

## **Bab-8**

### **Transaction & Triggers**

#### **Pokok Bahasan**

- Memahami konsep transaction database
- Memahami penggunaan transaction dan triggers pada database

#### **Tujuan Praktikum**

Setelah melakukan praktikum mahasiswa diharapkan:

- Mengetahui fitur transaction pada database
- Mampu menggunakan transaction pada proses entri data

#### **Tugas Pendahuluan**

- Jelaskan apa yang dimaksud dengan transaction
- Apa keuntungan dari penggunaan transaction

#### **Percobaan 1: Definiskan skema table**

1. Buatlah database dbinventori
2. Login ke database dbinventori
3. Buat table produk dan isi dengan data berikut ini:

```
dbinventori=# select * from produk;
```

id	nama	harga	stok
1	Teh Kotak	2500	20
2	Pulpen	3500	30
3	Pensil Mekanik	5000	10

1. Buat table pembelian dan isi dengan data berikut ini:

```
dbinventori=# select * from pembelian;
```

id	tanggal	produk_id	qty	hargabeli
1	2014-08-02	1	20	2500
2	2014-08-02	2	30	3500
3	2014-08-02	3	10	5000

## Percobaan 2: Transaksi

1. Input 3 data pembelian menggunakan transaction

```
dbinventori=# begin transaction ;
BEGIN
dbinventori=# insert into pembelian values
              (default, '2014-08-20', 3, 10, 5000);

dbinventori=# insert into pembelian values
              (default, '2014-08-20', 2, 10, 3500);

dbinventori=# insert into pembelian values
              (default, '2014-08-20', 1, 10, 2500);
```

2. Tampilkan seluruh data pembelian : data pembelian sekarang terekam 6 record

```
dbinventori=# SELECT * FROM pembelian;
 id | tanggal | produk_id | qty | hargabeli
-----+-----+-----+-----+-----
  1 | 2014-08-02 |          1 |  20 |        2500
  2 | 2014-08-02 |          2 |  30 |        3500
  3 | 2014-08-02 |          3 |  10 |        5000
  4 | 2014-08-20 |          3 |  10 |        5000
  5 | 2014-08-20 |          2 |  10 |        3500
  6 | 2014-08-20 |          1 |  10 |        2500
(6 rows)
```

3. Batalkan transaksi input 3 data baru dengan perintah ROLLBACK

```
dbinventori=# ROLLBACK ;
```

4. Tampilkan seluruh data pembelian : akan kembali ke kondisi awal

```
dbinventori=# SELECT * FROM pembelian;
 id | tanggal | produk_id | qty | hargabeli
-----+-----+-----+-----+-----
  1 | 2014-08-02 |          1 |  20 |        2500
  2 | 2014-08-02 |          2 |  30 |        3500
  3 | 2014-08-02 |          3 |  10 |        5000
```

5. Lakukan ulang percobaan 1 namun pada langkah ke 3 jalankan perintah COMMIT, untuk merekam transaksi karena transaksi tidak ada perubahan ?
6. Tampilkan seluruh data pembelian ? Apa bedanya dengan langkah 4 ? jelaskan

## Percobaan 2: Transaksi & Triggers

Pada praktikum 2 ini akan dibuat trigger yang akan otomatis mengupdate (menambah secara otomatis) stok produk ketika terjadi transaksi pembelian barang.

1. Buat function update stok produk

```
dbinventori=# CREATE OR REPLACE FUNCTION tambah_stok()
RETURNS TRIGGER AS
$BODY$
BEGIN
    UPDATE produk SET stok=stok+NEW.qty WHERE id=NEW.produk_id ;
    RETURN NEW;
END;
$BODY$
LANGUAGE plpgsql;
```

2. Buat trigger pada event INSERT di table pembelian yang menjalankan fungsi tambah\_stok()

```
dbinventori=# CREATE TRIGGER trig_tambah_stok AFTER INSERT ON
pembelian FOR EACH ROW EXECUTE PROCEDURE tambah_stok();
```

3. Sekarang uji penggunaan transaction dengan trigger

```
dbinventori=# BEGIN TRANSACTION ;
```

4. Tampilkan seluruh data produk, perhatikan stok pada produk yang akan dibeli : Teh Kotak stok awal 20

```
dbinventori=# SELECT * FROM produk ;
```

id	nama	harga	stok
1	Teh Kotak	2500	20
2	Pulpen	3500	30
3	Pensil Mekanik	5000	10

5. Masukkan transaksi pembelian barang Teh Kotak dengan harga 3000 sebanyak 10 buah

```
dbinventori=# INSERT INTO pembelian values
(default,current_date,1,12,3000);
```

6. Tampilkan data pembelian, terlihat data transaksi baru telah terekam

```
dbinventori=# select * from pembelian;
```

id	tanggal	produk_id	qty	hargabeli
1	2014-08-02	1	20	2500
2	2014-08-02	2	30	3500
3	2014-08-02	3	10	5000
8	2014-11-04	1	12	3000

7. Tampilkan seluruh data produk, perhatikan stok pada produk Teh Kotak stok telah berubah menjadi 30

<b>dbinventori=# SELECT * FROM produk ;</b>			
<b>id</b>	<b>nama</b>	<b>harga</b>	<b>stok</b>
-----+-----+-----+-----			
<b>1</b>	<b>Teh Kotak</b>	<b>2500</b>	<b>30</b>
<b>2</b>	<b>Pulpen</b>	<b>3500</b>	<b>30</b>
<b>3</b>	<b>Pensil Mekanik</b>	<b>5000</b>	<b>10</b>

8. Jika ingin membatalkan transaction, jalankan perintah ROLLBACK jika yakin transaksi benar jalankan perintah COMMIT, untuk latihan ini jalankan perintah ROLLBACK dan tampilkan kembali data pada table pembelian dan produk, apakah telah kembali ke kondisi semula ? Buktikan !!!

## **Mandiri !!**

Ubahlah trigger tambah\_stok dengan menambah kriteria perubahan data berikut harga pada table produk. Jika terjadi pembelian dengan harga pembelian lebih mahal dibandingkan harga pada table produk maka lakukan update data pada field harga pada tabel produk dengan margin keuntungan 20% dari harga pembelian. Jika tidak maka harga pada tabel produk tidak berubah.

Misal ditabel produk harga tercantum 2500 sedangkan harga pembelian adalah 3000 maka trigger akan otomatis mengupdate stok bertambah sesuai pembelian dan harga pada table produk berubah menjadi 3600. Laporkan kegiatan ini sesuai langkah-langkah pada percobaan 2