

# BAB 25 Trigger

Trigger, seperti terjemahannya yaitu pemicu adalah sekumpulan statemen SQL baik prosedural atau deklarasi, yang melekat pada suatu tabel, yang akan dieksekusi jika terdapat perubahan data pada tabel tersebut.

Trigger mirip dengan stored procedure dimana di dalamnya dapat dijalankan berbagai statemen SQL seperti CREATE, INSERT, UPDATE, IF-ELSEIF-ELSE, dll, bedanya, trigger dijalankan otomatis oleh MySQL, sedangkan Stored Procedure dipanggil secara manual menggunakan statemen CALL.

Kapan MySQL menjalankan trigger? MySQL menjalankan trigger hanya jika terjadi operasi data pada tabel yaitu: (1) penambahan baris pada tabel (INSERT), (2) pembaharuan data (UPDATE), dan (3) penghapusan baris (DELETE). Trigger dapat di jalankan baik sebelum atau sesudah ketiga operasi tersebut.

## 25.1. Membuat Trigger

Untuk membuat trigger kita gunakan statemen CREATE TRIGGER dengan format sebagai berikut:

---

```
1. CREATE
2.     TRIGGER nama_trigger
3.     {BEFORE | AFTER}
4.     {INSERT | UPDATE | DELETE}
5. ON  nama_tabel
6. FOR EACH ROW
7.     statemen
```

---

Penjelasan:

- **nama\_trigger.** Nama trigger mengikuti aturan penamaan objek database. Penamaan trigger sangat disarankan menggunakan pola sesuai fungsi trigger tersebut. Satu trigger hanya diperbolehkan

menggunakan salah satu dari opsi BEFORE dan AFTER dan salah satu dari opsi INSERT, UPDATE, dan DELETE (misal BEFORE UPDATE saja atau AFTER INSERT saja), nama trigger sebaiknya disertai dengan opsi yang digunakan, misal nama\_trigger\_before\_update atau cukup nama\_trigger\_bu.

Sebagai tambahan, dalam satu database (schema), tidak boleh ada nama trigger yang sama.

- **BEFORE** atau **AFTER**. Merupakan trigger\_time yang menentukan kapan trigger akan dijalankan, apakah sebelum DML (INSERT, UPDATE, DELETE) atau sesudah DML dieksekusi.
- **INSERT | UPDATE | DELETE**. Merupakan trigger\_event yang menentukan DML yang akan diterapkan pada trigger.
- **ON nama\_tabel**. Mendefinisikan nama tabel yang akan dikenai event, artinya jika terjadi perubahan data pada tabel tersebut, trigger akan dijalankan.
- **FOR EACH ROW**. Artinya trigger akan dijalankan pada setiap baris yang dikenai operasi DML. Standar ANSI menggunakan klausa FOR EACH STATEMEN, yang hingga saat ini belum disupport oleh MySQL.
- **statemen**. Merupakan statemen yang akan dieksekusi ketika trigger dijalankan, jika statemen terdiri dari lebih dari satu baris, maka gunakan compound statemen (statemen yang diawali dengan BEGIN dan diakhiri dengan END ).

Keyword INSERT, UPDATE, atau DELETE menentukan kapan trigger akan diaktifkan, perbedaan ketiganya adalah:

- **INSERT** berarti trigger akan aktif jika ada data ditambahkan pada tabel, yang biasanya dilakukan melalui statemen INSERT. Selain itu, trigger ini juga aktif pada implisit INSERT, seperti pada statemen LOAD DATA dan REPLACE, dimana pada kedua statemen tersebut, MySQL di belakang layar akan menjalankan statemen INSERT
- **UPDATE** berarti trigger akan aktif ketika ada perubahan data pada tabel yang dilakukan melalui statemen UPDATE.

- DELETE berarti trigger akan aktif ketika ada baris yang dihapus misal pada statemen DELETE atau REPLACE. Khusus untuk statemen REPLACE, jika statemen tersebut perlu untuk menghapus data, maka trigger ini akan aktif. Statemen DROP TABLE dan TRUNCATE TABLE tidak mengaktifkan trigger ini, karena tidak mengandung implisit DELETE.
- Sebagai tambahan, perubahan data terkait foreign key (ON DELETE CASCADE dan ON UPDATE CASCADE) tidak mengaktifkan trigger.

Sebagai contoh, kita akan membuat trigger dengan melibatkan tabel buku, tabel penjualan\_detail, dan tabel penjualan seperti yang kita gunakan pada bab sebelumnya:

Tabel buku:

```
mysql> SELECT id_buku, id_kategori, judul, harga, stok FROM buku LIMIT 5;
```

id_buku	id_kategori	judul	harga	stok
1	3	Pemrograman Database Menggunakan MySQL	59000	5
2	2	PHP dan MySQL Langkah Demi Langkah + CD	75000	6
3	3	MySQL Untuk Pemula	34800	4
4	2	Kumpulan Aplikasi PHP untuk Pemula	34800	3
5	2	Pemrograman PHP Dan MySQL Untuk Pemula	69000	2

5 rows in set (0.00 sec)

Tabel penjualan\_detail:

```
mysql> SELECT * FROM penjualan_detail LIMIT 5;
```

id_trx_detail	id_trx	id_buku	jml_barang	harga_satuan	diskon	total
1	2	9	1	46800	0	46800
2	2	4	1	34800	0	34800
3	2	6	2	33800	0	67600
4	3	9	1	46800	0	46800
5	3	10	2	39800	0	79600

5 rows in set (0.02 sec)

Tabel penjualan: