

Pertemuan 10

View

View bagaikan sebuah jendela untuk menampilkan data apa saja yang boleh ditunjukkan atau diubah.

Data yang disimpan oleh seorang user bisa saja diperlukan oleh *user* yang berbeda walau ada kemungkinan tidak semua kolom dibutuhkan. Pertimbangannya adalah perbedaan otoritas, perspektif, lokasi kerja, departemen dan sebagainya sehingga data suatu tabel yang diimplementasikan dengan *view* dapat dipandang secara berbeda menurut perspektif *user*.

Sumber data view dapat berasal dari tabel atau view lain. Mirip dengan tabel, update, delete dan insert dapat dilakukan pada *view* sehingga perubahan itu akan direfleksikan pada tabel asalnya.

View dapat diperlakukan sama seperti tabel. Perintah untuk memanipulasi data pada view sama seperti halnya pada tabel.

Namun berbeda dengan tabel, view tidak dapat menyimpan data, view hanya menyimpan definisi query pada *data dictionary* dan tidak memerlukan ruang penyimpanan data.

Penerapan view dapat diaplikasikan pada situasi berikut:

- membatasi akses sesuai otoritas user
- memudahkan pemahaman terhadap kolom penampung data yang mungkin berbeda dengan definisi kolom pada tabel dasar.
- Menyederhanakan pandangan *user* terhadap data
- Menangani data kompleks
- Memudahkan penggunaan query yang berulang karena disimpan sebagai stored-query.

Mendefinisikan View

View didefinisikan melalui query pada tabel atau view lain dengan perintah `CREATE VIEW` dan dihapus dengan `DROP VIEW`. Untuk lengkapnya sintaksis yang dapat digunakan adalah sebagai berikut:

```
CREATE VIEW <nama_view> AS  
    pernyataan_select;
```

Menghapus view

Penghapusan view tidak akan berefek pada tabel yang digunakan untuk membuat view tersebut. Perintah yang dapat digunakan untuk menghapus sebuah view:

```
DROP VIEW <nama_view>;
```

Memodifikasi View

Untuk memodifikasi sebuah view, dapat digunakan tambahan perintah `OR REPLACE` setelah klausa `CREATE`.

```
CREATE OR REPLACE VIEW <nama_view> AS  
    pernyataan_query;
```

Adapun beberapa keterbatasan untuk sebuah view di dalam MySQL adalah:

- Di dalam sebuah perintah select tidak boleh mengandung variabel maupun parameter.
- Semua tabel atau view lain yang ada di dalam sebuah view harus sudah ada sebelumnya.
- Tidak boleh menunjuk ke sebuah tabel TEMPORARY, dan tidak dapat menciptakan view TEMPORARY.
- Tidak dapat diasosiasikan dengan trigger.
- Alias yang digunakan dalam perintah SELECT hanya boleh selebar 64 karakter untuk lebar kolom dengan panjang karakter maksimum nama alias sepanjang 256 karakter.

Catatan: dalam membuat sebuah view, cara termudah melakukannya adalah memastikan bahwa query yang akan dibuat menjadi sebuah view telah dapat tereksekusi dengan baik terlebih dahulu, dengan hasil yang benar. Apabila hasil query telah benar, tambahkan perintah CREATE VIEW (lihat sintaks) pada bagian atas query tersebut.

PRAKTIK

Dalam bagian praktik ini, mahasiswa diasumsikan telah menguasai materi JOIN dan KONDISI pada query. Oleh karena itu mahasiswa dituntut untuk mengerjakan sendiri soal-soal praktik yang ada di bawah ini dengan view nomor 1 sebagai contoh.

Kerjakan latihan berikut ini dan tampilkan hasilnya (nomor 1 sebagai contoh).

1. Buatlah sebuah view untuk menampilkan nama-nama dosen yang mengajar di tahun akademik 2014 semester 1. Beri nama view ini dengan dosen_2013_satu.

Diasumsikan nilai semester dalam *field* semester pada tabel KRS adalah 1,2,3,...,8 yang mengidentifikasikan pada semester berapa MK tersebut ditawarkan, maka query-nya adalah sebagai berikut.

```
CREATE VIEW dosen_2014_satu AS
SELECT nama_dosen
FROM dosen d, KRS
WHERE d.NID=KRS.NID AND tahun_akademik = 2014 AND semester = 1;
```

Untuk memanggil view tersebut dapat digunakan perintah:

```
SELECT *
FROM dosen_2014_satu;
```

2. Buatlah sebuah view bernama umur_mhs untuk menampilkan umur mahasiswa. Tampilkan hasil view tersebut! (Petunjuk: carilah tahun dari tanggal sekarang dan tanggal lahir dan kurangkan keduanya)
3. Buatlah sebuah view untuk menampilkan nama-nama dosen jurusan TI beserta dengan pendidikan terakhirnya dan beri nama view tersebut dengan nama dosen_TI.

```
CREATE VIEW v_dosen_TI AS
SELECT nama_dosen, pendidikan_terakhir
FROM dosen
WHERE jurusan='TI';
```

Cobalah memanggil view tersebut dengan menggunakan query berikut ini:

```
SELECT nama_dosen, tgl_lahir  
FROM v_dosen_TI;
```

Apa yang terjadi? Apa sebabnya?

Tampilkan semua isi view v_dosen_TI dengan:

```
SELECT * FROM v_dosen_TI;
```

4. Buatlah sebuah view untuk menampilkan semua nama mahasiswa yang mengambil kuliah di semester 3 beri nama view tersebut dengan nama kuliah_sem_3. Tampilkan isi view tersebut!
5. Buatlah sebuah view untuk menampilkan dosen yang jenjang D3, beri nama view tersebut dengan nama dosen_d3. Tampilkan isi view tersebut!
6. Buatlah sebuah view yang diberi nama mhs_sbd_A untuk menampilkan nama-nama mahasiswa dan asal sekolahnya yang pernah mendapatkan nilai A pada mata kuliah 'SBD'. Tampilkan isi view tersebut!