Basis Data Lanjut

Semester Genap 2018/2019

Procedure, Function dan Trigger

Salhazan Nasution, S.Kom, MIT

Procedure, Function dan Trigger

Stored Program

- Procedure

⇨Program yang tersusun atas serangkaian statement SQL dan atau blok PL/SQL dengan nama tertentu, yang dikompilasi dan disimpan dalam server database

- Function

⇨prinsipnya sama dengan prosedur, perbedaannya adalah fungsi merupakan subprogram yang berfungsi mengembalikan suatu nilai balikan (*return values*)

Procedure

```
CREATE [OR REPLACE] PROCEDURE nama_prosedur
 [(parameter tipedata,...)]
 IS|AS
   --bagian deklarasi variabel
 BEGIN
   --bagian eksekusi perintah/program
        (statement SQL dan atau blok PL/SQL)
   EXCEPTION
     --bagian deklarasi penanganan kesalahan
          (exception handling)
 END [nama_prosedur];
 /
```

Procedure (2)

Pemanggilan dengan SQL*Plus (Command Line) :

```
EXECUTE nama_prosedur
```

Pemanggilan dengan i-SQL*Plus (Browser) :

```
BEGIN
  nama_prosedur
END;
```

Menghapus prosedur :

```
DROP PROCEDURE nama_prosedur;
```

Procedure – sederhana

```
CREATE OR REPLACE PROCEDURE selamat_datang
IS
BEGIN
   DBMS_OUTPUT.PUT_LINE('SELAMAT DATANG ');
   DBMS_OUTPUT.PUT_LINE('DI BASIS DATA LANJUT ');
END;
/

EXECUTE selamat_datang;
```

Procedure – tanpa parameter

```
CREATE or REPLACE PROCEDURE lihat_stok
IS
  stk barang.stok%type;
BEGIN
  select SUM (stok) into stk from barang;
  dbms_output.put_line ('Jumlah stok barang : ' ||
  stk);
END;
/


EXECUTE lihat_stok;
```

Procedure – dengan parameter

```
CREATE or REPLACE PROCEDURE update_stok
(nm_brg barang.nama_barang%type,
stk barang.stok%type)
IS
BEGIN
  UPDATE barang SET stok=stk WHERE
   nama_barang=nm_brg;
END;
/


EXECUTE update_stok('fanta',15);
```

Function

```
CREATE [OR REPLACE] FUNCTION nama_fungsi [(parameter
   tipedata,...)]
  RETURN tipedata_keluaran
  IS|AS
       --bagian deklarasi variabel
BEGIN
       --bagian eksekusi perintah/program
       {statement SQL dan atau blok PL/SQL)
RETURN nilai_keluaran;
EXCEPTION
--bagian deklarasi penanganan kesalahan (exception
   handling)
END [nama_fungsi];
/
```

Function (2)

Pemanggilan dengan SQL*Plus (Command Line) :

```
EXECUTE dbms_output.put_line(nama_fungsi);
```

Pemanggilan dengan i-SQL*Plus (Browser) :

```
BEGIN
  dbms_output.put_line(nama_fungsi);
END;
```

Menghapus fungsi :

```
DROP FUNCTION nama_fungsi;
```

Function – tanpa parameter

```
CREATE OR REPLACE FUNCTION cari_harga_tertinggi
RETURN barang.harga%type
IS
hrg barang.harga%type;
BEGIN
 SELECT max(harga) INTO hrg FROM barang;
  DBMS_OUTPUT.PUT('Harga yang tertinggi sebesar= ');
  RETURN hrg;
 END;
/


EXECUTE dbms_output.put_line(cari_harga_tertinggi);
```

Function – dengan parameter

```
CREATE OR REPLACE FUNCTION stok_barang
(nama barang.nama_barang%type)
RETURN barang.stok%type
IS
stock barang.stok%type;
BEGIN
  SELECT stok INTO stock FROM barang
  WHERE nama_barang=nama;
  DBMS_OUTPUT.PUT('Stok '||nama ||' ada sebanyak= ');
RETURN stock;
END;
/

EXECUTE dbms_output.put_line(stok_barang('relaxa'));
```

Trigger

⇨Store-procedure yang otomatis dipanggil oleh Oracle, ketika diberikan perintah manipulasi data terhadap suatu tabel (insert, update, delete)

Trigger (2)

Kegunaan Trigger :

- Menentukan nilai kolom-kolom tertentu secara otomatis
- Menghindari transaksi data yang tidak valid
- Membuat autorisasi sekuriti yang kompleks
- Membuat business rule yang kompleks

Trigger (3)

Trigger dapat dipasangkan sebelum atau sesudah (before atau after) perintah tersebut. Pemanggilan trigger tidak dilakukan secara eksplisit oleh pemakai, melainkan oleh Oracle pada saat Oracle mendeteksi adanya kejadian (*event*) yang menjadi syarat pemanggilan trigger.

Tipe Trigger

- Berdasarkan jumlah aksi :

⇨Trigger row

⇨Trigger statement

- Berdasarkan waktu kejadian :

⇨Trigger after

⇨Trigger before

Aktivasi Trigger

- Meng-aktif/non-aktif trigger:

```
ALTER TRIGGER nama_trigger ENABLE|DISABLE;
```

- Menghapus trigger:

```
DROP TRIGGER nama_trigger;
```

Trigger - Statement

```
create or replace trigger barang_a_i_s
after insert on barang
begin
 dbms_output.put_line('Proses penambahan data barang berhasil');
end;
/
```

Menguji trigger :

```
insert into barang values (4,1,'sprite',4500,10);
```

Trigger - Statement

```
create or replace trigger barang_a_iud_s
after insert or update or delete on barang
begin
if inserting then
 dbms_output.put_line('Proses penambahan data barang
berhasil');
elsif updating then
 dbms_output.put_line('Data barang berhasil diubah');
elsif deleting then
 dbms_output.put_line('Proses penghapusan data barang
berhasil');
else null;
end if;
end;
/
```

Trigger - Row

```
create or replace trigger pengurangan_stok
after insert on detail_penjualan
for each row
declare
  id detail_penjualan.id_barang%type;
  jumlah detail_penjualan.jml%type;
begin
  id := :NEW.id_barang;
  jumlah := :NEW.jml;
  update barang set stok=(stok-jumlah) where id_barang=id;
end;
/
```

Trigger – Row (2)

Trigger di atas digunakan untuk mengurangi stok secara otomatis ketika terjadi penjualan terhadap barang yang bersangkutan. Sehingga tidak perlu melakukan update terhadap stok yang ada.

Menurut Anda, apakah logika trigger di atas benar?

Bagaimana jika stok barang sudah habis? Apakah tetap bisa melakukan penjualan?

Trigger – Row (3)

```
create or replace trigger pengurangan_stok
before insert on detail_penjualan
for each row
declare
  id detail_penjualan.id_barang%type;
  jumlah detail_penjualan.jml%type;
  sisa detail_penjualan.jml%type;
begin
  id := :NEW.id_barang;
  jumlah := :NEW.jml;
  select stok into sisa from barang where id_barang=id;
  if (sisa-jumlah) < 0 then
  raise_application_error(-20005,'Stok tidak mencukupi');
  else
  update barang set stok=(stok-jumlah) where id_barang=id;
  dbms_output.put_line('Penjualan sukses !');
  end if;
end;
/
```

Trigger – Row (4)

- Trigger di atas digunakan untuk mengurangi stok terhadap barang yang dijual.
- Sebelum melakukan penjualan, trigger tersebut akan melakukan pengecekan terhadap stok barang yang ada (tabel barang), apabila mencukupi maka penjualan akan dilakukan, tetapi apabila stok barang tidak mencukupi maka penjualan tidak akan dapat dilakukan (tabel detail_penjualan).

Any Question?
See you next time..