

**PRAKTIKUM DATABASE**

# **TRIGGER**

**UIN MAULANA MALIK IBRAHIM MALANG**

# Tujuan

- Mahasiswa akan mempelajari cara menggunakan create trigger di MariaDB
- Mahasiswa dapat membuat trigger baru di MariaDB.

# Pengertian

- TRIGGER adalah kumpulan kode SQL yang berjalan secara otomatis untuk mengeksekusi perintah INSERT, UPDATE, DELETE.
- Biasanya TRIGGER akan dijalankan sebelum atau sesudah proses INSERT, UPDATE, DELETE (Perintah DML)

# Sintaks Trigger

```
DELIMITER $$  
CREATE TRIGGER nama_trigger  
{BEFORE | AFTER} {INSERT | UPDATE | DELETE }  
    ON nama_table  
    FOR EACH ROW  
BEGIN  
    KODE SQL  
END$$  
DELIMITER ;
```

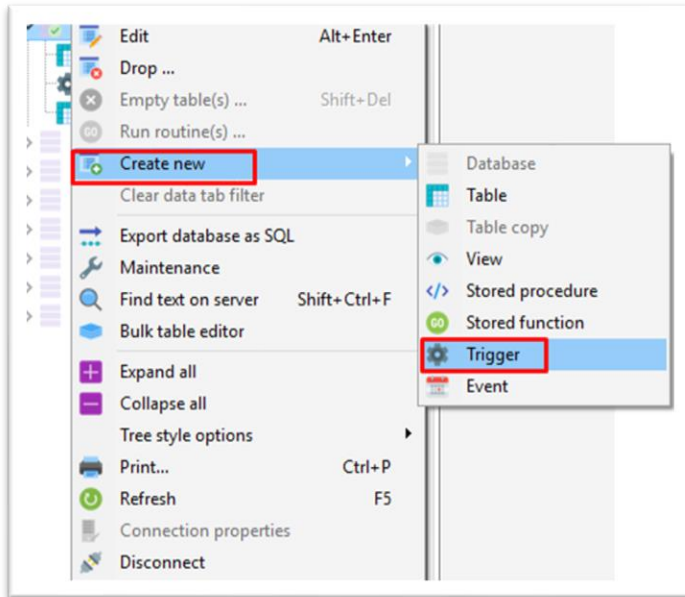
# Detail Waktu TRIGGER

No.	Waktu	Keterangan
1.	BEFORE INSERT	TRIGGER dijalankan sebelum record dimasukkan ke database
2.	AFTER INSERT	TRIGGER dijalankan sesudah record dimasukkan ke database
3.	BEFORE UPDATE	TRIGGER dijalankan sebelum record dirubah di database
4.	AFTER UPDATE	TRIGGER dijalankan sesudah record dirubah database
5.	BEFORE DELETE	TRIGGER dijalankan sebelum record dihapus di database
6.	AFTER DELETE	TRIGGER dijalankan sesudah record dihapus di database

# DELIMITER

- Delimiter adalah pembatas
- Biasanya berupa semicolon (;) atau \$\$ atau ||
- Biasanya digunakan untuk **stored program** (*procedure, function, trigger* dan *event*)
- Simbol delimiter bisa menggunakan simbol/karakter/huruf atau gabungan dari ketiganya.

# Cara membuat Trigger di HeidiSQL



The screenshot shows the HeidiSQL Trigger configuration window. The 'Options' tab is selected, and the 'CREATE code' sub-tab is active. The 'Name' field is highlighted with a red rectangle and contains the text 'Enter trigger name'. The 'Definer' field contains 'Current user (root@localhost)'. The 'On table' field contains 'jurusan'. The 'Event' field is set to 'BEFORE' and 'INSERT'. The 'Trigger statement' field contains the following SQL code:

```
1 BEGIN
2
3 END
```

The 'Trigger statement' field is highlighted with a red rectangle. At the bottom of the window, there are buttons for 'Help', 'Discard', and 'Save'.

# Praktikum delimiter

---

```
1 DELIMITER agung
2 CREATE TRIGGER update_dtbrg
3     BEFORE UPDATE
4     ON pembeli
5     FOR EACH ROW
6 BEGIN
7 UPDATE barang
8 SET barang.stokbrg=barang.stokbrg-pembeli.jumlah
9 WHERE kodebrg=pembeli.kodebrg;
10 END
11
12 agung
13 DELIMITER ;
```

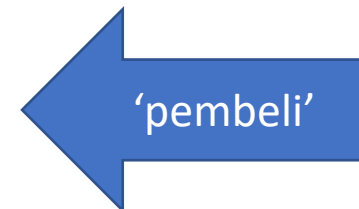


# Data Tabel (Praktikum)

kodebrg		namabrg	stokbrg	satuan
1		Kecap	100	Botol
2		Mie Goreng Sedap	100	Kardus
3		Sabun Cuci Attack	40	Kardus



kodebeli		kodebrg		nama	jumlah
1		3		cus1	10



Options `</>` CREATE code

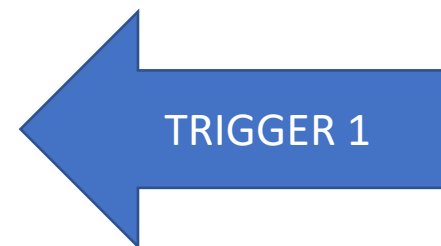
Name:  Definer:

On table:  ✓

Event:

Trigger statement: (e.g. "SET NEW.columnA = TRIM(OLD.columnA)")

```
1 BEGIN
2 UPDATE barang
3 SET barang.stokbrg=barang.stokbrg-NEW.jumlah
4 WHERE kodebrg=NEW.kodebrg;
5 END
```



Options `</>` CREATE code

Name:  Definer:

On table:  ✓

Event:

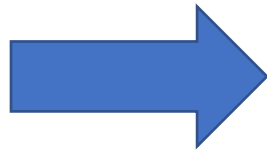
Trigger statement: (e.g. "SET NEW.columnA = TRIM(OLD.columnA)")

```
1 BEGIN
2 UPDATE barang
3 SET barang.stokbrg=barang.stokbrg-pembeli.jumlah
4 WHERE kodebrg=pembeli.kodebrg;
5 END
```

# Praktikum

jejak.jurusan: 2 rows total (approximately)

idjurusan		nmjurusan
1		Teknik Informatika
2		Kimia



jejak.log: 2 rows total (approximately)

idlog		idjurusan		status
1		1		1
2		2		1

Options CREATE code

Name:

jurusan\_after\_insert

Definer:

root@localhost

On table:

jurusan

Event:

BEFORE

INSERT

Trigger statement: (e.g. "SET NEW.columnA = TRIM(OLD.columnA)")

```
1 BEGIN
2 INSERT INTO log (idjurusan, status) VALUES (NEW.idjurusan, '1');
3 END
```

# Tugas Praktikum

Buatlah suatu trigger yang dapat mengisi tabel **absensi** secara otomatis setelah tabel **karyawan** diisi, struktur data tabelnya sebagai berikut:

jejak.jurusan: 2 rows total (approximately)

idjurusan	nmjurusan
1	Teknik Informatika
2	Kimia

jejak.karyawan: 1 rows total (approximately)

idkaryawan	idjurusan	nama	alamat
1	1	Agung Teguh WA	Batu

jejak.absensi: 1 rows total (approximately)

idabsensi	idkaryawan	waktu
1	1	2022-11-08 04:40:12