tbl_siswa (
`nis` char(10) NOT NULL,
`nama` varchar(255) NOT NULL,
`kelas` int(11) NOT NULL
) ENGINE=MyISAM DEFAULT CHARSET=latin1
```

3. Input data

```
insert into tbl_siswa VALUES ('30108927','fahrian','1');
insert into tbl_siswa VALUES ('30108927','ucay','1');
insert into tbl_siswa VALUES ('30108927','ugur','1');
```

4. Membuat Function untuk mengembalikan jumlah data dari setiap kelas

DELIMITER \$\$

```
CREATE FUNCTION sf_tampil_siswa_kelas (p_kelas int) RETURNS INT DETERMINISTIC
```

```
BEGIN
```

```
DECLARE jml INT;
```

```
SELECT COUNT(*) AS jml_kelas INTO jml FROM tbl_siswa WHERE kelas = p_kelas;
```

```
RETURN jml;
```

```
END$$
```

DELIMITER ;



PERTANYAAN

Setelah mengikuti tahap diatas, tuliskan script Function untuk pemanggilannya :

- Tampilkan jumlah kelas pada kelas 1

RINGKASAN

Dalam MySQL, 2 fitur utama terpenting ialah Function dan Stored Procedure, yakni suatu kumpulan perintah atau statement yang disimpan serta dijalankan di server database MySQL.

Melalui SP (Stored Procedure), program sederhana berbasis sintaks SQL dapat disusun untuk mengoperasikan fungsi tertentu, sehingga aplikasi bisa lebih efisien dan efektif



quiz



Fungsi pada statement SELECT bisa dipakai untuk menampilkan data yang berbeda-beda, serta tidak akan menampilkan data yang sama sebanyak dua kali

- a. count
- b. average
- c. distinct
- d. sum

Apabila data dari 2 tabel akan digabungkan maka harus disamakan antara nilai primary key pada tabel A dengan pada tabel B.

- a. Foreign key
- b. Composite key
- c. Primary key
- d. Candidate key

```
SELECT LAST_NAME, SALARY, 12*SALARY+100 FROM EMPLOYEES;
```

```
SELECT LAST_NAME, SALARY, 12*(SALARY+100) FROM EMPLOYEES;
```

Hasil kedua SQL ini berbeda karena

- a. USING PARENTHESES
- b. USING SCOPE
- c. USELESS SCRIPT
- d. USING SCRIPT

Parameter yang digunakan untuk membandingkan 2 nilai adalah

- a. Between
- b. equals
- c. Foreign key
- d. Candidate key

```
SELECT LAST_NAME, SALARY, SALARY+300 FROM EMPLOYEES;
```

Ada berapa jumlah operator pada SQL tersebut?

- a. 2
- b. 1
- c. tidak ada
- d. salah semua



Fungsi agregat yang dipakai untuk menghitung jumlah record ialah

- a. equals
- b. dislike
- c. like
- d. count

Parameter yang mengidentifikasi bahwa stored procedure bisa mengubah parameter dan mengirimkan kembali ke program pemanggil, mode parameter tersebut ialah

- a. OUT
- b. IN
- c. ALL
- d. INOUT

Operator yang mengizinkan kita menentukan beberapa nilai field dalam suatu klausa WHERE ialah

- a. in
- b. 2
- c. =
- d. 1

Sebuah sub program di dalam bahasa pemrograman regular yang tersimpan didalam katalog basis data, disebut:

- a. Trigger
- b. View
- c. Function
- d. Stored Procedure

Parameter yang merupakan mode default ini mengidentifikasi bahwa sebuah parameter bisa di-pass ke stored procedure namun nilainya tidak bisa diubah dari dalam stored procedure, mode parameter tersebut ialah...

- a. IN
- b. ALL
- c. OUT
- d. INOUT



REFERENSI

- Nugroho, A. 2004. Konsep Perancangan Sistem Basis Data. Yogyakarta. Andi Offset.
- Fathansyah. 2002. Basis Data. Bandung. Informatika.
- Jogiyanto. 2002. Analisis dan Desain Sistem Informasi. Yogyakarta. Andi Offset.
- Hariyanto, B. 2004. Sistem Manajemen Basis Data. Bandung. Informatika.
- Prasetyo, D.D. 2003. Administrasi Database Server MySQL. Jakarta. Flex Media Komputindo.
- Abdul, K. 2008. Belajar Database Menggunakan MySQL. Yogyakarta. Andi Offset

