

**LAPORAN WORKSHOP BASIS DATA PENGGUNAAN
SQL TRIGGER AFTER INSERT PADA STUDI KASUS
DATABASE KOPERASI PUPUK**

TUGAS BKPM 15



oleh

**Faiq Raihan Albaihaqi
NIM E41241011
Golongan C**

**PROGRAM STUDI TEKNIK INFORMATIKA
JURUSAN TEKNOLOGI INFORMASI
POLITEKNIK NEGERI JEMBER
2024**

DAFTAR ISI

DAFTAR ISI	ii
DAFTAR GAMBAR	iii
BAB I PEMBAHASAN	1
1.1 Acara 53 dan 54	1
1.1.1 Trigger After Insert (Koperasi Pupuk)	1
1.1.2 Membuat Trigger After Insert (Koperasi Pupuk).....	1
1.1.3 Awal Tampilan Tabel (<i>Before Insert</i>)	4
1.1.4 Uji Coba Sintaks Insert	4
1.1.5 Akhir Tampilan Tabel (<i>After Insert</i>).....	5
1.1.6 Kesimpulan Trigger After Insert	5
1.2 Acara 55 dan 56	6
1.2.1 Trigger After Insert (Final Project)	6
1.2.2 Membuat Trigger After Insert (Final Project).....	6
1.1.3 Awal Tampilan Tabel (<i>Before Insert</i>)	8
1.1.4 Uji Coba Sintaks Insert	8
1.1.5 Akhir Tampilan Tabel (<i>After Insert</i>).....	9
1.1.6 Kesimpulan Trigger After Insert	9

DAFTAR GAMBAR

Gambar 1.1 Structure Database Koperasi Pupuk	1
Gambar 1.2 Query Create Trigger After Update Kurang Stok.....	2
Gambar 1.3 Query Create Trigger After Update Tambah Stok.....	3
Gambar 1.4 Tabel Penjualan Awal	4
Gambar 1.5 Query SQL Insert	4
Gambar 1.6 Tabel Penjualan Akhir	5
Gambar 1.7 Structure Database Final Project	6
Gambar 1.8 Query Create Trigger After Update Tambah Stok Produk	7
Gambar 1.9 Tabel Pembelian Awal	8
Gambar 1.10 Tabel Produk Awal	8
Gambar 1.11 Query SQL Insert	8
Gambar 1.12 Tabel Pembelian Akhir	9
Gambar 1.13 Tabel Produk Akhir	9

BAB I PEMBAHASAN

1.1 Acara 53 dan 54

Buka kembali database “Koperasi Pupuk”, buatlah trigger untuk mengurangi dan menambah stok pada tabel pupuk.

Table	Action
<input type="checkbox"/> anggota	★ Browse Structure Search Insert Empty Drop
<input type="checkbox"/> detail_beli	★ Browse Structure Search Insert Empty Drop
<input type="checkbox"/> detail_jual	★ Browse Structure Search Insert Empty Drop
<input type="checkbox"/> pembelian	★ Browse Structure Search Insert Empty Drop
<input type="checkbox"/> penjualan	★ Browse Structure Search Insert Empty Drop
<input type="checkbox"/> pupuk	★ Browse Structure Search Insert Empty Drop
<input type="checkbox"/> supplier	★ Browse Structure Search Insert Empty Drop

Gambar 1.1 Structure Database Koperasi Pupuk

1.1.1 Trigger After Insert (Koperasi Pupuk)

Trigger after insert adalah sebuah objek basis data yang secara otomatis dieksekusi setelah operasi penyisipan (*insert*) data baru ke dalam sebuah tabel telah berhasil dilakukan. Jenis Trigger ini biasa digunakan untuk berbagai tujuan seperti pencatatan log, validasi tambahan, atau pembaruan tabel terkait. Selain itu, Trigger After Insert memiliki fungsi utama seperti melakukan tindakan tambahan setelah data baru dimasukkan, memungkinkan otomatisasi proses transaksi penjualan maupun pembelian barang dan menjaga integritas dan konsistensi data.

1.1.2 Membuat Trigger After Insert (Koperasi Pupuk)

Trigger pada Gambar 1.2 berfungsi untuk memperbarui data pupuk perubahan “jumlah” setelah perintah INSERT dijalankan.

```

1 DELIMITER $$
2 CREATE TRIGGER kurang_stok
3 AFTER INSERT
4 ON detail_jual
5 FOR EACH ROW
6 BEGIN
7     UPDATE pupuk
8     SET stok = stok - NEW.jumlah
9     WHERE kode_pupuk = NEW.kode_pupuk;
10 END $$

```

Gambar 1.2 Query Create Trigger After Update Kurang Stok

1. `DELIMITER \$\$` merupakan deklarasi pembatas yang digunakan untuk membedakan antara perintah SQL dan deklarasi trigger.
2. `CREATE TRIGGER` kurang_stok` merupakan deklarasi pembuatan trigger dengan nama "kurang_stok".
3. `AFTER INSERT` ini menentukan bahwa trigger akan dijalankan setelah operasi insert dilakukan.
4. `ON detail_jual` ini menentukan bahwa trigger akan dijalankan pada tabel "detail_jual".
5. `FOR EACH ROW` ini menentukan bahwa trigger akan dijalankan untuk setiap baris yang diupdate.
6. `BEGIN` ini menandai awal dari blok kode yang akan dijalankan oleh trigger.
7. ` UPDATE pupuk SET stok = stok - NEW.jumlah` pernyataan ini memperbarui kolom stok (persediaan) di tabel pupuk dengan mengurangi nilai jumlah (kuantitas) dari baris baru yang dimasukkan ke dalam tabel detail_jual.
8. `WHERE kode_pupuk = NEW.kode_pupuk` klausa ini memastikan bahwa pembaruan diterapkan pada baris di tabel pupuk yang cocok dengan kode_pupuk (kode produk) dari baris baru yang dimasukkan ke dalam tabel detail_jual.
9. `END \$\$` ini menandai akhir dari blok kode trigger.
10. `DELIMITER;` baris ini mengatur ulang pembatas kembali ke titik koma default (;).

```

1 DELIMITER $$
2 CREATE TRIGGER tambah_stok
3 AFTER INSERT
4 ON detail_jual
5 FOR EACH ROW
6 BEGIN
7     UPDATE pupuk
8     SET stok = stok + NEW.jumlah
9     WHERE kode_pupuk = NEW.kode_pupuk;
10 END $$

```

Gambar 1.3 Query Create Trigger After Update Tambah Stok

1. 'DELIMITER \$\$' merupakan deklarasi pembatas yang digunakan untuk membedakan antara perintah SQL dan deklarasi trigger.
2. 'CREATE TRIGGER' tambah_stok' merupakan deklarasi pembuatan trigger dengan nama "tambah_stok".
3. 'AFTER INSERT' ini menentukan bahwa trigger akan dijalankan setelah operasi insert dilakukan.
4. 'ON detail_jual' ini menentukan bahwa trigger akan dijalankan pada tabel "detail_jual".
5. 'FOR EACH ROW' ini menentukan bahwa trigger akan dijalankan untuk setiap baris yang diupdate.
6. 'BEGIN' ini menandai awal dari blok kode yang akan dijalankan oleh trigger.
7. 'UPDATE pupuk SET stok = stok + NEW.jumlah' pernyataan ini memperbarui kolom stok (persediaan) di tabel pupuk dengan menambah nilai jumlah (kuantitas) dari baris baru yang dimasukkan ke dalam tabel detail_jual.
8. 'WHERE kode_pupuk = NEW.kode_pupuk' klausa ini memastikan bahwa pembaruan diterapkan pada baris di tabel pupuk yang cocok dengan kode_pupuk (kode produk) dari baris baru yang dimasukkan ke dalam tabel detail_jual.
9. 'END \$\$' ini menandai akhir dari blok kode trigger.

10. 'DELIMITER;' baris ini mengatur ulang pembatas kembali ke titik koma default (;).

1.1.3 Awal Tampilan Tabel (*Before Insert*)

	No_bon	Tanggal_bon	No_anggota	Kode_pupuk	Harga_jual	Jumlah	Total
<input type="checkbox"/> Edit Copy Delete	B001	2024-11-01	A001	P001	180000.00	5	1800000.00
<input type="checkbox"/> Edit Copy Delete	B002	2024-11-02	A002	P002	230000.00	5	1150000.00
<input type="checkbox"/> Edit Copy Delete	B003	2024-11-03	A003	P003	155000.00	8	1240000.00
<input type="checkbox"/> Edit Copy Delete	B004	2024-11-04	A004	P004	250000.00	4	1000000.00
<input type="checkbox"/> Edit Copy Delete	B005	2024-11-05	A005	P005	210000.00	7	1470000.00
<input type="checkbox"/> Edit Copy Delete	B006	2024-11-06	A006	P006	190000.00	6	1140000.00
<input type="checkbox"/> Edit Copy Delete	B007	2024-11-07	A007	P007	110000.00	15	1650000.00
<input type="checkbox"/> Edit Copy Delete	B008	2024-11-08	A008	P008	70000.00	20	1400000.00
<input type="checkbox"/> Edit Copy Delete	B009	2024-11-09	A009	P009	65000.00	25	1625000.00
<input type="checkbox"/> Edit Copy Delete	B010	2024-11-10	A010	P010	150000.00	12	1800000.00

Gambar 1.4 Tabel Penjualan Awal

1.1.4 Uji Coba Sintaks Insert

```

1 INSERT INTO `penjualan`
2 (`No_bon`, `Tanggal_bon`, `No_anggota`, `Kode_pupuk`, `Harga_jual`, `Jumlah`, `Total`)
3 VALUES ('B012', '2024-12-09', 'A006', 'P004', '180000', '3', '540000');

```

Gambar 1.5 Query SQL Insert

Jadi, pernyataan SQL ini merupakan pernyataan INSERT yang digunakan untuk memasukkan data baru ke dalam tabel bernama "penjualan". Data yang dimasukkan terdiri dari 7 kolom, yaitu "No_bon", "Tanggal_bon", "No_anggota", "Kode_pupuk", "Harga_jual", "Jumlah", dan "Total". Secara keseluruhan, kode ini berfungsi untuk memasukkan satu baris data baru transaksi penjualan ke dalam tabel "penjualan" dengan nilai-nilai yang telah ditentukan.

1.1.5 Akhir Tampilan Tabel (*After Insert*)

←T→	No_bon	Tanggal_bon	No_anggota	Kode_pupuk	Harga_jual	Jumlah	Total
<input type="checkbox"/> Edit Copy Delete	B001	2024-11-01	A001	P001	180000.00	5	1800000.00
<input type="checkbox"/> Edit Copy Delete	B002	2024-11-02	A002	P002	230000.00	5	1150000.00
<input type="checkbox"/> Edit Copy Delete	B003	2024-11-03	A003	P003	155000.00	8	1240000.00
<input type="checkbox"/> Edit Copy Delete	B004	2024-11-04	A004	P004	250000.00	4	1000000.00
<input type="checkbox"/> Edit Copy Delete	B005	2024-11-05	A005	P005	210000.00	7	1470000.00
<input type="checkbox"/> Edit Copy Delete	B006	2024-11-06	A006	P006	190000.00	6	1140000.00
<input type="checkbox"/> Edit Copy Delete	B007	2024-11-07	A007	P007	110000.00	15	1650000.00
<input type="checkbox"/> Edit Copy Delete	B008	2024-11-08	A008	P008	70000.00	20	1400000.00
<input type="checkbox"/> Edit Copy Delete	B009	2024-11-09	A009	P009	65000.00	25	1625000.00
<input type="checkbox"/> Edit Copy Delete	B010	2024-11-10	A010	P010	150000.00	12	1800000.00
<input type="checkbox"/> Edit Copy Delete	B011	2024-12-09	A001	P001	180000.00	3	540000.00

Gambar 1.6 Tabel Penjualan Akhir

1.1.6 Kesimpulan Trigger After Insert

Pernyataan INSERT INTO untuk memasukkan data baru transaksi penjualan ke dalam tabel "penjualan". Data yang dimasukkan mencakup informasi seperti nomor bon, tanggal, nomor anggota, kode produk, harga jual, jumlah, dan total nilai transaksi dan Trigger "kurang_stok" yang didefinisikan sebelumnya pada tabel "detail_jual". Trigger ini akan secara otomatis dieksekusi setiap kali terjadi operasi INSERT pada tabel "detail_jual". Tugasnya adalah memperbarui stok produk di tabel "pupuk" dengan mengurangi jumlah sesuai dengan data transaksi penjualan yang baru dimasukkan.

Jadi proses insert data penjualan di tabel "penjualan" akan secara otomatis mengurangi stok produk terkait di tabel "pupuk" melalui trigger "kurang_stok". Hal ini menjaga konsistensi data persediaan produk seiring dengan transaksi penjualan yang terjadi. Dengan demikian, proses transaksi penjualan dan pembelian pupuk dapat berjalan secara terintegrasi dan otomatis melalui kombinasi pernyataan INSERT dan trigger yang saling terkait ini.

1.2 Acara 55 dan 56

Buka kembali database studi kasus project akhir “Final Project”, buatlah trigger untuk mengurangi dan menambah stok pada tabel produk.

Table	Action
<input type="checkbox"/> detail_pembelian	★ Browse Structure Search Insert Empty Drop
<input type="checkbox"/> detail_penjualan	★ Browse Structure Search Insert Empty Drop
<input type="checkbox"/> pelanggan	★ Browse Structure Search Insert Empty Drop
<input type="checkbox"/> pembelian	★ Browse Structure Search Insert Empty Drop
<input type="checkbox"/> penjualan	★ Browse Structure Search Insert Empty Drop
<input type="checkbox"/> pn_sementara	★ Browse Structure Search Insert Empty Drop
<input type="checkbox"/> produk	★ Browse Structure Search Insert Empty Drop
<input type="checkbox"/> supplier	★ Browse Structure Search Insert Empty Drop
<input type="checkbox"/> user	★ Browse Structure Search Insert Empty Drop

Gambar 1.7 Structure Database Final Project

1.2.1 Trigger After Insert (Final Project)

Trigger after insert adalah sebuah objek basis data yang secara otomatis dieksekusi setelah operasi penyisipan (*insert*) data baru ke dalam sebuah tabel telah berhasil dilakukan. Jenis Trigger ini biasa digunakan untuk berbagai tujuan seperti pencatatan log, validasi tambahan, atau pembaruan tabel terkait. Selain itu, Trigger After Insert memiliki fungsi utama seperti melakukan tindakan tambahan setelah data baru dimasukkan, memungkinkan otomatisasi proses transaksi penjualan maupun pembelian barang dan menjaga integritas dan konsistensi data.

1.2.2 Membuat Trigger After Insert (Final Project)

Trigger pada Gambar 1.8 berfungsi untuk memperbarui data pupuk perubahan “jumlah” setelah perintah INSERT dijalankan.

```

1 DELIMITER $$
2 CREATE TRIGGER update_stok
3 AFTER INSERT
4 ON detail_pembelian
5 FOR EACH ROW
6 BEGIN
7     UPDATE produk
8     SET stok = stok + NEW.jumlah
9     WHERE id_produk = NEW.id_produk;
10 END $$

```

Gambar 1.8 Query Create Trigger After Update Tambah Stok Produk

1. `DELIMITER \$\$` merupakan deklarasi pembatas yang digunakan untuk membedakan antara perintah SQL dan deklarasi trigger.
2. `CREATE TRIGGER` update_stok` merupakan deklarasi pembuatan trigger dengan nama "update_stok".
3. `AFTER INSERT` ini menentukan bahwa trigger akan dijalankan setelah operasi insert dilakukan.
4. `ON detail_pembelian` ini menentukan bahwa trigger akan dijalankan pada tabel "detail_pembelian".
5. `FOR EACH ROW` ini menentukan bahwa trigger akan dijalankan untuk setiap baris yang didelete.
6. `BEGIN` ini menandai awal dari blok kode yang akan dijalankan oleh trigger.
7. `UPDATE produk SET stok = stok + NEW.jumlah` pernyataan ini memperbarui kolom stok (persediaan) di tabel produk dengan menambah nilai jumlah (kuantitas) dari baris baru yang dimasukkan ke dalam tabel detail_pembelian
8. `WHERE id_produk = NEW.id_produk` klausa ini memastikan bahwa pembaruan diterapkan pada baris di tabel produk yang cocok dengan id_produk (id produk) dari baris baru yang dimasukkan ke dalam tabel detail_pembelian
9. `END \$\$` ini menandai akhir dari blok kode trigger.
10. `DELIMITER;` baris ini mengatur ulang pembatas kembali ke titik koma default (;).

1.1.3 Awal Tampilan Tabel (*Before Insert*)

id_pembelian	total_beli	total_harga	tanggal_transaksi	id_supplier	id_user
PM001	5	260000	2024-12-09	SP005	USR01
PM002	3	108000	2024-12-09	SP005	USR01

Gambar 1.9 Tabel Pembelian Awal

				id_produk	nama_produk	satuan	stok	harga_beli	harga_jual
<input type="checkbox"/>				SL001	Refill Galon	Liter	93	5000	6000
<input type="checkbox"/>				SL002	Galon Aqua	Liter	45	52000	54000
<input type="checkbox"/>				SL003	Galon Cleo	Liter	42	36000	38000
<input type="checkbox"/>				SL004	Galon Vit	Liter	35	43000	45000
<input type="checkbox"/>				SL005	Tukar Galon Aqua	Liter	31	21000	23000
<input type="checkbox"/>				SL006	Tukar Galon Cleo	Liter	26	16000	18000
<input type="checkbox"/>				SL007	Tukar Galon Vit	Liter	30	18000	20000
<input type="checkbox"/>				SL008	Tisu Antiseptik	Pcs	37	1000	1500
<input type="checkbox"/>				SL009	Tutup Galon	Pcs	40	1000	1500

Gambar 1.10 Tabel Produk Awal

1.1.4 Uji Coba Sintaks Insert

```

1 INSERT INTO `pembelian`
2 (`id_pembelian`, `total_beli`, `total_harga`, `tanggal_transaksi`, `id_supplier`, `id_user`)
3 VALUES ('PM003', '9', '189000', '2024-12-09', 'SP004', 'USR01');

```

Gambar 1.11 Query SQL Insert

Jadi, pernyataan SQL ini merupakan pernyataan INSERT yang digunakan untuk memasukkan data baru ke dalam tabel bernama "pembelian". Data yang dimasukkan terdiri dari 6 kolom, yaitu "id_pembelian", "total_beli", "total_harga", "tanggal_transaksi", "id_supplier", dan "id_user". Secara keseluruhan, kode ini berfungsi untuk memasukkan satu baris data baru transaksi pembelian ke dalam tabel "pembelian" dengan nilai-nilai yang telah ditentukan.

1.1.5 Akhir Tampilan Tabel (*After Insert*)

				id_pembelian	total_beli	total_harga	tanggal_transaksi	id_supplier	id_user
<input type="checkbox"/>	Edit	Copy	Delete	PM001	5	260000	2024-12-09	SP005	USR01
<input type="checkbox"/>	Edit	Copy	Delete	PM002	3	108000	2024-12-09	SP005	USR01
<input type="checkbox"/>	Edit	Copy	Delete	PM003	9	189000	2024-12-09	SP004	USR01

Gambar 1.12 Tabel Pembelian Akhir

				id_produk	nama_produk	satuan	stok	harga_beli	harga_jual
<input type="checkbox"/>	Edit	Copy	Delete	SL001	Refill Galon	Liter	93	5000	6000
<input type="checkbox"/>	Edit	Copy	Delete	SL002	Galon Aqua	Liter	45	52000	54000
<input type="checkbox"/>	Edit	Copy	Delete	SL003	Galon Cleo	Liter	42	36000	38000
<input type="checkbox"/>	Edit	Copy	Delete	SL004	Galon Vit	Liter	35	43000	45000
<input type="checkbox"/>	Edit	Copy	Delete	SL005	Tukar Galon Aqua	Liter	40	21000	23000
<input type="checkbox"/>	Edit	Copy	Delete	SL006	Tukar Galon Cleo	Liter	26	16000	18000
<input type="checkbox"/>	Edit	Copy	Delete	SL007	Tukar Galon Vit	Liter	30	18000	20000
<input type="checkbox"/>	Edit	Copy	Delete	SL008	Tisu Antiseptik	Pcs	37	1000	1500
<input type="checkbox"/>	Edit	Copy	Delete	SL009	Tutup Galon	Pcs	40	1000	1500

Gambar 1.13 Tabel Produk Akhir

1.1.6 Kesimpulan Trigger After Insert

Jadi proses insert data penjualan di tabel "pembeliann" akan secara otomatis menambah stok produk terkait di tabel "produk" melalui trigger "update_stok". Hal ini menjaga konsistensi data persediaan produk seiring dengan transaksi pembelian yang terjadi. Dengan demikian, proses transaksi penjualan dan pembelian pupuk dapat berjalan secara terintegrasi dan otomatis melalui kombinasi pernyataan INSERT dan trigger yang saling terkait ini.