

# TRIGGER

## Praktikum Sistem Basis Data

Gentisya Tri Mardiani, S.Kom., M.Kom



# TRIGGER

- Trigger adalah sebuah objek database yang diasosiasikan dengan sebuah tabel dan akan aktif (terpicu/ trigger) ketika sebuah event terjadi pada tabel tersebut.
- Trigger hanya terjadi ketika ada eksekusi INSERT, DELETE, UPDATE pada tabel yang bersangkutan.
- Waktu eksekusi trigger yang mungkin terjadi yaitu BEFORE dan AFTER dari statement SQL nya



# Keuntungan

- Trigger dapat digunakan untuk mengubah data sebelum proses INSERT dilakukan atau untuk memberikan nilai default.  
Misalnya mengubah data diluar nilai yang diperbolehkan.
- Dapat menyimpan data suatu record ke tabel lain (misal history) sebelum data tersebut diupdate atau didelete.  
Sehingga semua perubahan data data terlacak.



# Struktur Trigger

```
CREATE TRIGGER nama_trigger  
  { BEFORE | AFTER }  
  { INSERT | UPDATE | DELETE }  
ON nama_tabel  
FOR EACH ROW  
  statement sql
```



# Mengakses nilai baru dan lama

- Data lama dapat di-*reference* dengan record OLD dan data baru dapat di-*reference* dengan record NEW.
- Untuk mengacu ke sebuah field dapat ditulis dengan **NEW.namafield** atau **OLD.namafield**



# Contoh

- Tabel pegawai  
{ NIP, Nama, Tgl\_Lahir, Kelamin, Alamat, Kota, Tgl\_masuk, Kode\_jabatan, Kode\_area }
- Buatlah sebuah trigger yang akan menyimpan history Lokasi alamat pegawai, jika sebuah alamat berubah, maka **alamat lama** harus disimpan ke **tabel history alamat**.



# Buat tabel history

```
CREATE TABLE history_alamat (  
    Waktu DATETIME,  
    NIP CHAR(5),  
    Alamat VARCHAR (35),  
    Kota VARCHAR (15),  
    User VARCHAR(20)  
);
```



# Contoh Trigger

```
CREATE TRIGGER trig_update_alamat  
AFTER UPDATE ON pegawai  
FOR EACH ROW  
    INSERT INTO history_alamat  
    VALUES (now(), OLD.NIP, OLD.Alamat, OLD.Kota, USER() );
```





# Contoh update data

1. Ubahlah alamat dari pegawai yang NIP nya 12349 menjadi di Jl. Dipatiukur, dan kota Bandung
2. Ubahlah alamat dari pegawai yang NIP nya 12348 menjadi di Jl. Karangwaru, dan kota Magelang
3. Setelah melakukan update, tampilkan data yang ada pada tabel history\_alamat



# Contoh update data

```
UPDATE pegawai SET Alamat='Jl. Dipatiukur',  
    Kota='Bandung'  
WHERE NIP=12349;
```

```
UPDATE pegawai SET Alamat='Jl. Karangwaru',  
    Kota='Magelang'  
WHERE NIP=12348;
```

```
SELECT * FROM history_alamat;
```



# Contoh 2

- Buatlah suatu trigger yang mencegah perubahan pada primary key tabel penerbit (Kode\_Penerbit). Jika ada perubahan, maka Kode\_Penerbit di tabel buku pun tidak boleh berubah.
- Hal ini dapat dilakukan dengan menetapkan nilai Kode\_Penerbit yang baru (NEW. Kode\_Penerbit) dengan nilai yang lama (OLD. Kode\_Penerbit)



## Contoh 2

```
CREATE TRIGGER trig_update_kodepenerbit  
BEFORE UPDATE ON buku  
FOR EACH ROW  
SET NEW. Kode_Penerbit =OLD. Kode_Penerbit;
```



## Contoh 2

```
UPDATE buku SET Kode_Penerbit='AF'  
WHERE Kode_Penerbit='AO';
```

```
SELECT * FROM buku;
```



# Menghapus Trigger

- `DROP TRIGGER nama_trigger;`
- `DROP TRIGGER trig_update_kodepenerbit;`



# Latihan

1. Buat tabel **history\_lokasi**:
  - Waktu DATETIME,
  - Kode\_Penerbit VARCHAR, 2 karakter
  - Lokasi VARCHAR, 20 karakter
  - User VARCHAR, 20 karakter
2. Buatlah sebuah trigger yang akan menyimpan history Lokasi alamat penerbit, jika sebuah alamat berubah, maka **alamat lama** harus disimpan ke **tabel history lokasi**.



# Latihan

3. Ubahlah data penerbit:
  - a. Lokasi penerbit Gramedia menjadi di Bandung,
  - b. Lokasi penerbit Banyumedia menjadi di Solo





# Latihan

```
2. CREATE TRIGGER trig_update_penerbit  
AFTER UPDATE ON penerbit  
FOR EACH ROW  
INSERT INTO history_lokasi  
VALUES (now(), OLD.Kode_Penerbit, OLD.Lokasi, USER() );
```



3. a. UPDATE penerbit SET Lokasi='Bandung'  
WHERE Nama\_Penerbit='Gamedia';
  
- b. UPDATE penerbit SET Lokasi='Solo'  
WHERE Nama\_Penerbit='Banyumedia';



# KUIS

