

## TRIGGER

### Tujuan :

- Mampu menggunakan dan memahami perintah TRIGGER

### Materi :

- TRIGGER

### Petunjuk :

- Ikuti langkah-langkah pengerjaan.
- Seluruh perintah atau query dalam pengerjaan langkah-langkah beserta hasil dari eksekusi perintah atau query di print screen dan di copy paster pada word sesuai dengan langkah langkah pengerjaan.
- Document word diupload di schoology dengan nama document 'modul\_nim'

### Pengantar :

Trigger dalam database adalah kode prosedural yang secara otomatis dijalankan untuk menanggapi perubahan tertentu pada table tertentu atau tampilan dalam database. Trigger dapat didefinisikan untuk menjalankan perintah sebelum atau setelah eksekusi DML (Data Manipulation Language) seperti INSERT, UPDATE, dan DELETE. Trigger banyak digunakan untuk menjaga integritas informasi pada database.

Sintak umum.

```
CREATE  
[DEFINER = { user | CURRENT_USER }]  
TRIGGER trigger_name trigger_time trigger_event  
ON tbl_name FOR EACH ROW trigger_stmt
```

### Keterangan :

*\_[DEFINER = { user | CURRENT\_USER }]*: Definisi *user* yang sedang aktif, sifatnya opsional.

*\_trigger\_name*: Nama *trigger*.

*\_trigger\_time*: waktu menjalankan *trigger*. Ini dapat berupa BEFORE atau AFTER.

*\_BEFORE*: Membuat *trigger* diaktifkan sebelum dihubungkan dengan suatu operasi.

*\_AFTER*: Membuat *trigger* diaktifkan setelah dihubungkan dengan suatu operasi.

*\_trigger\_event*: berupa kejadian yang akan dijalankan *trigger*.

*\_trigger\_event* dapat berupa salah satu dari berikut:

\_INSERT : *trigger* diaktifkan ketika sebuah *record* baru disisipkan ke dalam tabel.

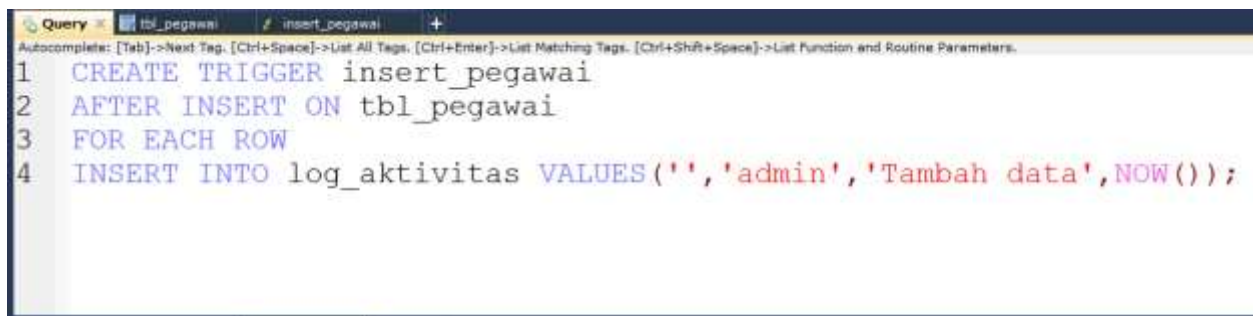
### Langkah-langkah :

1. Gunakan database sebelumnya yang terdapat `tbl_pegawai`, `tbl_penggajian`, `tbl_golongan` dan `tbl_tempatkerja`.
2. Tambahkan table di bawah sesuai dengan tipe data, panjang data.

LOG_AKTIVITAS		TBL_CASBON	
NAMA FIELD	TIPE,PJG	NAMA FIELD	TIPE,PJG
Int_id	int,4	Int_id	Int,4
Str_user	Varchar,50	str_nm_casbon	Varchar,50
Str_action	text	int_jml_casbon	Int,11
Tgl_action	date	tgl_casbon	date
		str_status	Varchar,10
		str_nip	Varchar,4

3. Pada `tbl_penggajian` tambahkan kolom dengan nama `int_tot_gajim` type data `int,11`.
4. Membuat Triger sederhana.

### Membuat trigger ketika melakukan insert pada table pegawai



```
Query | tbl_pegawai | insert_pegawai | +
Autocomplete: [Tab]->Next Tag. [Ctrl+Space]->List All Tags. [Ctrl+Enter]->List Matching Tags. [Ctrl+Shift+Space]->List Function and Routine Parameters.
1 CREATE TRIGGER insert_pegawai
2 AFTER INSERT ON tbl_pegawai
3 FOR EACH ROW
4 INSERT INTO log_aktivitas VALUES('','admin','Tambah data',NOW());
```

- Jika sudah berhasil membuat trigger dengan nama `insert_pegawai`, selanjutnya tambahkan satu data pada `tbl_pegawai`.

```
Query x1 tbl_pegawai insert_pegawai +
Autocomplete: [Tab]->Next Tag, [Ctrl+Space]->List All Tags, [Ctrl+Enter]->List Matching Tags, [Ctrl+Shift+Space]->List Function and Routine Parameters.
1 INSERT INTO tbl_pegawai VALUES
2 ('0006','Made Gede','Denpasar Barat','A4','L','Hindu')
```

- Selanjutnya select pada tabel log\_aktivitas, maka data juga akan muncul.

int_id	str_user	str_action	tgl_action
1	admin	Tambah data	2019-06-17
(Auto): (NULL)	(NULL)	OK	(NULL)

- Dari contoh diatas dapat dilihat bahwa ketika satu *record* pada tabel **tbl\_pegawai** disisipkan (*insert*), maka secara otomatis tabel **log\_aktivitas** akan disisipkan satu *record*, yaitu kejadian ‘Tambah data’ dan waktu saat *record* pada tabel **tbl\_pegawai** disisipkan.

## Membuat trigger ketika melakukan update pada tbl\_pegawai

```
Query x1 tbl_pegawai cari_nilai update_pegawai ifelse upd_pegawai upd_pegawai +
Autocomplete: [Tab]->Next Tag, [Ctrl+Space]->List All Tags, [Ctrl+Enter]->List Matching Tags, [Ctrl+Shift+Space]->List Function and Routine Parameters.
1 DELIMITER //
2 CREATE TRIGGER upt_pegawai AFTER UPDATE ON tbl_pegawai
3 FOR EACH ROW
4 BEGIN
5 SET @aktivitas = CONCAT(new.str_nip,' ubah nama dari ',new.str_nm_pegawai,'
6 menjadi ',old.str_nm_pegawai);
7 INSERT INTO log_aktivitas VALUES ('','admin',@aktivitas,NOW());
8 END;
9 //
10 DELIMITER;
```

- Jika sudah berhasil membuat trigger dengan nama upt\_pegawai, selanjutnya update nama salah satu pegawai pada tbl\_pegawai.

- Catatan :
  - DELIMITER, berfungsi untuk membari tahu kepada MySQL, delimiter yang digunakan, secara default biasanya menggunakan semicolon (;) tapi diatas kita menggunakan //
  - “new” yang di ikuti nama kolom contoh “new.str\_nm\_pegawai” mengambil data str\_nm\_pegawai yang baru di inputkan.
  - “old” yang di ikuti nama kolom contoh “old.str\_nm\_pegawai” mengambil data str\_nm\_pegawai yang lama atau yang di ubah.

```

7
8
9
10 UPDATE tbl_pegawai SET str_nm_pegawai='Moh. Riyan'
11 WHERE str_nip='0005';
12
13
14
15

```

- Selanjutnya select pada tabel log\_aktivitas, maka data juga akan muncul.

int_id	str_user	str_action	tgl_action
1	admin	Tambah data	2019-06-17
25	admin	0001 ubah nama dari Moh. Riyan menjadi Baharudin	2019-06-17
*(Auto)	(NULL)	(NULL)	OK (NULL)

Database: db\_sisiam Table: log\_aktivitas

- Dari contoh diatas dapat dilihat bahwa ketika satu *record* pada tabel **tbl\_pegawai** diperbarui (*update*), maka secara otomatis tabel **log\_aktivitas** akan disisipkan

satu *record*, yaitu kejadian ‘ubah data’ dan waktu saat *record* pada tabel **tbl\_pegawai** disisipkan.

## Membuat trigger ketika melakukan delete pada tbl\_pegawai

```
Query x tbl_pegawai cari_nilai update_pegawai ifelse upd_pegawai upd_pegawai +
Autocomplete: [Tab]->Next Tag. [Ctrl+Space]->List All Tags. [Ctrl+Enter]->List Matching Tags. [Ctrl+Shift+Space]->List Function and Routine Parameters.
15
16 DELIMITER //
17 CREATE TRIGGER del_pegawai AFTER DELETE ON tbl_pegawai
18 FOR EACH ROW
19 BEGIN
20 SET @aktivitas = CONCAT('menghapus NIP ',old.str_nip);
21 INSERT INTO log_aktivitas VALUES ('','admin',@aktivitas,NOW());
22 END;
23 //
24 DELIMITER;
25
26
27
```

- Jika sudah berhasil membuat trigger dengan nama del\_pegawai, selanjutnya delete salah satu pegawai pada tbl\_pegawai.

```
Query x tbl_pegawai cari_nilai update_pegawai ifelse upd_pegawai upd_pegawai +
Autocomplete: [Tab]->Next Tag. [Ctrl+Space]->List All Tags. [Ctrl+Enter]->List Matching Tags. [Ctrl+Shift+Space]->List Function and Routine Parameters.
25
26
27 DELETE FROM tbl_pegawai WHERE str_nip='0001';
28
```

- Selanjutnya select pada tabel log\_aktivitas, maka data juga akan muncul.

1 Profiler 2 Messages 3 Table Data 4 Info				
int_id	str_user	str_action	tgl_action	
1	admin	Tambah data	11B	2019-06-17
25	admin	0001 ubah nama dari Moh. Riyan menjadi Baharudin	49B	2019-06-17
26	admin	menghapus NIP 0001	18B	2019-06-17
*	(Auto)	(NULL)	OK	(NULL)

- Dari contoh diatas dapat dilihat bahwa ketika satu *record* pada tabel **tbl\_pegawai** dihapus (*delete*), maka secara otomatis tabel **log\_aktivitas** akan disisipkan satu

*record*, yaitu kejadian 'mehhapus data' dan waktu saat *record* pada tabel **tbl\_pegawai** disisipkan.

- Selain perintah insert, perintah seperti update dan delete juga dapat di gunakan pada trigger, bahkan di dalam perintah trigger dapat melakukan perhitungan dan oprasi logika.

## 5. Triger dengan perintah Update dan Delete

Membuat trigger update pada **tbl\_gaji** ketika melakukan insert pada **tbl\_cas\_bon**

```
Query x tbl_cas_bon ipk +
Autocomplete: [Tab]->Next Tag, [Ctrl+Space]->List All Tags, [Ctrl+Enter]->List Matching Tags, [Ctrl+Shift+Space]->List Function and Routine Parameters.
29
30 DELIMITER //
31 CREATE TRIGGER insert_casbon AFTER INSERT ON tbl_cas_bon
32 FOR EACH ROW
33 BEGIN
34 SET @gaji = (SELECT int_tot_gaji FROM tbl_gaji
35 WHERE MONTH(tgl_gaji)=MONTH(new.tgl_casbon));
36 SET @total_gaji = @gaji - new.int_jml_casbon;
37 UPDATE tbl_gaji SET int_tot_gaji = @total_gaji WHERE str_nip = new.str_nip;
38 END;
39 //
40 DELIMITER;
41
```

- Jika sudah berhasil membuat trigger tersebut, selanjutnya insert data pada tabel **cas\_bon**.

```
Query x tbl_cas_bon ipk +
Autocomplete: [Tab]->Next Tag, [Ctrl+Space]->List All Tags, [Ctrl+Enter]->List Matching Tags, [Ctrl+Shift+Space]->List Function and Routine Parameters.
41
42
43
44 INSERT INTO tbl_cas_bon VALUES ('','0002','pinjam',500000,NOW(),'Belum Lunas')
45
46
47
48
49
50
51
52
53
```

- Selanjutnya select pada tabel **tbl\_cas\_bon** maka data juga akan muncul



1 Profiler 2 Messages 3 Table Data 4 Info					
<input type="checkbox"/>	str_nip	tgl_ambil	tgl_gaji	str_instansi	int_tot_gaji
<input type="checkbox"/>	0001	2018-12-01	2018-12-03	K1	2500000
<input type="checkbox"/>	0002	2018-12-01	2019-06-17	K1	2000000
<input type="checkbox"/>	0003	2019-01-01	2019-01-03	K2	2500000
<input type="checkbox"/>	0004	2019-01-01	2019-01-01	K3	2500000
<input type="checkbox"/>	0006	2019-01-01	2019-01-01	K3	2500000
<input type="checkbox"/>	0007	2019-01-01	2019-01-01	K3	2500000
*	(NULL)	(NULL)	(NULL)	(NULL)	(NULL)

- Dari contoh diatas dapat dilihat bahwa ketika satu *record* pada tabel **tbl\_cas\_bon** diinsert, maka secara otomatis *record* pada kolom int\_tot\_gaji pda tabel tbl\_gaji akan dirubah.

## **Latihan**

Petunjuk :

1. Tuliskan query atau perintah pada sqlyog dan eksekusi query tersebut.
2. Print screen hasil query dan copy paste pada word sesuai dengan no pertanyaan.
3. Pengumpulan latihan menggunakan document word dengan judul 'latihan\_nim'

Soal :

1. Buatlah sebuah trigger dengan kondisi, ketika melakukan delete data pada tabel pegawai, maka pada tabel penggajian dan tabel cas bon pegawai tersebut ikut terhapus.
2. Buatlah sebuah trigger dengan kondisi, ketika str\_status pada tabel tbl\_cas\_bon di update menjadi "Lunas" maka data pada tbl\_penggajian sesuai dengan nip tersebut dan sesuai bulan casbon yang di update, data yang terdapat pada tbl\_penggajian terhapus.