



Pertemuan VII

TRIGGERS



TUJUAN

1. Mahasiswa mampu untuk menjelaskan tentang jenis-jenis dari triggers.
2. Mahasiswa mampu untuk menjelaskan tentang database triggers dan penggunaannya.
3. Mahasiswa mampu untuk membuat triggers.
4. Mahasiswa mampu untuk menjelaskan aturan-aturan di lingkungan trigger.
5. Mahasiswa mampu untuk meremove triggers.

PENGERTIAN DARI TRIGGER

- Trigger merupakan nama object database yang berhubungan dengan sebuah tabel dan aktif ketika suatu kejadian muncul pada suatu tabel tertentu
- Trigger adalah :
 - Sebuah blok PL/SQL atau prosedur PL/SQL yang berhubungan dengan table, view, schema atau database.
 - Akan dieksekusi secara implicit pada saat sebuah kejadian tertentu terjadi.
- Trigger merupakan bentuk khusus suatu Store Procedure yang akan dilaksanakan secara otomatis jika terjadi perubahan (INSERT, UPDATE dan DELETE) pada sebuah tabel.

PENGERTIAN DARI TRIGGERS (cont)



- Ada 2 jenis trigger yaitu :
 - Application trigger: Terjadi pada saat sebuah kejadian terjadi dengan aplikasi tertentu.
 - Database trigger : Terjadi pada saat terjadi sebuah perubahan data seperti DML pada table (INSERT, UPDATE atau DELETE)

KEUNTUNGAN TRIGGER



Dalam sebuah tabel dapat mempunyai beberapa trigger. Trigger sangat berguna karena dapat secara otomatis dilaksanakan dalam server, sehingga menyederhanakan pemrograman, sekaligus menjaga konsistensi informasi dalam database

KEUNTUNGAN TRIGGER (cont)

Keuntungan penggunaan Trigger antara lain :

- a. Standarisasi (penyeragaman) proses. Trigger dibuat satu kali dan tersimpan dalam database, sehingga semua client yang menggunakan database tersebut sekaligus adalah pengguna Trigger. Dengan demikian seluruh client akan menggunakan operasi yang sama terhadap suatu proses dalam database, karena Trigger yang dipakai sama.
- b. Menyederhanakan pemrograman. Apabila Trigger sudah dibuat dalam database server, seluruh program dalam bahasa apapun, yang mengakses database tersebut akan secara otomatis menggunakan Trigger yang ada tanpa perlu membuat perintahnya dalam program aplikasi.

KEUNTUNGAN TRIGGER (cont)

- c. Mudah diperbaharui. Apabila Trigger pernah di-update dalam database server, semua client akan menggunakan perbaruan yang terakhir
- d. Kecepatan dan efisiensi proses. Oleh karena Trigger disimpan dalam server dalam kode yang sudah terkompilasi, dan dilaksanakan dalam server sehingga lalu-lintas jaringan menjadi berkurang.
- e. Memudahkan kerja secara tim. Apabila Trigger dan Stored Procedure sudah dibuat, anggota tim kerja dapat menggunakan bahasa pemrograman yang berbeda dalam mengaksesnya, karena Trigger disimpan dan dilaksanakan oleh server, tanpa melihat bahasa pemrograman apa yang dipakai oleh pengguna database.

PETUNJUK UNTUK DISAIN TRIGGER

- Gunakan trigger untuk memastikan pada saat sebuah operasi tertentu terbentuk, aksi yang berhubungan dengannya juga terbentuk.
- Gunakan database trigger hanya untuk operasi yang global.
- Tidak mendisain trigger untuk fungsi-fungsi yang telah dibuat oleh server database dan tidak boleh membuat duplikasi trigger.
- Buatlah store procedur dan invoke prosedur dalam sebuah trigger jika kode PL/SQL terlalu panjang.
- Gunakan trigger untuk menghasilkan hasil dengan tingkat interdependensi yang tinggi, yang sulit untuk dimaintain dalam aplikasi yang besar.
- Database trigger akan terjadi untuk setiap user pada saat sebuah kejadian yang mentrigger dibuat.

MEMBUAT TRIGGER



Syntaks

```
CREATE
```

```
  [DEFINER = { user | CURRENT_USER }]
```

```
  TRIGGER trigger_name trigger_time trigger_event
```

```
  ON tbl_name FOR EACH ROW trigger_stmt
```

Sebelum membuat coding untuk trigger, tentukan terlebih dahulu nilai-nilai dari komponen trigger yaitu: trigger timing, trigger event dan trigger type.

MEMBUAT TRIGGER (cont)

Part	Keterangan	Nilai yang mungkin
Trigger timing (TRIGGER_TIME)	Kapan trigger terjadi sesuai dengan trigger event.	BEFORE AFTER
Trigger_event	Operasi manipulasi data tertentu pada tabel atau view yang menyebabkan trigger dilaksanakan	INSERT UPDATE DELETE
Trigger_stmt	Aksi yang terjadi pada saat trigger terjadi	Block PL/SQL

MEMBUAT TRIGGER (cont)

- Trigger timing → kapan trigger terjadi ?
 - BEFORE → Mengeksekusi body trigger sebelum mentrigger event DML pada sebuah table.
 - AFTER → Mengeksekusi body trigger setelah mentrigger event DML pada sebuah table

MEMBUAT TRIGGER (cont)

- Triggering user event (Trigger_event) → DML statement yang menyebabkan trigger dieksekusi. Peristiwa atau kejadian yang memicu terjadinya Trigger. Kita dapat menggunakan statement INSERT, UPDATE, DELETE.
 - Saat menggunakan UPDATE maka perlu memasukkan list kolom, untuk mengidentifikasi kolom mana yang berubah,.
 - Tidak perlu menspesifikasikan list kolom untuk statement INSERT atau untuk DELETE karena perintah tersebut akan berpengaruh pada suatu row.
 - Trigger dapat berisi satu, dua atau ketiganya dari operasi DML.INSERT or UPDATE or DELETE

MEMBUAT TRIGGER (cont)



- Trigger body (trigger_statement) → aksi apa yang trigger buat ?
Trigger body merupakan pernyataan atau perintah yang dilaksanakan Trigger. Jika pernyataan yang dilaksanakan lebih dari satu perintah, beberapa perintah dapat dimasukkan ke dalam block BEGIN .. END .

Temporer Tabel (Old dan New)

- Perlu diketahui pada saat proses menambahkan data, MySQL akan membuat semacam tabel temporer dengan struktur yang sama dengan tabel yang akan diisi. Tabel temporer diberi nama **new**. Pada saat proses pemasukan data (INSERT), nilai-nilai yang akan dimasukkan ke dalam tabel disalin ke dalam tabel NEW
- Hal yang hampir sama juga berlaku pada proses penghapusan data (DELETE). Pada perintah menghapus data akan dibuat tabel temporer OLD dengan struktur yang sama dengan tabel yang record-nya akan dihapus.
- Sedangkan untuk proese updating data (UPDATE) kedua tabel temporer yaitu OLD dan NEW dibuat. Old akan menyimpan data yang dihapus (sebelum di-update) dan New berisi data pengganti (data baru).

Contoh

Perhatikan tabel berikut ini:

	KodeBarang	NamaBarang	Satuan	HargaBeli	Stock	HargaJual	JenisBarang
<input type="checkbox"/>	BG.001	Buku Gambar Dora	PCS	900	0	1000	Inventory
<input type="checkbox"/>	BG.002	Buku Gambar 10 Besar	PCS	1500	0	2000	Inventory
<input type="checkbox"/>	PS.001	Pensil Staedler	PCS	1200	0	1500	Inventory
<input type="checkbox"/>	RT.001	Rautan Kecil	PCS	200	0	400	Inventory
<input type="checkbox"/>	RT.002	Rautan Besar	PCS	700	0	800	Inventory
<input type="checkbox"/>	RT.003	Rautan Kotak Sedang	PCS	0	0	0 (NULL)	

Misalkan stock pada tabel barang masih kosong. Kemudian akan dibuat Trigger yang otomatis akan mengupdate stock pada tabel barang jika terdapat proses pembelian

Contoh (cont)

Adapun tabel DetailPembelian mempunyai struktur sebagai berikut:

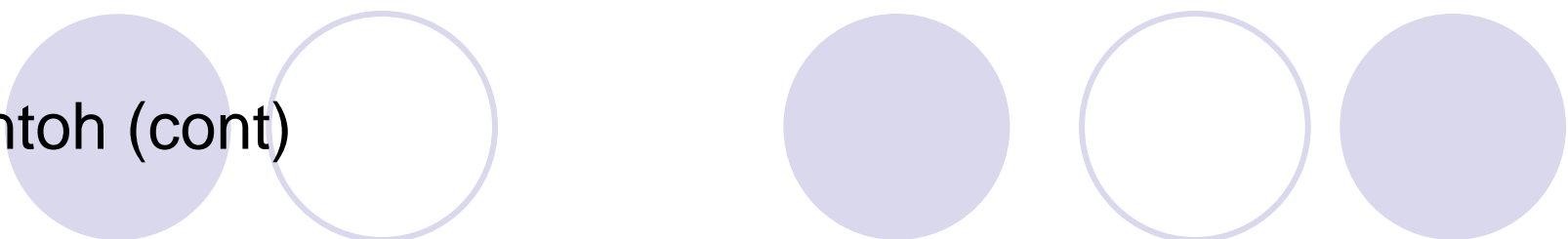
Table	Create Table
-----	-----
detailpembelian	<pre>CREATE TABLE `detailpembelian` (`NoReference` varchar(10) NOT NULL default '', `KodeBarang` varchar(12) NOT NULL default '', `HargaBeli` double NOT NULL default '0', `JumlahBarang` int(8) NOT NULL default '0', `Operator` varchar(8) NOT NULL default '') ENGINE=MyISAM DEFAULT CHARSET=latin1</pre>

Contoh (cont)

Kemudian dibuat Trigger dengan nama triggerbeli

```
DELIMITER $$
create trigger `trigerbeli` BEFORE INSERT on
  `detailpembelian`
for each row BEGIN
  UPDATE barang SET
  Stock=IF(ISNULL(Stock),0,Stock)+new.JumlahBarang
  WHERE KodeBarang = new.kodeBarang;
END;
$$
DELIMITER ;
```

Contoh (cont)



Contoh diatas berarti perintah untuk membuat Trigger bernama **trigerbeli** yang akan otomatis dilaksanakan sebelum tabel detailpembelian diisi dengan suatu record. Pada setiap sebelum satu baris (record) ditambahkan ke dalam tabel DetailPembelian, tabel Barang akan di-update, yaitu isi kolom Stock di-update dengan isi sebelumnya ditambah banyak barang yang akan ditambahkan ke dalam tabel detailpembelian, untuk barang dalam tabel barang yang kode barangnya sama dengan kode barang dari baris yang akan disisipkan ke dalam tabel detailpembelian.

Contoh (cont)

- Perhatikan bahwa Trigger menggunakan pernyataan **new.JumlahBarang** dan **new.kodeBarang**.
- Sebelum sebuah trigger dibuat dan belum dihapus, Trigger akan selalu siap dan akan bekerja sesuai dengan tugasnya, sehingga siapapun dan dari program apapun menambahkan data ke dalam tabel detailpembelian, berarti pula akan mengupdate jumlah stok dalam tabel Barang.

Isikan tabel detailpembelian dengan data :

```
INSERT INTO detailpembelian`  
( '1', 'PS.001', 900, 50, 'SAYA' );
```

Berapakah stock untuk tabel Barang untuk barang PS.001 ???

MENGHAPUS DAN MEMPERBAIKI TRIGGER

MySQL tidak menyediakan fasilitas untuk memperbaiki atau mengedit Trigger. Oleh karena itu cara untuk memperbaiki Trigger adalah dengan menghapusnya, kemudian membuat yang baru.

```
DROP TRIGGER Trigger_name
```