



Basis Data 2

Sirojul Munir S.Si,M.Kom

rojulman@nurulfikri.ac.id |  rojulman
dev.xbata.com

Triggers

Apa itu Triggers !

- Object pada database yang berupa aksi untuk menjalankan fungsi/procedure ketika terjadi proses manipulasi data (DML) pada baris data table.
- Trigger dapat dikatakan sebagai store procedure spesial yang dijalankan pada suatu event (kejadian).

Event Triggers

- Event atau Kejadian dimana sebuah fungsi/procedure dijalankan ketika terjadi sebuah proses DML pada table
- Event:
 - AFTER: INSERT, UPDATE, DELETE
 - BEFORE: INSERT, UPDATE,DELETE

Database Trigger - SQL

- Belum secara spesifik ada pada SQL-92, tetap sudah di standarisasi pada SQL3 (SQL1999)
- Tersedia di hampir semua vendor database: Oracle, IBM DB2, MS SQL server, PostgreSQL, MySQL.

Tidak tersedia di versi desktop (Oracle Lite), Ms.Access dan SQLite

- Beberapa vendor DBMS menyediakan native extensions to SQL untuk menjalankan Triggers

Contoh: e.g. PL/SQL in Oracle, Transact SQL in MS SQL Server, PLPGSQL PostgreSQL

- Beberapa DBMS juga memperbolehkan penggunaan bahasa pemrograman lain selain SQL

Contoh: C/C++ in Poet, Java in Oracle, C#/VB in SQL Server

Sintaks – Trigger

CREATE

[DEFINER = { user | CURRENT_USER }]

TRIGGER trigger_name trigger_time trigger_event

ON tbl_name FOR EACH ROW trigger_body

Sintaks - Trigger

- **trigger_name** : nama trigger.
- **trigger_time** : kapan kita mengeksekusi trigger, apakah sebelum atau sesudah perubahan pada row data table. Jadi pilihannya adalah AFTER atau BEFORE.
- **trigger_event** : merupakan event atau peristiwa yang menyebabkan trigger dilakukan. Pilihan event tersebut adalah INSERT, UPDATE, DELETE.
- **tbl_name** : nama table.
- **trigger_body** : statement-statement perintah SQL yang akan dilakukan. Jika perintahnya lebih dari satu maka gunakan dalam blok statement BEGIN ... END.

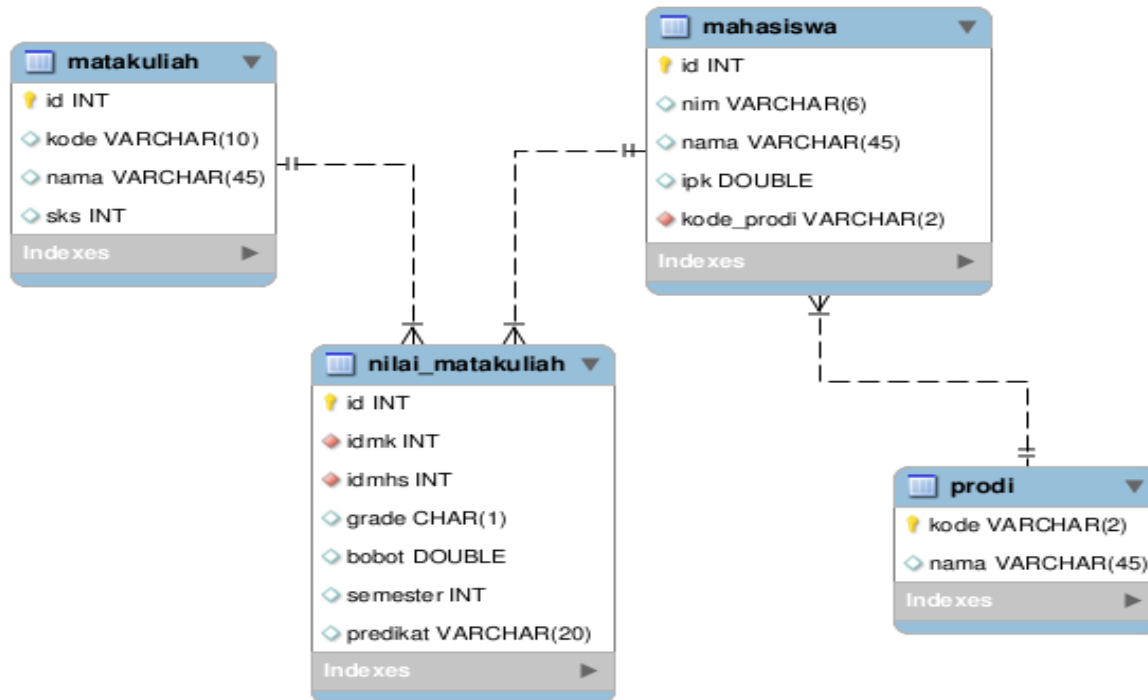
Trigger body dapat juga berupa fungsi yang telah didefinisikan

- Jika DEFINER dispesifikasikan maka kita memutuskan trigger tersebut dijalankan hanya oleh user tertentu (dalam format penulisan user@host). Jika tidak dispesifikasikan, maka user yang melakukan perubahan (CURRENT_USER) adalah pilihan default

NEW | OLD data

- Pada trigger dengan event AFTER/BEFORE INSERT , data baru ditandai dengan keyword NEW
- Pada trigger dengan event AFTER/BEFORE UPDATE/DELETE, data yang diubah/hapus ditandai dengan keyword OLD

Contoh Kasus : Log Table



Proses INSERT terhadap table mahasiswa akan direkam dalam log_table

Table : Log Table

1. Buat table **log_table**

yang akan mencatat aktifitas dari table (DML)

```
create table log_table (  
  id serial primary key,  
  nama_table varchar(30),  
  aksi varchar(30),  
  isi text,  
  waktu timestamp default now()  
);
```

Fungsi : tambah_mahasiswa_log()

```
CREATE OR REPLACE FUNCTION tambah_mahasiswa_log()
RETURNS TRIGGER AS
$$
BEGIN
    INSERT INTO log_table(nama_table,aksi,isi)
        VALUES ('mahasiswa','INSERT',NEW.id || '-' || NEW.nama);
    RETURN NEW;
END;
$$
LANGUAGE plpgsql;
~
```

Trigger: trig_log_input_mahasiswa

```
CREATE TRIGGER trig_log_input_mahasiswa  
AFTER INSERT ON mahasiswa  
FOR EACH ROW  
EXECUTE PROCEDURE tambah_mahasiswa_log();
```

Tes Trigger

dbkampus=# insert into mahasiswa (id,nama)
values (default,'Anita Utama');

dbkampus=# select * from log_table;

id	nama_table	aksi	isi	waktu
----	------------	------	-----	-------

1	mahasiswa	INSERT	1-Budiman	2014-12-11 12:41:22.498722
---	-----------	--------	-----------	----------------------------

2	mahasiswa	INSERT	2-Anita Utama	2014-12-11 12:46:08.23914
---	-----------	--------	---------------	---------------------------

Trigger pada Aplikasi SIM

- **Auditing Table Operations**

Setiap kali sebuah table diakses, informasi akses dicatat

- **Tracking Record Value Changes**

Memasitikan setiap nilai data/record berubah maka data sebelum perubahan dicatat/direkam juga

- **Protecting Database Referential Integrity:**

Mencatat jika terjadi perubahan foreign key agar referential integrity tetap terjaga/terjamin

- **Maintenance of Semantic Integrity :** salah satu cara untuk memastikan aturan proses bisnis terjadi dan sekaligus melindungi integritas data

contohnya: the factory is closed, all employees should become unemployed

- **Storing Derived Data**

contohnya: the number of items in the trolley should correspond to the current session selection

- **Security Access Control**

contohnya: cek user privileges ketika mengakses informasi data sensitif

Langkah2 Triger PostgreSQL

- 1) Analisa dimana dan kapan event trigger dijalankan ? Table apa, AFTER/BEFORE
- 2) Buat fungsi dengan nilai balik TRIGGERS
- 3) Panggil fungsi dalam TRIGGER yang didefinisikan

Aturan Trigger : Good Practice

Rule 1: Do not *change data* in the primary key, foreign key, or unique key columns of any table

Rule 2: Do not *update records* in the same table you read during the same transaction

Rule 3: Do not *aggregate* over the same table you are updating

Rule 4: Do not *read data* from a table which is updated during the same transaction

Rule 5: Do not use SQL DCL (Data Control Language) statements in triggers