

Basis Data 2

09

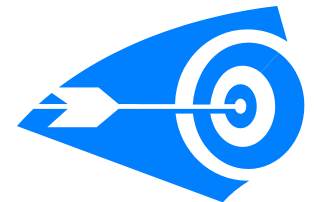
Procedure dan Function Database

*Arif Basofi, S.Kom. MT.
Teknik Informatika, PENS*

Tujuan

Dapat Memahami Pembuatan Procedure dan Function dalam Database:

- ▶ PL/SQL Procedure
- ▶ PL/SQL Function
- ▶ PL/SQL Stored Procedure



PL/SQL Procedure

- ▶ Procedure dalam pemrograman sangat membantu dalam menangani kompleksitas tugas dalam proses sistem, yang dapat membagi menjadi beberapa potongan tugas.
- ▶ **Procedure** adalah subprogram yg digunakan untuk melakukan proses tertentu.
- ▶ **PL/SQL Procedure** adalah procedure dalam bahasa pemrograman PL/SQL database Oracle, yang berbentuk block dengan rangkaian optional parameter dan mendukung penuh standar SQL.
- ▶ Procedure bisa disimpan dalam database sebagai object schema, shg suatu procedure bisa digunakan berulang kali tanpa harus melakukan parsing dan compile ulang.
- ▶ Procedure yang disimpan didatabase biasa disebut **stored procedure**.

PL/SQL Procedure

- ▶ Tiap parameter mengandung:
 - ▶ nama parameter,
 - ▶ penggunaan (IN, OUT, IN-OUT), dan
 - ▶ tipe data
- ▶ **(IN)** adalah parameter input yang seharusnya tidak mengalami perubahan dalam procedure.
- ▶ **(OUT)** adalah parameter output yang memberikan suatu nilai didalam procedure.
- ▶ **(IN-OUT)** adalah parameter input-output, yang seharusnya memiliki suatu nilai input yang telah disediakan diluar procedure, akan tetapi dapat diubah saat didalam procedure.

PL/SQL Procedure (con't)

- ▶ Spesifikasi **tipe data** seharusnya **tidak** mengandung beberapa **constraint** seperti **length**.
- ▶ Misal untuk parameter dengan tipe string, gunakan tipe data **VARCHAR2**, tidak perlu menyertakan ukuran panjangnya.
- ▶ **Syntax Struktur Procedure:**

```
CREATE [OR REPLACE] PROCEDURE ProcedureName
    [ (Parameter1,..., ParameterN) ]
IS
    [ sequence of declarations ]
BEGIN
    sequence of statements
    [ EXCEPTION
        sequence of statements to respond to exceptions ]
END;
```

PL/SQL Procedure (con't)

▶ Contoh-1:

Berikut contoh sederhana procedure **pr_InsertRegistration**: yang akan memasukkan suatu row kedalam table **Registration** dalam database **University**. Parameter input mengandung sejumlah nilai parameter untuk di-insert-kan. Procedure **dbms_output.put_line** akan call (memanggil) tampilan pesan bahwa insert was successful

```
CREATE OR REPLACE PROCEDURE pr_InsertRegistration
(aRegNo IN Registration.RegNo%TYPE,
 aSTDSSN IN Registration.StdSSN%TYPE,
 aRegStatus IN Registration.RegStatus%TYPE,
 aRegDate IN Registration.RegDate%TYPE,
 aRegTerm IN Registration.RegTerm%TYPE,
 aRegYear IN Registration.RegYear%TYPE
)
IS
BEGIN
    INSERT INTO Registration (RegNo, StdSSN, RegStatus, RegDate,
        RegTerm, RegYear)
    VALUES (aRegNo, aStdSSN, aRegStatus, aRegDate, aRegTerm, aRegYear);
    DBMS_OUTPUT.PUT_LINE('Added a row to the Registration table');
END;
```

PL/SQL Procedure (con't)

--- Testing Code Contoh-1 ---

```
SET SERVEROUTPUT ON;
-- Cek total row sebelum eksekusi procedure
SELECT COUNT(*) FROM Registration;

BEGIN
    pr_InsertRegistration (1250,'901-25-4567',
        To_Date('21-Sep-2002'),'Spring',2002);
END;
/

--Cek total row sesudah eksekusi procedure
SELECT COUNT(*) FROM Registration;
--Hapus row yang telah di-insert
ROLLBACK;
```

PL/SQL Procedure (con't)

▶ Contoh-2:

Agar procedure diatas dapat digunakan oleh **procedure** yang lain, sebaiknya struktur code-nya diubah menjadi rancangan yang baik dengan mengganti tampilan output yang akan menangkap success atau failure yang akan terjadi, dengan menambahkan **EXCEPTION OTHER**. **Exception OTHER** akan menangkap segala kesalahan yang terjadi, misal karena violation of primary atau foreign key constraint.

```
CREATE OR REPLACE PROCEDURE pr_InsertRegistration
(aRegNo IN Registration.RegNo%TYPE,
 aSTDSSN IN Registration.StdSSN%TYPE,
 aRegStatus IN Registration.RegStatus%TYPE,
 aRegDate IN Registration.RegDate%TYPE,
 aRegTerm IN Registration.RegTerm%TYPE,
 aRegYear IN Registration.RegYear%TYPE,
 aResult OUT BOOLEAN
)
IS
BEGIN
    aResult := TRUE;
    INSERT INTO Registration (RegNo,StdSSN,RegStatus,RegDate,RegTerm,RegYear)
    VALUES (aRegNo, aStdSSN, aRegStatus, aRegDate, aRegTerm, aRegYear);
    EXCEPTION
    WHEN OTHERS THEN aResult := FALSE;
END;
```


PL/SQL Procedure (con't)

--- Testing Code Contoh-2---

```
SET SERVEROUTPUT ON;
-- Cek row sebelum eksekusi procedure
SELECT COUNT(*) FROM Registration;
DECLARE
-- Parameter output harus dideklarasikan dahulu Result BOOLEAN;
BEGIN
pr_InsertRegistration (1250,'901-23-4567',To_Date('21-Sep-2002'),'Spring',2002);
IF Result THEN
    Dbms_output.put_line('Added a row to the Registration table');
ELSE
    Dbms_output.put_line('Row not added to the Registration table');
END IF;
END;
/
--Cek row sesudah eksekusi procedure
SELECT COUNT(*) FROM Registration;
--Hapus row yang telah di-insert
ROLLBACK;
```



PL/SQL Stored Procedure

- ▶ **Database Procedure** atau **Stored Procedure** adalah prosedur bahasa pemrograman akan tetapi tersimpan dan dimanage oleh DBMS (tersimpan pada sisi server database), berbeda seperti pada prosedur bahasa pemrograman yang lain.
- ▶ Beberapa alasan perlunya stored procedure dalam DBMS:
 1. DBMS dapat meng-compile struktur bahasa pemrograman dalam stored procedure selama menggunakan standar SQL statement.
 2. Stored Procedure memberikan **kemudahan** dalam pengembangan dan pembuatan aplikasi **client-server**.

Stored Procedure disimpan dalam server database sehingga tidak perlu membuat replikasi pada tiap client.

Pada awal munculnya aplikasi **client-server**, kemampuan stored procedure pada server merupakan **teknik penting** pada saat itu, akan tetapi dengan kemajuan teknologi saat ini sudah banyak alternatif lain selain menggunakan stored procedure, misal. dalam pembuatan aplikasi web dibutuhkan teknologi dalam menangani **distributed object** di web. (pendekatan object oriented)

PL/SQL Stored Procedure (con't)

3. **Stored Procedure** memberikan kemudahan dalam pembuatan operator dan fungsi yang lebih kompleks dari dukungan standar SQL.
4. Administrator database dapat manage stored procedure dengan menggunakan **tool yang sama** dalam menangani bagian-bagian lain aplikasi database. Sebagian besar stored procedure dimanage oleh **sistem security DBMS**.

PL/SQL Function

- ▶ Fungsi berbeda dengan **procedure**.
- ▶ Fungsi seharusnya memberikan nilai balik (**return value**).
- ▶ PL/SQL Function mirip dengan PL/SQL Procedure yang mengandung rangkaian parameter. Akan tetapi, PL/SQL Function hanya menggunakan parameter input saja.
- ▶ Setelah rangkaian parameter didefinisikan, tipe data return juga didefinisikan. Sehingga dalam *function body* akan terdapat statement **RETURN** untuk men-generate nilai output dari fungsi tersebut.

PL/SQL Function (con't)

▶ **Syntax struktur function:**

```
CREATE [OR REPLACE] FUNCTION FunctionName
    [ (Parameter1,..., ParameterN) ]
RETURN DataType
IS
    [ sequence of declarations ]
BEGIN
    sequence of statements including RETURN statement
[ EXCEPTION
    sequence of statements to respond to exceptions ]
END;
```

PL/SQL Function (con't)

▶ Contoh-function:

Berikut contoh fungsi sederhana **fn_RetrieveStdName**:

- ▶ Yang akan me-retrieve (mendapatkan) nama student dari *Social Security Number (SSN)*.
- ▶ Predefined exception *No_Data_Found* adalah true jika statement SELECT tidak akan mengembalikan minimal satu nilai row.
- ▶ Statement SELECT menggunakan klausa INTO untuk menghubungkan variable dengan kolom database.
- ▶ Klausa **INTO** dapat digunakan hanya jika statement SELECT *return at most one row* (hanya mengembalikan nilai maks. 1 baris data).
- ▶ Jika klausa INTO digunakan saat statemen SELECT *return more than one row*, maka **exception** perlu **digenerate** untuk menampilkan **pesan kesalahan** dengan menggunakan prosedur *Raise_Application_Error*.

PL/SQL Function (con't)

▶ Contoh-function:

```
CREATE OR REPLACE FUNCTION fn_RetrieveStdName
(aSTDSSN IN Student.StdSSN%TYPE)
RETURN VARCHAR2
IS
  aFirstName Student.StdFirstName%TYPE;
  aLastName Student.StdLasttName%TYPE;
BEGIN
  SELECT StdFirstName, StdLastName
     INTO aFirstName, aLastName
  FROM Student
 WHERE StdSSN = aSTDSSN;
  RETURN(aLastName||', '||aFirstName);
EXCEPTION
  --No_Data_Found is raised if the SELECT statement return no data.
  WHEN No_Data_Found THEN
    RETURN(NULL);
  WHEN OTHERS THEN
    raise_application_error(-20001,'Database Error');
END;
```

PL/SQL Function (con't)

--- Testing Code Contoh-Function---

```
SET SERVEROUTPUT ON;
DECLARE
  aStdName VARCHAR2(50);
BEGIN
  --This call should display a student name
  aStdName:= fn_RetrieveStdName('901-23-4567');
  IF aStdName IS NULL THEN
    Dbms_output.put_line('Student not found');
  ELSE
    Dbms_output.put_line('Name is ' || aStdName);
  END IF;
  --This call should not display a student name
  aStdName:= fn_RetrieveStdName('905-23-4567');
  IF aStdName IS NULL THEN
    Dbms_output.put_line('Student not found');
  ELSE
    Dbms_output.put_line('Name is ' || aStdName);
  END IF;
END;
```