

Query MySQL untuk Range Umur

Tutorial ini ditulis oleh **Achmad Solichin**
dan sudah dipublikasikan di <http://achmatim.net> pada tanggal 3 Oktober 2014

Beberapa hari yang lalu saya mendapatkan pertanyaan menarik melalui email dari seseorang. Inti pertanyaannya adalah bagaimana perintah query untuk menghitung jumlah mahasiswa berdasarkan range umur. Melalui postingan ini, saya ingin menjawab pertanyaan tersebut secara langkah demi langkah. Semoga bermanfaat.

Contoh Data

Untuk mempermudah pembahasan, kita akan menggunakan contoh data mahasiswa dengan struktur dan data sebagai berikut:

nim	nama	tgllahir
0111500382	Andy	1982-06-10
0211500220	Indy	1980-02-12
0311500292	Edy	1986-10-10
0911500232	Ani	1988-11-15
1011500299	Ida	1990-12-20
1111500200	Ina	1993-10-12
1211500199	Eni	1995-03-12
1211500333	Uni	1996-01-10
1011501222	Juki	1990-05-15
1111500283	Okky	1991-01-25

Buatlah terlebih dahulu tabel dan data di atas di MySQL.

Dari tabel di atas, tampilan yang diinginkan adalah sebagai berikut:

RANGE UMUR	JUMLAH
<20	2
20-24	4
25-29	2
>=30	2

Query untuk Mencari Umur

Langkah pertama dari data di atas adalah menghitung dan menampilkan umur untuk setiap mahasiswa. Teknik untuk menghitung umur beragam. Saya pernah saya contohkan querynya pada postingan mengenai [variasi perintah berhubungan waktu dan tanggal](#), menggunakan pengurangan tanggal sederhana. Selain teknik tersebut, kita juga bisa menggunakan fungsi **TIMESTAMPDIFF()** yang tersedia di MySQL.

Berikut ini query dan hasil untuk menghitung umur setiap mahasiswa. Umur dihitung dengan mencari selisih antara tanggal lahir dan tanggal sekarang menggunakan fungsi **TIMESTAMPDIFF()**. Parameter pertama dari fungsi tersebut adalah "YEAR" untuk menghasilkan format tanggal dalam bentuk tahun saja. Parameter kedua adalah field "tgllahir" yang merupakan tanggal lahir setiap mahasiswa dan parameter ketiga adalah fungsi **CURDATE()** yang akan menghasilkan tanggal saat ini.

Query 1. Menghitung Umur Berdasarkan Tanggal Lahir

```
SELECT nim, nama, tgllahir,  
TIMESTAMPDIFF(YEAR, tgllahir, CURDATE()) AS umur  
FROM mhs
```

Hasil eksekusi query tersebut dengan menggunakan tools [Adminer](#) adalah:

nim	nama	tgllahir	umur
0111500382	Andy	1982-06-10	32
0211500220	Indy	1980-02-12	34
0311500292	Edy	1986-10-10	27
0911500232	Ani	1988-11-15	25
1011500299	Ida	1990-12-20	23
1011501222	Juki	1990-05-15	24
1111500200	Ina	1993-10-12	20
1111500283	Oky	1991-01-25	23
1211500199	Eni	1995-03-12	19
1211500333	Uni	1996-01-10	18

10 rows (0.001 s) [Edit](#), [EXPLAIN](#), [Export](#)

Query untuk Mengelompokkan Umur

Lalu bagaimana mengelompokkan umur yang sudah diperoleh pada query sebelumnya ke dalam distribusi (range) tertentu. Secara sederhana, kita akan mengelompokkan umur mahasiswa berdasarkan sebaran distribusi sebagai berikut:

- < 20 tahun
- 20-24 tahun
- 25-29 tahun
- >= 30 tahun

Query 2. Mengelompokkan Umur

```
1  SELECT
2      CASE
3          WHEN umur < 20 THEN '... - 20'
4          WHEN umur BETWEEN 20 and 24 THEN '20 - 24'
5          WHEN umur BETWEEN 25 and 29 THEN '25 - 29'
6          WHEN umur >= 30 THEN '30 - ...'
7          WHEN umur IS NULL THEN '(NULL)'
8      END as range_umur,
9      COUNT(*) AS jumlah
10 FROM (select nim, nama, tgllahir, TIMESTAMPDIFF(YEAR, tgllahir,
11 CURDATE()) AS umur FROM mhs) as dummy_table
12 GROUP BY range_umur
ORDER BY range_umur
```

Berikut ini beberapa catatan penting dari query 2 di atas.

- Untuk nama tabel digantikan dengan tabel dummy yang merupakan hasil perintah query 1 (query untuk menghitung umur).
- Untuk membuat range umur, digunakan struktur CASE-WHEN-THEN. Penjelasan mengenai struktur kondisi ini dapat ditemukan di [dokumentasi MySQL](http://dev.mysql.com/doc/refman/5.6/en/control-flow-functions.html#operator_case)¹.
- Fungsi COUNT() digunakan untuk menghitung jumlah record berdasarkan **range_umur** yang telah ditentukan. Fungsi COUNT() merupakan salah satu fungsi agregasi, sehingga diperlukan tambahan [perintah GROUP BY](#) untuk mengelompokkan hasil penjumlahan record. Dalam hal ini dikelompokkan berdasarkan **range_umur**.
- Pengurutan dengan ORDER BY dapat ditambahkan agar tampilan menjadi terurut.

1 http://dev.mysql.com/doc/refman/5.6/en/control-flow-functions.html#operator_case

Hasil dari perintah query 2 di atas adalah sebagai berikut:

range_umur	jumlah
... - 20	2
20 - 24	4
25 - 29	2
30 - ...	2

4 rows (0.002 s) [Edit](#), [EXPLAIN](#), [Export](#)

Alternatif lain adalah menggunakan query 3 sebagai berikut:

Query 3. Mengelompokkan Umur

```

1  SELECT
2      COUNT(IF(umur < 20,1,NULL)) AS '... - 20',
3      COUNT(IF(umur BETWEEN 20 and 24,1,NULL)) AS '20 - 24',
4      COUNT(IF(umur BETWEEN 25 and 29,1,NULL)) AS '25 - 29',
5      COUNT(IF(umur >= 30,1,NULL)) AS '30 - ...'
6  FROM (select nim, nama, tgl_lahir, TIMESTAMPDIFF(YEAR, tgl_lahir,
CURDATE()) AS umur FROM mhs) as dummy_table

```

Terlihat query 3 di atas lebih sederhana dan singkat. Namun demikian, hasilnya ditampilkan dalam bentuk kolom. Perhatikan tampilan hasil query 3 berikut ini! Terlihat bedanya kan dibanding hasil dari query sebelumnya?

... - 20	20 - 24	25 - 29	30 - ...
2	4	2	2

1 row (0.001 s) [Edit](#), [EXPLAIN](#), [Export](#)

Kesimpulan

Pengelompokkan data seperti pada kasus umur yang dibahas di tutorial ini seringkali ditemui dalam pembangunan suatu aplikasi berbasis web. Contoh lainnya adalah distribusi data berdasarkan agama serta berdasarkan nilai tertentu. Tekniknya pun sebenarnya beragam. Tinggal pilih mana yang paling sesuai dan paling sederhana, sesuai kebutuhan masing-masing. Semoga tutorial ini bermanfaat.

Links

- Tutorial ini juga telah dipublikasikan di <http://achmatim.net/2014/10/03/query-mysql-untuk-range-umur/>

Semoga tutorial ini bermanfaat untuk kita semua dan berbagilah hal baik ke sebanyak-banyaknya orang maka kebahagiaan akan menghampiri Anda. Satu Karya untuk Indonesia!

Tentang Penulis



Achmad Solichin. Adalah Lulusan Teknik Informatika, [Fakultas Teknologi Informasi, Universitas Budi Luhur](#), Jakarta (S1, 2005) dan [Magister Teknologi Informasi, Universitas Indonesia](#) (S2, 2010). Saat ini sedang menempuh program Doktorat dalam bidang Ilmu Komputer di [Universitas Gadjah Mada](#), Jogjakarta. Kegiatan sehari-hari adalah sebagai Dosen di Universitas Budi Luhur (<http://www.budiluhur.ac.id>). Kegiatan lain aktif sebagai programmer, web developer, system analyst, konsultan dan memberikan pelatihan di berbagai bidang komputer serta membuat tutorial-tutorial praktis di bidang komputer. Penulis memiliki situs utama di <http://achmatim.net> yang berisi berbagai tutorial praktis di bidang komputer serta menyediakan buku gratis komputer. Penulis dapat dihubungi melalui email di achmad.solichin@budiluhur.ac.id dan achmatim@gmail.com, YM [achmatim](#), Facebook [achmatim](#) dan Twitter [@achmatim](#).

Lisensi Dokumen



Seluruh isi dalam dokumen ini dapat digunakan, dimodifikasi dan disebarluaskan secara bebas untuk tujuan pendidikan, pembelajaran dan bukan komersial (*non profit*), dengan syarat tidak menghilangkan, menghapus atau mengubah atribut penulis dokumen ini dan pernyataan dalam lisensi dokumen yang disertakan di setiap dokumen. Tidak diperbolehkan mengkomersialkan tutorial ini kecuali mendapatkan ijin terlebih dahulu dari penulis.