



**Program Studi : Teknik Informatika**

**Laporan Praktikum : Basis Data 2**

**Praktikum 8**

Muhammad Azhar Rasyad  
0110217029

**STT Terpadu Nurul Fikri  
Tahun 2018**

# Transaction

## Tugas Pendahuluan

1. Jelaskan apa yang dimaksud dengan transaction !

Transaction adalah **penyebaran** satu atau lebih **perubahan** ke **database**. Misalnya, jika Anda membuat catatan, memperbarui catatan, atau menghapus rekaman dari tabel, maka Anda melakukan transaksi pada tabel. Penting untuk mengontrol transaksi untuk memastikan integritas data dan menangani kesalahan basis data.

2. Jelaskan keuntungan dari penggunaan transaction !

Berikut merupakan keuntungan dari penggunaan transaction yaitu :

- **Atomicity** : Memastikan bahwa semua operasi dalam unit kerja telah selesai dengan sukses; jika tidak, transaksi tersebut dibatalkan pada titik kegagalan dan operasi sebelumnya digulirkan kembali ke kondisi sebelumnya.
- **Konsistensi** : Memastikan bahwa database dengan benar mengubah status atas transaksi yang berhasil dilakukan.
- **Isolasi** : Memungkinkan transaksi untuk beroperasi secara independen dan transparan satu sama lain.
- **Ketahanan** : Memastikan bahwa hasil atau efek dari transaksi yang dilakukan tetap ada jika terjadi kegagalan sistem.

## Percobaan 1 : Definisikan Skema Table

1. Buatlah database dbinventori

```
apsql@mazharrasyad: ~
mzharrasyad@mzharrasyad:~$ sudo su - apsql
[sudo] password for mzharrasyad:
apsql@mzharrasyad:~$

apsql@mzharrasyad:~$ /home/apsql/pg105/bin/pg_ctl -D /home/apsql/datapg/ -l log
file start
waiting for server to start.... done
server started
apsql@mzharrasyad:~$

apsql@mzharrasyad:~$ /home/apsql/pg105/bin/createdb dbinventori -U apsql -p5555
-h localhost
Password:
apsql@mzharrasyad:~$
```

2. Login ke database dbinventori

```
apsql@mzharrasyad: ~
apsql@mzharrasyad:~$ /home/apsql/pg105/bin/psql dbinventori -U apsql -p5555 -h
localhost
Password for user apsql:
psql (10.5)
Type "help" for help.

dbinventori=#
```

3. Buat table produk dan isi dengan data berikut ini :

```
apsql@mzharrasyad: ~
GNU nano 2.5.3      File: /tmp/psql.edit.2802.sql      Modified

create table produk(
id serial primary key,
nama varchar(20),
harga double precision,
stok double precision
);

```

<b>^G</b> Get Help	<b>^O</b> Write Out	<b>^W</b> Where Is	<b>^K</b> Cut Text	<b>^J</b> Justify	<b>^C</b> Cur Pos
<b>^X</b> Exit	<b>^R</b> Read File	<b>^I</b> Replace	<b>^U</b> Uncut Text	<b>^T</b> To Spell	<b>^_</b> Go To Line

```
apsql@mazharrasyad: ~
GNU nano 2.5.3      File: /tmp/psql.edit.2802.sql      Modified
insert into produk values
(1,'Teh Kotak',2500,20),
(2,'Pulpen',3500,30),
(3,'Pensil Mekanik',5000,10);

^G Get Help  ^O Write Out ^W Where Is  ^K Cut Text  ^J Justify   ^C Cur Pos
^X Exit      ^R Read File ^\ Replace   ^U Uncut Text ^T To Spell  ^_ Go To Line
```

```
apsql@mazharrasyad: ~
dbinventori=# select * from produk;
 id |      nama      | harga | stok
-----+-----+-----+-----
  1 | Teh Kotak      |  2500 |   20
  2 | Pulpen         |  3500 |   30
  3 | Pensil Mekanik |  5000 |   10
(3 rows)

dbinventori=#
```

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

```
apsql@mazharrasyad: ~
GNU nano 2.5.3      File: /tmp/psql.edit.2802.sql      Modified
create table pembelian(
id serial primary key,
tanggal date,
produk_id int references produk(id),
qty double precision,
hargabeli double precision
);

^G Get Help  ^O Write Out ^W Where Is  ^K Cut Text  ^J Justify   ^C Cur Pos
^X Exit      ^R Read File ^\ Replace   ^U Uncut Text ^T To Spell  ^_ Go To Line
```

```
apsql@mazharrasyad: ~
GNU nano 2.5.3      File: /tmp/psql.edit.2802.sql      Modified
insert into pembelian values
(1,'20140802',1,20,2500),
(2,'20140802',2,30,3500),
(3,'20140802',3,10,5000);

[ Read 4 lines ]

^G Get Help  ^O Write Out ^W Where Is  ^K Cut Text  ^J Justify   ^C Cur Pos
^X Exit      ^R Read File ^\ Replace   ^U Uncut Text ^T To Spell  ^_ Go To Line
```

apsql@mazharrasyad: ~

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

(3 rows)

dbinventori=#

## Percobaan 2 : Transaksi

1. Input 3 data pembelian menggunakan transaction

```
apsql@mazharrasyad: ~
dbinventori=# begin transaction;
BEGIN
dbinventori=# insert into pembelian values
dbinventori=# (4,'20140820',3,10,5000);
INSERT 0 1
dbinventori=# insert into pembelian values
dbinventori=# (5,'20140820',2,10,3500);
INSERT 0 1
dbinventori=# insert into pembelian values
dbinventori=# (6,'20140820',1,10,2500);
INSERT 0 1
dbinventori=#
```

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

```
apsql@mazharrasyad: ~
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)
dbinventori=#
```

3. Batalkan transaksi input 3 data baru dengan perintah ROLLBACK

```
apsql@mazharrasyad: ~
dbinventori=# ROLLBACK;
ROLLBACK
dbinventori=#
```

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

```
apsql@mazharrasyad: ~
dbinventori=# select * from pembelian;
 id | tanggal   | produk_id | qty | hargabeli
-----+-----+-----+-----+-----
  1 | 2014-08-02 |          1 |  20 |        3600
  2 | 2014-08-02 |          2 |  30 |        3500
  3 | 2014-08-02 |          3 |  10 |        5000
(3 rows)
dbinventori=#
```

5. Lakukan ulang percobaan 1 namun pada langkah ke 3 jalankan perintah COMMIT, untuk merekam transaksi karena transaksi tidak ada perubahan ?

```
apsql@mazharrasyad: ~
dbinventori=# begin transaction;
BEGIN
dbinventori=# insert into pembelian values
dbinventori-# (4,'20140820',3,10,5000);
INSERT 0 1
dbinventori=# insert into pembelian values
dbinventori-# (5,'20140820',2,10,3500);
INSERT 0 1
dbinventori=# insert into pembelian values
dbinventori-# (6,'20140820',1,10,2500);
INSERT 0 1
dbinventori=#
```

```
apsql@mazharrasyad: ~
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)
dbinventori=#
```

```
apsql@mazharrasyad: ~
dbinventori=# commit;
COMMIT
dbinventori=#
```

6. Tampilkan seluruh data pembelian ? Apa bedanya dengan langkah 4 ? jelaskan

```
apsql@mazharrasyad: ~
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)
dbinventori=#
```

- Ketika sudah masuk kedalam mode transaksi (begin transaction) maka dapat diakhiri dengan 2 cara yaitu rollback (dibatalkan) atau commit (disetujui).
- Bedanya dengan langkah ke 4 adalah menampilkan table pembelian seperti sebelumnya tanpa ada perubahan karena menggunakan perintah rollback

### Percobaan 3 : Transaksi & Triggers

1. Buat function update stok produk

```
apsql@mazharrasyad: ~
GNU nano 2.5.3      File: /tmp/psql.edit.2802.sql      Modified
create or replace function tambah_stok()
returns trigger as
$$
    begin
        update produk set stok=stok + new.qty where id=new.produk_id;
        return new;
    end
$$ language plpgsql;
```

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

```
apsql@mazharrasyad: ~
GNU nano 2.5.3      File: /tmp/psql.edit.2802.sql      Modified
create trigger trig_tambah_stok
after insert on pembelian for each row
execute procedure tambah_stok();
```

3. Sekarang uji penggunaan transaction dengan trigger

```
apsql@mazharrasyad: ~
dbinventori=# begin transaction;
BEGIN
dbinventori=#
```

```
apsql@mazharrasyad: ~
dbinventori=# begin transaction;
BEGIN
dbinventori=# insert into pembelian values
dbinventori-# (7,current_date,1,20,3000);
INSERT 0 1
dbinventori=#
```



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

```
apsql@mazharrasyad: ~
dbinventori=# select * from produk;
id |      nama      | harga | stok
----+-----+-----+----
 1 | Teh Kotak      | 2500  | 20
 2 | Pulpen         | 3500  | 40
 3 | Pensil Mekanik | 5000  | 20
(3 rows)

dbinventori=#
```

5. Masukkan transaksi pembelian barang Teh Kotak dengan harga 3000 sebanyak 12 buah

```
apsql@mazharrasyad: ~
dbinventori=# insert into pembelian values (7,current_date,1,12,3000);
INSERT 0 1
dbinventori=#
```

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

```
apsql@mazharrasyad: ~
dbinventori=# select * from pembelian;
id | tanggal       | produk_id | qty | hargabeli
----+-----+-----+----+-----
 1 | 2014-08-02    |          1 | 20 | 3600
 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
 7 | 2018-11-22    |          1 | 12 | 3000
(7 rows)

dbinventori=#
```

7. Tampilkan seluruh data produk, perhatikan stok pada produk Teh Kotak stok telah berubah menjadi 32

```
apsql@mazharrasyad: ~
dbinventori=# select * from produk;
id |      nama      | harga | stok
----+-----+-----+----
 2 | Pulpen         | 3500  | 30
 3 | Pensil Mekanik | 5000  | 10
 1 | Teh Kotak      | 2500  | 32
(3 rows)

dbinventori=#
```

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 !!!
- Tampilkan table produk sebelum ada transaksi dan pada latihan kali ini akan mencoba mengubah data produk dengan id = 3 dengan stok = 20

```
apsql@mazharrasyad: ~
dbinventori=# select * from produk;
 id |      nama      | harga | stok
----+-----+-----+----
  1 | Teh Kotak      |  2500 |   32
  2 | Pulpen         |  3500 |   40
  3 | Pensil Mekanik |  5000 |   20
(3 rows)
dbinventori=#
```

- Tampilkan table pembelian sebelum ada transaksi dengan id terakhir adalah 7

```
apsql@mazharrasyad: ~
dbinventori=# select * from pembelian;
 id | tanggal   | produk_id | qty | hargabeli
----+-----+-----+----+-----
  1 | 2014-08-02 |         1 |  20 |       3600
  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
  7 | 2018-11-22 |         1 |  12 |       3000
(7 rows)
dbinventori=#
```

- Memulai transaction

```
apsql@mazharrasyad: ~
dbinventori=# begin transaction;
BEGIN
dbinventori=#
```

- Masukkan data ke table pembelian dengan id = 8

```
apsql@mazharrasyad: ~
dbinventori=# insert into pembelian values (8,current_date,3,5,4000);
INSERT 0 1
dbinventori=#
```

- Menampilkan table produk dengan id = 3 dan pada stoknya akan bertambah 5

```

apsql@mazharrasyad: ~
dbinventori=# select * from produk;
 id |      nama      | harga | stok
-----+-----+-----+-----
  1 | Teh Kotak      |  2500 |   32
  2 | Pulpen         |  3500 |   40
  3 | Pensil Mekanik |  5000 |   25
(3 rows)

dbinventori=#

```

- Menampilkan table pembelian dan id yang terakhir adalah id = 8

```

apsql@mazharrasyad: ~
dbinventori=# select * from pembelian;
 id | tanggal   | produk_id | qty | hargabeli
-----+-----+-----+-----+-----
  1 | 2014-08-02 |          1 |  20 |        3600
  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
  7 | 2018-11-22 |          1 |  12 |        3000
  8 | 2018-11-23 |          3 |   5 |        4000
(8 rows)

dbinventori=#

```

- Lakukan ROLLBACK untuk membuktikan bahwa kondisi diatas akan kembali seperti semula

```

apsql@mazharrasyad: ~
dbinventori=# ROLLBACK;
ROLLBACK
dbinventori=#

```

- Table produk

```

apsql@mazharrasyad: ~
dbinventori=# select * from produk;
 id |      nama      | harga | stok
-----+-----+-----+-----
  1 | Teh Kotak      |  2500 |   32
  2 | Pulpen         |  3500 |   40
  3 | Pensil Mekanik |  5000 |   20
(3 rows)

dbinventori=#

```

- Table pembelian

```
apsql@mazharrasyad: ~
dbinventori=# select * from pembelian;
 id |  tanggal  | produk_id | qty | hargabeli
-----+-----+-----+-----+-----
  1 | 2014-08-02 |          1 |  20 |        3600
  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
  7 | 2018-11-22 |          1 |  12 |        3000
(7 rows)

dbinventori=#
```

- Kesimpulannya : Terbukti

## 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 di **tabel produk** **harga** tercantum **2500** sedangkan **harga pembelian** adalah **3000** maka trigger akan

- Otomatis mengupdate stok bertambah sesuai pembelian
- Harga pada table produk berubah menjadi 3600.

Laporkan kegiatan ini sesuai langkah-langkah pada percobaan 2 :

1. Buat function `tambah_stok` yang telah diubah

```
apsql@mazharrasyad: ~
GNU nano 2.5.3      File: /tmp/psql.edit.6170.sql      Modified
create or replace function tambah_stok()
returns trigger as
$$
    declare
        v_harga double precision;
    begin
        select into v_harga harga from produk where id = new.produk_id;

        if v_harga < new.hargabeli then
            v_harga = new.hargabeli + (new.hargabeli * 0.2);
            update produk set harga = v_harga where id = new.produk_id;
        end if;

        update produk set stok = stok + new.qty where id = new.produk_id;
        return new;
    end
$$ language plpgsql;
```

^G Get Help ^O Write Out ^W Where Is ^K Cut Text ^J Justify ^C Cur Pos  
^X Exit ^R Read File ^\ Replace ^U Uncut Text ^T To Spell ^ Go To Line

2. Buat trigger untuk menjalankan function tambah\_stok

```
apsql@mazharrasyad: ~
GNU nano 2.5.3 File: /tmp/psql.edit.6170.sql Modified
create trigger trig_tambah_stok
after insert on pembelian for each row
execute procedure tambah_stok();
^G Get Help ^O Write Out ^W Where Is ^K Cut Text ^J Justify ^C Cur Pos
^X Exit ^R Read File ^\ Replace ^U Uncut Text ^T To Spell ^_ Go To Line
```

3. Tampilkan table produk dan yang akan ditambah stok adalah id = 1

```
apsql@mazharrasyad: ~
dbinventori=# select * from produk;
 id |      nama      | harga | stok
-----+-----+-----+-----
  1 | Teh Kotak      |  2500 |   32
  2 | Pulpen         |  3500 |   40
  3 | Pensil Mekanik |  5000 |   20
(3 rows)

dbinventori=#
```

4. Tampilkan table pembelian untuk melihat stok yang terakhir ditambah

```
apsql@mazharrasyad: ~
dbinventori=# select * from pembelian;
 id | tanggal   | produk_id | qty | hargabeli
-----+-----+-----+-----+-----
  1 | 2014-08-02 |          1 |  20 |        3600
  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
  7 | 2018-11-22 |          1 |  12 |        3000
(7 rows)

dbinventori=#
```

5. Mulai transaksi dan menginputkan data stok seperti contoh pada soal

```
apsql@mazharrasyad: ~
dbinventori=# begin transaction;
BEGIN
dbinventori=# insert into pembelian values (8,current_date,1,3,3000);
INSERT 0 1
dbinventori=#
```

6. Terlihat bahwa harga pada table produk dengan id = 1 berubah yang tadinya 2500 menjadi 3600 dari penambahan keuntungan 20% hargabeli pembelian 3000

```
apsql@mazharrasyad: ~
dbinventori=# select * from produk;
 id |      nama      | harga | stok
-----+-----+-----+-----
  2 | Pulpen         | 3500  | 40
  1 | Teh Kotak      | 3600  | 35
  3 | Pensil Mekanik | 5000  | 20
(3 rows)

dbinventori=#
```

7. Memeriksa table pembelian

```
apsql@mazharrasyad: ~
dbinventori=# select * from pembelian;
 id | tanggal   | produk_id | qty | hargabeli
-----+-----+-----+-----+-----
  1 | 2014-08-02 |          1 | 20 | 3600
  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
  7 | 2018-11-22 |          1 | 12 | 3000
  8 | 2018-11-23 |          1 |  3 | 3000
(8 rows)

dbinventori=#
```

8. Jika sudah seperti diatas maka langkah terakhir untuk mengakhiri transaction adalah COMMIT

```
apsql@mazharrasyad: ~
dbinventori=# commit;
COMMIT
dbinventori=#
```

9. Kemudian periksa kembali table produk dan pembelian apakah sudah berubah atau tidak

```
apsql@mazharrasyad: ~
dbinventori=# select * from produk;
id |      nama      | harga | stok
---+-----+-----+---
 2 | Pulpen         | 3500  | 40
 1 | Teh Kotak      | 3600  | 35
 3 | Pensil Mekanik | 5000  | 20
(3 rows)

dbinventori=# select * from pembelian;
id | tanggal   | produk_id | qty | hargabeli
---+-----+-----+---+-----
 1 | 2014-08-02 |          1 | 20 | 3600
 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
 7 | 2018-11-22 |          1 | 12 | 3000
 8 | 2018-11-23 |          1 | 3  | 3000
(8 rows)

dbinventori=#
```

----- Selesai -----



## Referensi

- [https://www.tutorialspoint.com/postgresql/postgresql\\_transactions.htm](https://www.tutorialspoint.com/postgresql/postgresql_transactions.htm)