

QUERY LANJUTAN

Menampilkan Record secara Random

MySQL memiliki fungsi khusus yang dapat digunakan untuk menampilkan record secara acak (random). Seperti kita ketahui bahwa pada perintah SELECT record akan ditampilkan secara urut berdasarkan urutan saat penginputan (FIFO = First In First Out).

Berikut ini contoh perintah query untuk menampilkan data pelanggan secara acak (random):

```
SELECT id_pelanggan, nm_pelanggan, email  
FROM pelanggan ORDER BY RAND()
```

Fungsi String

MySQL memiliki banyak sekali fungsi yang berhubungan dengan operasi String. Berikut ini beberapa fungsi string yang disediakan MySQL.

CONCAT (str1, str2, ...)

Fungsi ini digunakan untuk menggabungkan dua atau lebih string (kolom). Sebagai contoh, misalnya akan menggabungkan kolom alamat dan telepon pada tabel pelanggan menjadi sebuah kolom:

```
SELECT kode_karyawan, CONCAT(nama_depan,'',nama_belakang)  
FROM ms_karyawan;
```

CONCAT_WS (separator, str1, str2, ...)

Fungsi ini digunakan untuk menggabungkan dua atau lebih string (kolom) dengan separator diantara masing-masing string. Contoh:

```
SELECT CONCAT_WS (',',kode_cabang,nama_depan,nama_belakang) From ms_karyawan;
```

SUBSTRING (string, awal, panjang)

Fungsi ini digunakan untuk mengambil atau memotong string dimulai dari karakter awal sebanyak panjang karakter. Sebagai catatan bahwa di MySQL, index string dimulai dengan 1, bukan 0. Contoh:

```
SELECT SUBSTRING ('Robby Prihandaya',1,4);
```

LTRIM (string)

Fungsi ini digunakan untuk menghilangkan spasi di awal string (kiri).

```
SELECT LTRIM (' Robby Prihandaya');
```

RTRIM (string)

Fungsi ini digunakan untuk menghilangkan spasi di akhir string (kanan).

```
SELECT RTRIM ('Robby Prihandaya ');
```

TRIM (string)

Fungsi ini digunakan untuk menghilangkan spasi di awal dan akhir string (kiri dan kanan)

```
SELECT RTRIM (' Robby Prihandaya ');
```

REPLACE (string, from_str, to_str)

Fungsi ini digunakan untuk mengganti suatu string dengan string yang lain.

```
SELECT REPLACE ('www.mysql.com', 'w', 'x');
```

REPEAT (string, jumlah)

Fungsi ini digunakan untuk menduplikasi suatu string sebanyak jumlah.

```
SELECT REPEAT ('Mont', 3);
```

REVERSE (string)

Fungsi ini digunakan untuk membalik string. Contoh :

```
SELECT REVERSE ('mysql.com');
```

LOWER dan UPPER (string)

Fungsi ini digunakan untuk Memperbesar dan Memperkecil String

```
SELECT LOWER ('MySQL');
```

COUNT(range)

Fungsi digunakan untuk mengambil jumlah baris dari suatu query.

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

MAX dan MIN(range)

Fungsi digunakan untuk mengambil nilai terbesar dan terkecil dari suatu ekspresi (query)

```
SELECT MAX(nilai) FROM nilai_ujian;  
SELECT MIN(nilai) FROM nilai_ujian;
```

SUM(range)

Fungsi digunakan untuk menjumlahkan total nilai dari suatu ekspresi (query)

```
SELECT SUM(nilai) FROM nilai_ujian;
```

AVG(range)

Fungsi digunakan untuk menghitung rata-rata nilai dari suatu ekspresi (query)

```
SELECT AVG(nilai) FROM nilai_ujian;
```

MD5(str)

Fungsi digunakan untuk melakukan enkripsi suatu string str menggunakan metode md5. Fungsi ini juga tidak dapat didekripsi.

```
SELECT MD5('udin');
```

LAST_INSERT_ID()

Fungsi digunakan untuk mengambil id terakhir dalam proses insert dimana

tabelnya mengandung field yang bersifat AUTO INCREMENT.

```
SELECT LAST_INSERT_ID();
```

Pengelompokkan Hasil Query dengan GROUP BY

Hasil query terkadang perlu dikelompokkan berdasarkan kriteria atau kondisi tertentu. Misalnya kita akan menampilkan jumlah barang yang dibeli untuk masing-masing transaksi (pemesanan).

Perhatikan perintah query berikut ini dan lihat hasilnya:

```
SELECT pesan.id_pesan, pesan.tgl_pesan,  
SUM(detil_pesan.jumlah) as jumlah  
FROM pesan, detil_pesan  
WHERE pesan.id_pesan=detil_pesan.id_pesan  
GROUP BY id_pesan;
```

Selain hasil di atas, kita juga dapat menggunakan tambahan WITH ROLLUP di belakang group by untuk menampilkan jumlah total seluruh barang. Berikut ini perintah query dan hasilnya:

```
SELECT pesan.id_pesan, pesan.tgl_pesan,  
SUM(detil_pesan.jumlah) as jumlah  
FROM pesan, detil_pesan  
WHERE pesan.id_pesan=detil_pesan.id_pesan  
GROUP BY id_pesan WITH ROLLUP;
```

SubSELECT

Mulai versi 4.1, MySQL mendukung perintah query SubSELECT dimana memungkinkan untuk melakukan query di dalam query. Misalnya kita akan menampilkan data yang kondisinya merupakan hasil dari query lain.

Dan berikut ini beberapa contoh perintah query yang menggunakan SubSELECT.

1. Menampilkan daftar pelanggan yang pernah melakukan transaksi (pemesanan).

```
SELECT id_pelanggan, nm_pelanggan FROM pelanggan  
WHERE id_pelanggan IN (SELECT id_pelanggan FROM  
pesan);
```

2. Menampilkan data pemesanan dengan jumlah barang terbanyak.

```
SELECT id_pesan, jumlah FROM detil_pesan  
WHERE jumlah = ( SELECT MAX(jumlah) FROM  
detil_pesan);
```

