

BAB 3 : Pemrograman dengan PL/SQL**Tujuan Pembelajaran :**

- Memahami bagian *executable* dari PL/SQL block
- Menggunakan identifier secara benar
- Memahami aturan pada blok bersarang (*nested block*)

3.1. Sintak PL/SQL

Baris-baris statement dalam PL/SQL merupakan kumpulan karakter yang dapat diklasifikasikan sebagai berikut :

- Delimiter (*simple* dan *compound symbol*)
- Identifiers, termasuk *reserved words* (*keywords*)
- Literal
- Komentar

3.2. Identifier

Identifier digunakan untuk memberi nama bagian PL/SQL, meliputi konstanta, varabel, exception, cursor, cursor variable, subprogram dan packages.

Aturan penamaan identifier :

- Maksimal 30 karakter
- Harus dimulai dengan karakter alfabet
- Dapat terdiri dari angka, symbol dollar, underscores (_), dan tanda bilangan (+ atau -)
- Tidak boleh ada spasi, tanda slash (/), hyphen (-), ampersand (&).
- Tidak memiliki nama yang sama dengan kolom database
- Bukan merupakan *reserved word* (*keyword*)

3.3. Literal dan Slash

- **Literal** terdiri dari karakter dan numerik.
- Karakter literal harus diapit oleh tanda petik. Contoh penggunaan literal :
 - v_name := 'Daisy Rahmania Syarif';
- Numerik literal dapat berupa suatu nilai atau notasi scientific
- Slash (/) digunakan untuk menjalankan PL/SQL blok dalam file script atau SQL*PLUS prompt.

3.4. Komentar

Komentar yang terdiri dari satu baris, menggunakan tanda minus dua kali (--). Sedangkan komentar yang terdiri dari lebih satu baris menggunakan symbol /* dan */ untuk mengawali dan mengakhiri komentar.

3.5. Fungsi SQL dalam PL/SQL

Sebagian besar fungsi SQL juga valid pada PL/SQL. Sedangkan fungsi yang hanya valid pada SQL tapi tidak pada PL/SQL adalah fungsi :

- DECODE
- Fungsi group seperti AVG, MIN, MAX, COUNT, SUM, dll.

Contoh fungsi SQL dalam PL/SQL :

- v_mailing_address := v_name || CHR(10) || v_address || CHR(10) || v_zip;
- v_ename := LOWER(v_ename);

3.6. Konversi Tipe Data

Fungsi untuk mengkonversi tipe data : TO_CHAR, TO_DATE dan TO_NUMBER.

Contoh konversi tipe data :

```
DECLARE
    v_date DATE := TO_DATE('12-JAN-2001', 'DD-MON-YYYY');
BEGIN
    .....
```

3.7. Blok Bersarang dan Ruang Lingkup Variabel

Blok dalam PL/SQL ditandai dengan **BEGIN END;**

Yang disebut dengan **blok bersarang** atau **nested block** adalah Blok (BEGIN ... END;) yang terletak di dalam blok yang lain.

Yang perlu diperhatikan dalam *nested block* adalah ruang lingkup variabel.

Contoh ruang lingkup variabel dalam *nested block* :

```
.....
x BINARY_INTEGER;
BEGIN                                     Scope of x
    .....
    DECLARE
        y NUMBER;
        BEGIN                             Scope of y
            y := x;
            END;
        .....
    END;
```

3.8. Penentuan Ruang Lingkup dengan <<>>

Suatu blok dapat ditandai dengan menggunakan **<<>>** untuk menentukan ruang lingkup dari suatu variabel. Seperti pada contoh berikut ini :

```
<<outer>>
DECLARE
    birthdate DATE;
BEGIN
    DECLARE
        birthdate DATE;
        BEGIN
            .....
            outer.birthdate := TO_DATE('03-AUG-1976', 'DD-MON-YYYY');

        END;
        .....
    END;
```

3.9. Operator dalam PL/SQL

Macam-macam operator dalam PL/SQL :

- Logika
- Aritmatika
- Concatenation (penyambunga)
- Paratheses (kurung buka, tutup) untuk mengatur urutan dari suatu operasi
- Eksponensial (**)

Contoh penggunaan operator dalam PL/SQL :

- Menaikkan nilai counter :
v_count := v_count +1;
- Mencari nilai Boolean
v_equal := (v_n1 = v_n2);
- Melakukan validasi apakah variable nomer employee (v_empno) memiliki suatu nilai
v_valid := (v_empno IS NOT NULL);

3.10. Ringkasan

- Telah dipelajari sintak dari bagian-bagian PL/SQL
- Memahami struktur blok PL/SQL dan blok bersarang (*nested block*)
- Menggunakan identifier secara benar
- Memahami aturan pada blok bersarang (*nested block*)
- PL/SQL Programming :
 - Function
 - Konversi tipe data
 - Operator

3.11. Latihan Soal

1. Perhatikan kode PL/SQL berikut :

```

DECLARE
  v_weight    NUMBER(3) := 600;
  v_message   VARCHAR2(255) := 'Product 10012';
BEGIN
  /* Subblock */
  DECLARE
    v_weight NUMBER(3) := 1;
    v_message VARCHAR2(255) := 'Product 11001';
    v_newlocn VARCHAR2(50) := 'Europe';
  BEGIN
    v_weight := v_weight +1;
    v_newlocn := 'Western ' || v_newlocn;

1  _____

    END;
    v_weight := v_weight +1;
    v_message := v_message || ' is in stock';
    v_newlocn := 'Western ' || v_newlocn;

2  _____

  END;
/

```

kemudian tentukan **nilai** dan **tipe data** dari variable berikut ini :

- a. v_weight pada posisi 1
 - b. v_newlocn pada posisi 1
 - c. v_weight pada posisi 2
 - d. v_message pada posisi 2
 - e. v_newlocn pada posisi 2
2. Buat blok PL/SQL yang dapat menerima dua buah bilangan yang berasal dari dua variable .
 - a. Gunakan perintah DEFINE

- b. Kemudian lewatkan dua nilai dari variable yang didefinisikan dengan perintah DEFINE tersebut ke block PL/SQL yang menghitung nilai pembagian bilangan pertama dibagi bilangan kedua, hasilnya disimpan dalam suatu variable yang akan ditampilkan dengan menggunakan DBMS_OUTPUT.PUT_LINE.
Buat programnya !