

Deklarasi Variabel PL/SQL

Tujuan Pembelajaran :

- Mengetahui blok PL/SQL beserta bagiannya
- Memahami variable dalam PL/SQL
- Dapat mendeklarasikan dan menggunakan variable pada PL/SQL

2.1. Struktur Blok PL/SQL

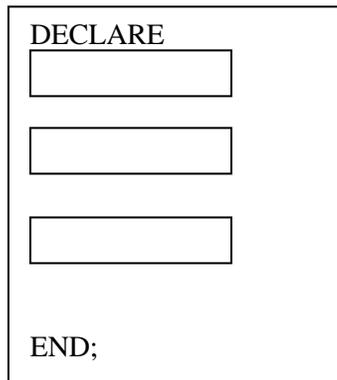
PL/SQL merupakan *block-structured language*, dalam arti : program yang ditulis dengan PL/SQL dapat dibagi-bagi ke dalam blok logika.

Blok dalam PL/SQL terdiri dari bagian-bagian, sebagai berikut :

1. Deklarasi
2. Bagian yang dapat dijalankan (executable)
3. Bagian penanganan kesalahan (exception handling)

Bagian deklarasi dan penanganan kesalahan sifatnya optional (bisa ditulis, bisa tidak ditulis).

Secara umum, struktur PL/SQL :



DECLARE merupakan bagian deklarasi.

Antara *BEGIN* dan *END* merupakan bagian yang dapat dijalankan (executable).

EXCEPTION merupakan bagian penanganan kesalahan.

2.2. Tipe-tipe Block

Ada 3 (tiga) tipe block dalam PL/SQL :

- Anonymous

Format umum :

```

  [DECLARE]
  BEGIN
  --Statements
  [EXCEPTION]
  END;
  
```

- Procedure

Format umum :

```

  PROCEDURE name
  IS
  BEGIN
  --Statements
  [EXCEPTION]
  END;
  
```

- Function

Format umum :

```
FUNCTION name
RETURN datatype
IS
BEGIN
  --Statements
  RETURN value;
[EXCEPTION]

END;
```

2.3. Konstruksi Program

PL/SQL dapat digunakan pada 2 (dua) lingkungan berikut :

- Oracle Server (Database Server constructs)
- Oracle Development Tools (Tools Construct)

Berikut ini bagian-bagian PL/SQL yang dapat mengkonstruksi kedua lingkungan yang berbeda tersebut :

- Tools Constructs
 - Anonymous blocks
 - Application procedures or functions
 - Application packages
 - Application triggers
 - Object types
- Database Server Constructs
 - Anonymous blocks
 - Stored procedures or function
 - Stored packages
 - Database triggers
 - Object types

2.4. Penggunaan Variabel

Variabel dalam PL/SQL dapat digunakan untuk :

- Penyimpanan data sementara
- Manipulasi nilai yang sudah tersimpan
Variabel dapat digunakan untuk proses manipulasi nilai yang sudah tersimpan dalam database Oracle.
- Reusability
Variabel dapat digunakan kembali dengan melakukan referensi dari statement yang lain.
- Memudahkan maintenance
Pada penggunaan %TYPE dan %ROWTYPE, deklarasi variable dibuat berdasarkan definisi dari kolom database -> in the next session.

2.5. Penanganan Variabel dalam PL/SQL

Variabel dalam PL/SQL dapat digunakan dalam berbagai cara berikut :

- Deklarasi dan inialisasi variable pada bagian deklarasi (DECLARE)
- Menandai nilai baru pada bagian yang dapat dijalankan (executable).
- Suatu nilai dapat dilewatkan melalui variable sebagai parameter dari suatu sub program
- Hasil dari suatu proses dapat ditampilkan melalui variable output.

2.6. Tipe-tipe Variabel

Tipe-tipe variable dalam PL/SQL :

- Scalar
- Composite
- Reference
- LOB (large object)

Tipe Data	Contoh
BOOLEAN	TRUE
DATE	25-JAN-01
BLOB	Data foto
LONG	Data percakapan
NUMBER	25610.08
BFILE	Data film
VARCHAR2	Nama kota

2.7. Penggunaan Variabel iSQL*PLUS dengan Blok PL/SQL

2.8. Deklarasi Variabel PL/SQL

Deklarasi Variabel dalam PL/SQL, mempunyai sintak umum berikut :

```
Identifier [CONSTANT] datatype [NOT NULL]
[:= | DEFAULT expr];
```

Contoh deklarasi variable :

```
DECLARE
  V_hiredate      DATE;
  V_deptno       NUMBER(2) NOT NULL :=10;
  V_location     VARCHAR2(20) := 'Atlanta';
  C_comm.        CONSTANT NUMBER := 1400;
```

2.9. Petunjuk dalam Pendeklarasian Variabel

2.10. Aturan Penamaan

Variabel diberi nama sesuai dengan aturan, misal *v_name* merepresentasikan suatu variable, jadi nama dari variable selalu diawali dengan v diikuti dengan underscore (_) lalu nama variabelnya. Sedangkan *c_comm.* merepresentasikan variable konstanta, jadi nama variable konstanta selalu diawali dengan c.

Dua variable boleh memiliki nama yang sama, asalkan terletak pada blok yang berbeda.

Nama variable (identifier) boleh sama dengan nama kolom dari suatu table. Misal pada contoh berikut :

```
DECLARE
  Employee_id NUMBER(6);
BEGIN
  SELECT employee_id
  INTO employee_id
  FROM employees
  WHERE last_name = 'Kochhar';
END;
/
```

2.11. Inisialisasi Variabel dan Keyword

Sintak untuk inisialisasi variable :

Identifier := *expr*;

Contoh inisialisasi variable : v_hiredate := '01-JAN-01';

Inisialisasi variable dapat juga dengan menggunakan keyword *DEFAULT* . Untuk contoh sebelumnya bisa ditulis, sebagai berikut :

v_hiredate DEFAULT '01-JAN-01';

Jika suatu variable harus berisi suatu nilai (tidak boleh kosong atau NULL), maka digunakan keyword NOT NULL.

Contoh : v_city VARCHAR2(30) NOT NULL := 'Oxford';

2.12. Tipe Data Skalar

Variabel yang memiliki tipe data Skalar, memiliki nilai tunggal. Tipe data yang termasuk skalar, yaitu number, character, date, boolean.

Contoh deklarasi variable skalar :

```

DECLARE
  V_job          VARCHAR2(9);
  V_count        BINARY_INTEGER := 0;
  V_total_sal    NUMBER(9,2) :=0;
  V_orderdate    DATE := SYSDATE + 7;
  C_tax_rate     CONSTANT NUMBER(3,2):=8.25;
  V_valid        BOOLEAN NOT NULL := TRUE;

```

2.13. Tipe Data Base Skalar

2.14. Deklarasi Variabel Skalar

2.15. Atribut %TYPE

Atribut %TYPE digunakan untuk mendeklarasikan variable yang memiliki tipe data yang sama dengan definisi kolom database.

2.16. Deklarasi Variabel dengan Atribut %TYPE

Sintak untuk deklarasi variable dengan atribut %TYPE :

```
Identifier table.column_name%TYPE;
```

Contoh penggunaan :

```

V_name          employees.last_name%TYPE;
V_min_balance   v_balance%TYPE :=10;

```

2.17. Deklarasi Variabel Boolean

2.18. Tipe Data Komposit dan LOB (Large Object)

Tipe data Composite memiliki komponen internal yang dapat dimanipulasi secara individual. Tipe data Composite dikenal dengan istilah *collection*, terdiri dari : TABLE, RECORD, NESTED TABLE, dan VARRAY.

Sedangkan tipe data LOB (*large objects*) dapat digunakan untuk menyimpan data yang tidak terstruktur (misal : teks, grafik, image, video dan suara). LOB juga men-support *random access* terhadap data.

Yang termasuk tipe data LOB : CLOB, BLOB, BFILE, NCLOB.

2.19. Variable Bind

Bind variable merupakan variable yang dideklarasikan dalam *host environment*. Variabel bind dapat digunakan untuk melewati suatu nilai yang dijalankan diluar blok PL/SQL.

Cara deklarasi bind variable :

```
VARIABLE nama_var TIPE_DATA
```

Untuk melakukan referensi terhadap *bind variable* digunakan tanda titik dua (:)

Contoh penggunaan *bind variable* :

```
VARIABLE g_salary NUMBER
BEGIN
  SELECT salary
  INTO :g_salary
  FROM employees
  WHERE employee_id = 178;
END;
/
PRINT g_salary
```

```
VARIABLE g_monthly_sal NUMBER
DEFINE p_annual_sal = 50000
SET VERIFY OFF
DECLARE
  v_sal NUMBER(9,2) := &p_annual_sal;
BEGIN
  :g_monthly_sal=v_sal/12;
END;
/
PRINT g_monthly_sal
```

2.20. Referensi ke Variabel Non PL/SQL

2.21. DBMS_OUTPUT.PUTLINE

Bind variable selain ditampilkan dengan *PRINT*, juga dapat ditampilkan dalam blok PL/SQL , dengan menggunakan fungsi standart yang disediakan oleh Oracle yaitu *DBMS_OUTPUT.PUT_LINE*. Untuk menggunakan fungsi ini, *SERVEROUTPUT* harus diset *ON*.

```
SET SERVEROUTPUT ON
DEFINE p_annual_sal 50000
DECLARE
  v_sal NUMBER(9,2) := &p_annual_sal;
BEGIN
  v_sal:=v_sal/12;
  DBMS_OUTPUT.PUT_LINE('The monthly salary is ' || to_char(v_sal));
END;
/
```

2.22. Ringkasan

- Blok dalam PL/SQL terdiri dari bagian-bagian, sebagai berikut :
 - Deklarasi
 - Bagian yang dapat dijalankan (executable)
 - Bagian penanganan kesalahan (exception handling)

- Ada 3 (tiga) tipe block dalam PL/SQL, yaitu Anonymous, Procedure dan Function
- PL/SQL Identifier :
 - Didefinisikan pada bagian deklarasi
 - Merupakan tipe data scalar, composite, reference atau LOB
 - Dapat menggunakan struktur dari variable lain atau object database yang lain
- Variabel yang dideklarasikan pada lingkungan eksternal semisal SQL*PLUS disebut dengan *host variable*
- DBMS_OUTPUT.PUT_LINE dapat digunakan untuk menampilkan data dalam blok PL/SQL

2.23. Latihan Soal

1. Evaluasi tiap-tiap deklarasi variable berikut ini. Mana yang legal dan mana yang illegal !
 - a. DECLARE
v_id NUMBER(4);
 - b. DECLARE
v_x, v_y, v_z VARCHAR2(10);
 - c. DECLARE
v_birthdate DATE NOT NULL;
 - d. DECLARE
V_in_stock BOOLEAN := 1;
2. Pada tiap *assignment* berikut, yang manakah statement yang valid dan mana yang tidak valid, kemudian nyatakan tipe datanya !
 - a. v_days_to_go := v_due_date - SYSDATE;
 - b. v_sender := USER || ':' || TO_CHAR(v_dept_no);
 - c. v_sum := \$100,000 + \$250,000
 - d. v_flag := TRUE;
 - e. v_n1 := v_n2 > (2 * v_n3);
 - f. v_value := NULL;
3. Buat anonymous block untuk menampilkan kata 'I love Oracle', buat dengan dua cara. Cara pertama menggunakan bind variable dan perintah PRINT untuk menampilkannya. Cara kedua menggunakan DBMS_OUTPUT.PUT_LINE
4. Buat kode PL/SQL yang mendeklarasikan dua variable dan masing-masing nilainya berikut ini :

Variabel	Value
V_CHAR	'42 adalah sebuah angka'
V_NUM	dua karakter pertama dari variable V_CHAR yaitu 42