

# **LAPORAN PRAKTIKUM BASIS DATA**

## **TRIGGER**



**DISUSUN OLEH :**

Oktario Mufti Yudha

2320506044

**PROGRAM STUDI S1 TEKNOLOGI INFORMASI**

**JURUSAN TEKNIK ELEKTRO FAKULTAS**

**TEKNIK UNIVERSITAS TIDAR**

**2024**

**LAPORAN**  
**PRAKTIKUM BASIS DATA**



Diisi Mahasiswa Praktikan								
Nama Praktikan	Oktario Mufti Yudha							
NPM	2320506044							
Rombel	4							
Judul Praktikum	TRIGGER							
Tanggal Praktikum	30 Mei 2024							
Diisi Asisten Praktikum								
Tanggal Pengumpulan								
Catatan								

PENGESAHAN		NILAI
Diperiksa oleh :	Disahkan oleh :	
Asisten Praktikum	Dosen Pengampu	
	Imam Adi Nata, S.Kom., M.Kom.	

**PROGRAM STUDI S1 TEKNOLOGI INFORMASI**  
**JURUSAN TEKNIK ELEKTRO FAKULTAS**  
**TEKNIK UNIVERSITAS TIDAR**  
**2024**

## **I. Tujuan Praktikum**

1. Mahasiswa dapat memahami konsep Trigger
2. Mahasiswa mampu memahami cara kerja Trigger
3. Mahasiswa mampu menggunakan Trigger

## **II. Dasar Teori**

Trigger dapat didefinisikan sebagai himpunan kode (prosedural) yang dieksekusi secara otomatis sebagai respon atas suatu kejadian berkaitan dengan tabel basis data. Kejadian (event) yang dapat membangkitkan trigger umumnya berupa pernyataan INSERT, UPDATE, dan DELETE. Berdasarkan ruang lingkupnya, trigger diklasifikasikan menjadi dua jenis: row trigger dan statement trigger. Trigger baris (row) mendefinisikan aksi untuk setiap baris tabel; trigger pernyataan hanya berlaku untuk setiap pernyataan INSERT, UPDATE, atau DELETE. Dari sisi perilaku (behavior) eksekusi, trigger dapat dibedakan menjadi beberapa jenis; namun umumnya ada dua jenis: trigger BEFORE dan AFTER. Sesuai penamaannya, jenis-jenis ini merepresentasikan waktu eksekusi trigger—misalnya sebelum atau sesudah pernyataan-pernyataan yang berkorespondensi.

## **III. Dasar Teori**

### **A. Alat dan bahan**

Alat :

1. Laptop

Bahan :

1. Operating System Linux (Ubuntu)
2. Mysql
3. phpMyAdmin
4. File Materi Praktikum

### **B. Langkah kerja**

1. Menyalakan laptop
2. Membuka modul praktikum
3. Mengerjakan tugas yang ada pada modul praktikum

## **IV. Hasil dan Analisis**

### **A. Praktikum**

Membuat tabel barang dan tabel pembelian pada database baru

+ Options

				id	nama	stok
<input type="checkbox"/>	Edit	Copy	Delete	A11	Keyboard	15
<input type="checkbox"/>	Edit	Copy	Delete	A12	DVD R-W	10
<input type="checkbox"/>	Edit	Copy	Delete	A13	Modem	6

Gambar 4.1 Tabel barang

+ Options

				id_pem	id_brg	jml_beli
<input type="checkbox"/>	Edit	Copy	Delete	1	A10	5

Gambar 4.2 Tabel pembelian

Masuk ke menu Trigger kemudian pilih Add Trigger pada bagian New dan diisi data seperti berikut:

Edit

Details

Trigger name: incrementStok2

Table: barang

Time: BEFORE

Event: INSERT

Definition

```
1 SET NEW.stok = NEW.stok + 1
```

Go Close

Gambar 4.3 Mengisi Trigger

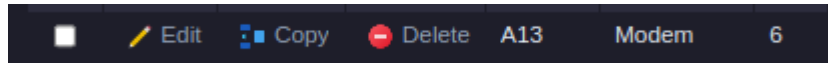
Tuliskan sintak seperti berikut pada SQL

Run SQL query/queries on database toko2: ?

```
1 INSERT INTO barang VALUES('A13', 'Modem', 5);
```

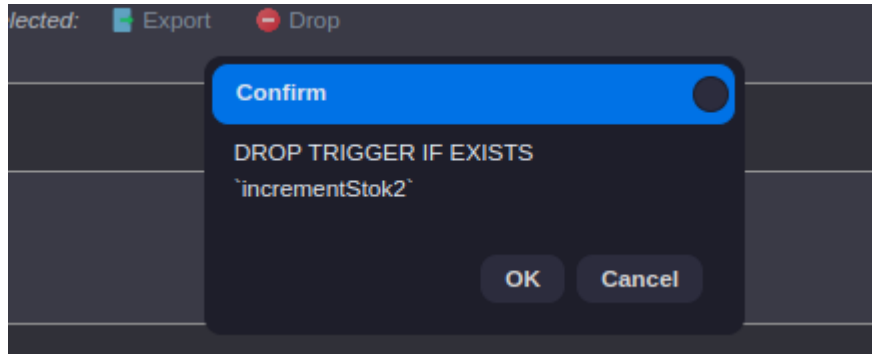
Gambar 4.4 Menulis sintak

Lalu akan menampilkan data seperti berikut:



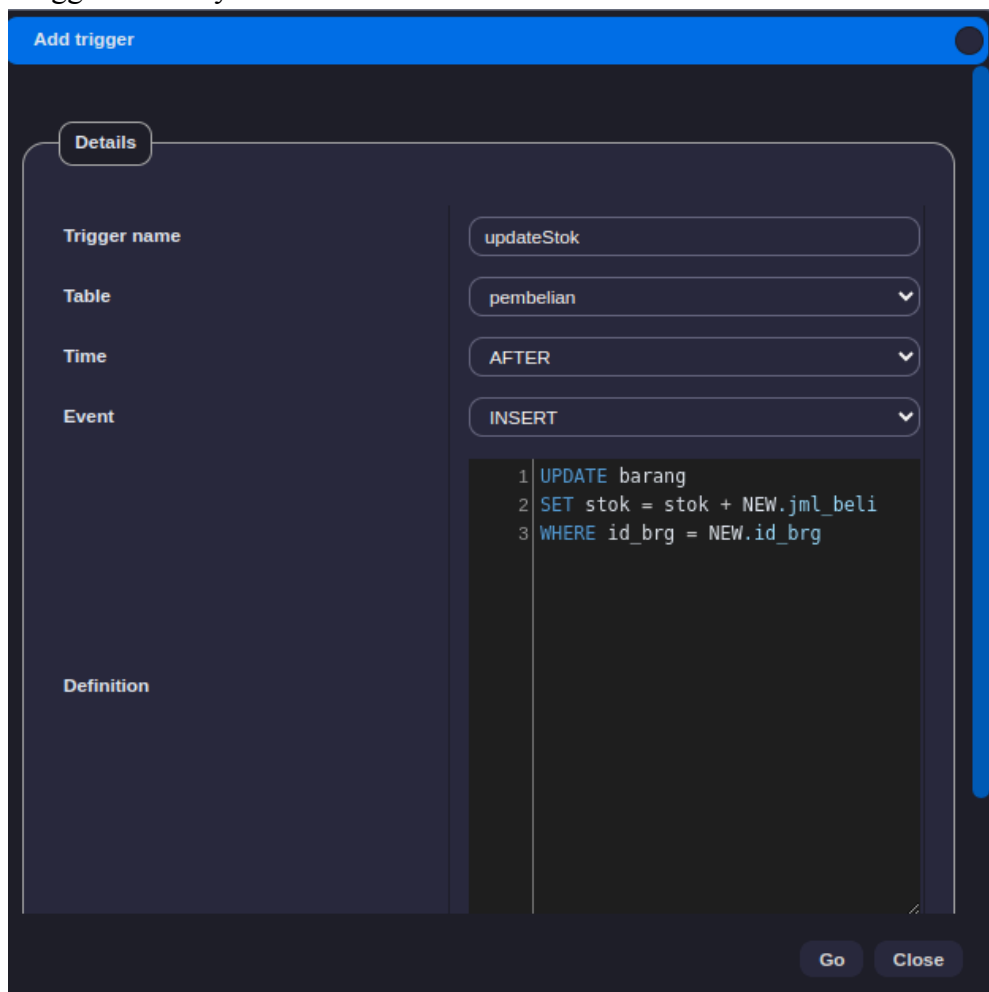
Gambar 4.5 menampilkan data

Sebagaimana objek-objek database lainnya, kita menghapus trigger dengan menggunakan perintah DROP dengan ketentuan DROP TRIGGER nama\_trigger. Atau bisa langsung ke menu trigger dan menghapusnya dengan cara berikut:



Gambar 4.5 menghapus trigger

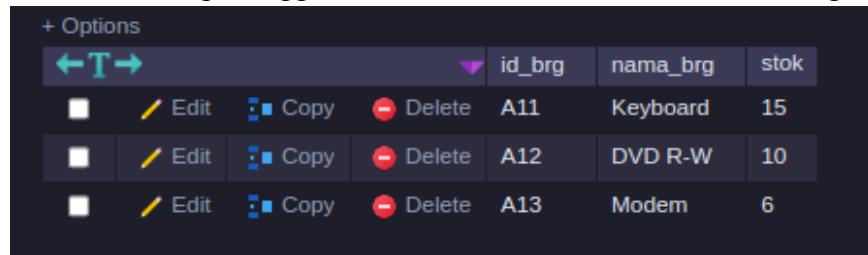
Untuk merujuk ke kolom-kolom tabel yang diasosiasikan dengan trigger, kita menggunakan keyword OLD dan NEW. Keyword OLD mengacu pada nilai lama, sedangkan NEW merepresentasikan nilai baru. Di trigger INSERT, kita hanya dapat menggunakan keyword NEW karena tidak ada data lama.



Gambar 4.5 old dan new

Pada contoh di atas, penambahan data pembelian akan mengakibatkan nilai stok barang berubah menyesuaikan banyaknya nilai jumlah pembelian.

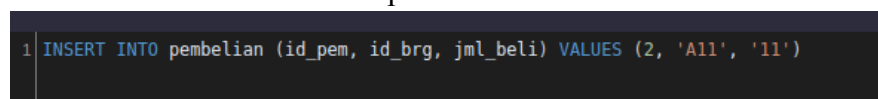
Cek tabel barang menggunakan sintaks `SELECT * FROM barang;`



				id_brg	nama_brg	stok
<input type="checkbox"/>	Edit	Copy	Delete	A11	Keyboard	15
<input type="checkbox"/>	Edit	Copy	Delete	A12	DVD R-W	10
<input type="checkbox"/>	Edit	Copy	Delete	A13	Modem	6

Gambar 4.5 cek tabel barang

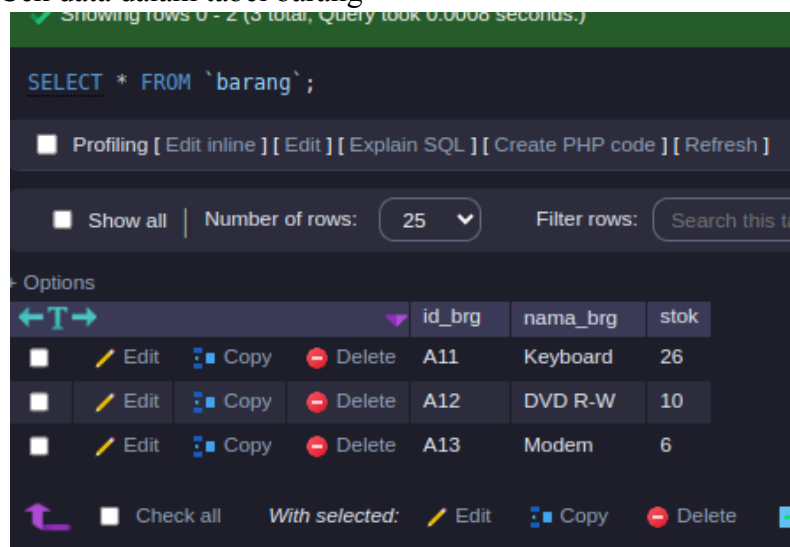
Masukkan data kedalam tabel pembelian



```
1 INSERT INTO pembelian (id_pem, id_brg, jml_beli) VALUES (2, 'A11', '11')
```

Gambar 4.5 memasukan data

Cek data dalam tabel barang



Showing rows 0 - 2 (3 total, Query took 0.0008 seconds.)

`SELECT * FROM `barang`;`

☐ Profiling [ Edit inline ] [ Edit ] [ Explain SQL ] [ Create PHP code ] [ Refresh ]

☐ Show all | Number of rows: 25 | Filter rows: Search this table

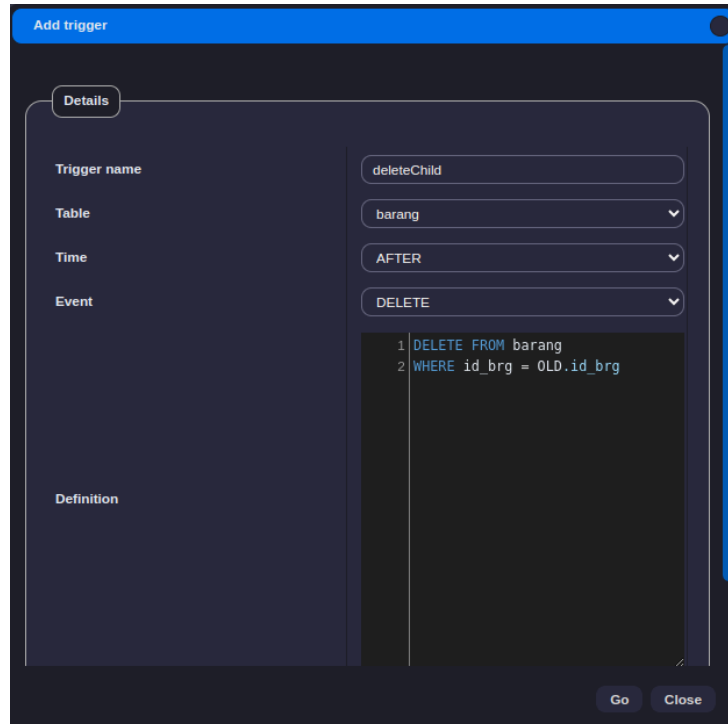
				id_brg	nama_brg	stok
<input type="checkbox"/>	Edit	Copy	Delete	A11	Keyboard	26
<input type="checkbox"/>	Edit	Copy	Delete	A12	DVD R-W	10
<input type="checkbox"/>	Edit	Copy	Delete	A13	Modem	6

☐ Check all | With selected: Edit Copy Delete

Gambar 4.5 cek data kembali

Ketika menggunakan trigger DELETE, hanya keyword OLD yang dapat digunakan. Jika ingin membuat trigger yang menghapus semua data pembelian saat data barang yang terkait diidentifikasi melalui primary key dan foreign key dihapus, bisa mendefinisikan trigger tersebut dengan cara berikut:

## Membuat trigger baru



Trigger name: deleteChild

Table: barang

Time: AFTER

Event: DELETE

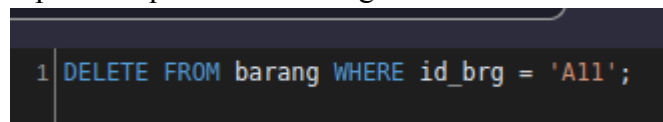
Definition:

```
1 DELETE FROM barang
2 WHERE id_brg = OLD.id_brg
```

Go Close

Gambar 4.5 membuat trigger

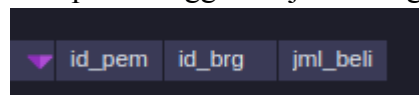
## Hapus data pada tabel barang



```
1 DELETE FROM barang WHERE id_brg = 'A11';
```

Gambar 4.5 menghapus data

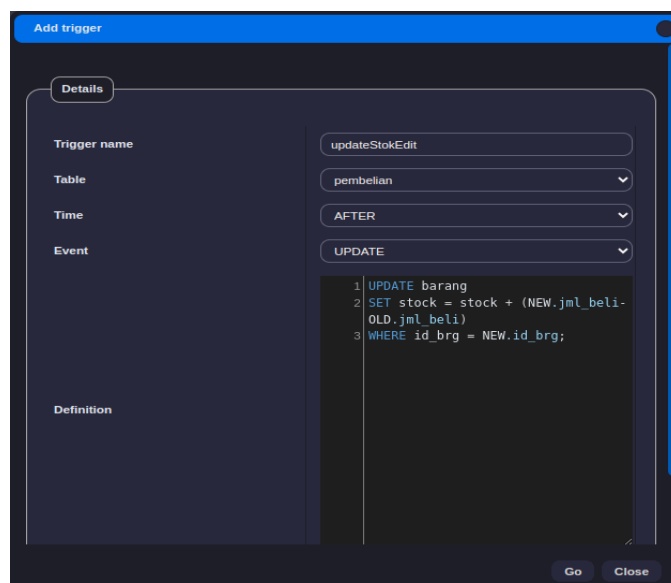
## Cek apakah trigger berjalan dengan baik dengan menampilkan tabel pembelian



	id_pem	id_brg	jml_beli
1	1	A11	5

Gambar 4.5 cek trigger

Khusus untuk operasi UPDATE, kita bisa memanfaatkan keyword NEW maupun OLD.



Trigger name: updateStokEdit

Table: pembelian

Time: AFTER

Event: UPDATE

Definition:

```
1 UPDATE barang
2 SET stock = stock + (NEW.jml beli -
  OLD.jml beli)
3 WHERE id_brg = NEW.id_brg;
```

Go Close

Gambar 4.5 update stok

Lakukan update pada tabel pembelian

```
Run SQL query/queries on table tokoz.pembelian.
1 UPDATE pembelian SET jml_beli = 25 WHERE id_pem = 3;
```

Gambar 4.5 melakukan update

Cek apakah trigger berjalan dengan baik dengan cara tampilkan pada tabel barang

SELECT \* FROM `barang`;

☐ Profiling [ Edit inline ] [ Edit ] [ Explain SQL ] [ Create PHP code ] [ Refresh ]

☐ Show all | Number of rows: 25 | Filter rows: Search this

+ Options

				id_brg	nama_brg	stok
<input type="checkbox"/>	Edit	Copy	Delete	A11	Keyboard	26
<input type="checkbox"/>	Edit	Copy	Delete	A12	DVD R-W	10
<input type="checkbox"/>	Edit	Copy	Delete	A13	Modem	6

Gambar 4.5 cek trigger

Trigger kompleks

```
Run SQL query/queries on table tokoz.barang.
1 CREATE TRIGGER auditBarang BEFORE INSERT ON barang FOR EACH ROW BEGIN IF NOT EXISTS (SELECT id_brg FROM barang WHERE id_brg = NEW.id_brg) THEN SET NEW.nama_brg = NEW.nama_barang; SET NEW.stok = NEW.stok; ELSE SET @status = CONCAT('Id', NEW.id_brg, 'sudah ada'); END IF; END;;
```

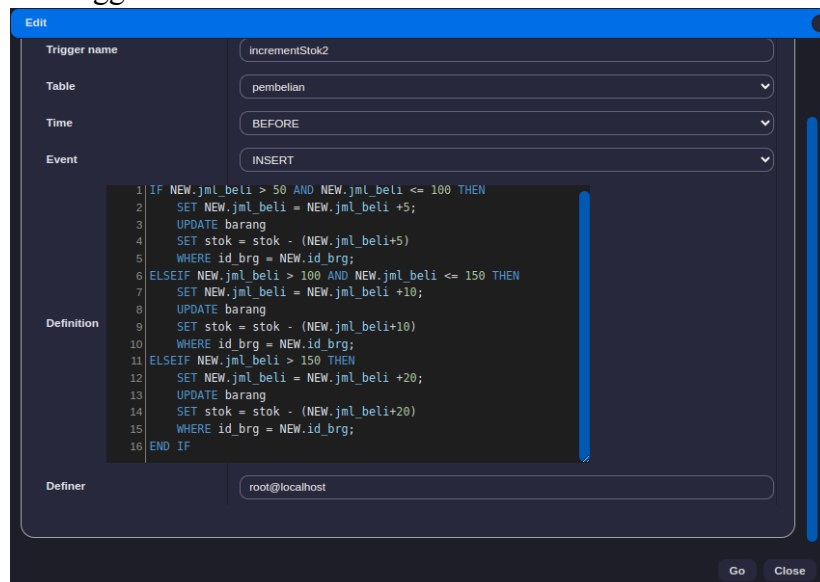
Gambar 4.5 triger kompleks



## B. Tugas

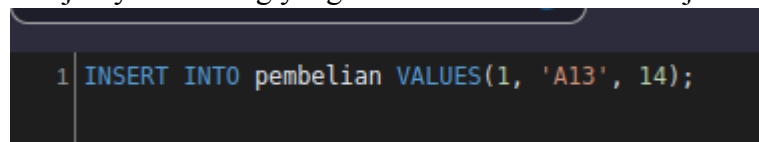
Modifikasi trigger INSERT pembelian untuk menambahkan fitur bonus di dalamnya. Jika pembelian > 50 dan < 100, maka bonus = 5; jika pembelian > 100 dan < 150, maka bonus = 10; jika pembelian > 150, maka bonus = 20. Ingat, aturan penyesuaian stok barang masih berlaku, setiap ada pembelian maka stok akan berkurang.

Buat trigger baru



Gambar 4.6 membuat trigger

Membuat perintah SQL. Pada VALUES terdapat 3 bagian, pertama adalah id pembeli, selanjutnya id barang yang dibeli dan terakhir adalah jumlah barang yang dibeli.



Gambar 4.7 Perintah SQL

## **V. Kesimpulan**

Dengan memanfaatkan trigger, mahasiswa dapat melihat bagaimana perubahan data dapat dikelola secara efisien dan otomatis sesuai dengan aturan yang telah ditetapkan. Namun, penting juga untuk memahami bahwa penggunaan trigger harus dilakukan dengan hati-hati untuk menghindari dampak negatif pada kinerja basis data. Praktikum ini memberikan pemahaman praktis tentang bagaimana merancang dan mengimplementasikan trigger dengan tepat untuk meningkatkan efisiensi dan keandalan sistem basis data.

## **VI. Referensi**

File modul praktikum