

# **LAPORAN PRAKTIKUM BASIS DATA LANJUT**

## **PRAKTIKUM KE – 4 (PLSQL Bab 1 dan 2)**



Nama : Muhammad Raihan

NRP : 3123500056

Dosen Pengajar : Wiratmoko Yuwono ST,.MT

**PROGRAM STUDI D3 TEKNIK INFORMATIKA  
POLITEKNIK ELEKTRONIKA NEGERI SURABAYA  
(PENS) TAHUN 2024**

## A. Tugas

1. Pilih blok PL/SQL yang dapat dijalankan, lalu praktekkan ke aplikasi SQL Developer

?

a. BEGIN END;

b. DECLARE

amount INTEGER(10);

END;

c. DECLARE

BEGIN

END;

d. DECLARE

amount INTEGER(10);

BEGIN DBMS\_OUTPUT.PUT\_LINE(amount);

END;

2. Buat dan eksekusi anonymous blok PL/SQL dengan output “Hello World NRP dan

Nama Anda”, simpan ke NRPanda.sql

3. Identify valid and invalid identifier names:

a. today

b. last\_name

c. today's\_date

d. Number\_of\_days\_in\_February\_this\_year

e. Isleap\$year

f. #number

g. NUMBER#

h. number1to7

4. Identify valid and invalid variable declaration and initialization:

a. number\_of\_copies PLS\_INTEGER;

b. printer\_name constant VARCHAR2(10);

c. deliver\_to VARCHAR2(10):=Johnson;

d. by\_when DATE:= SYSDATE+1;

5. Examine the following anonymous block and choose the appropriate statement.

```

SET SERVEROUTPUT ON
DECLARE
fname VARCHAR2(20);
lname VARCHAR2(15) DEFAULT 'fernandez';
BEGIN
DBMS_OUTPUT.PUT_LINE( FNAME || ' ' ||lname);
END;
/

```

- a. The block will execute successfully and print 'fernandez'
- b. The block will give an error because the fname variable is used without initializing.
- c. The block will execute successfully and print 'null fernandez'
- d. The block will give an error because you cannot use the DEFAULT keyword to initialize a variable of type VARCHAR2.
- e. The block will give an error because the variable FNAME is not declared.

6. Create an anonymous block. In iSQL\*Plus, load the script lab\_01\_02\_soln.sql,

which you created in question 2 of practice 1.

a. Add a declarative section to this PL/SQL block. In the declarative section declare the following variables:

1. Variable today of type DATE. Initialize today with SYSDATE.
2. Variable tomorrow of type today. Use %TYPE attribute to declare this variable.

b. In the executable section initialize the variable tomorrow with an expression

which calculates tomorrow's date (add one to the value in today). Print the value of today and tomorrow after printing 'Hello World'

c. Execute and save this script as lab\_02\_04\_soln.sql. Sample output is shown below.

```

Hello World
TODAY IS : 12-JAN-04
TOMORROW IS : 13-JAN-04
PL/SQL procedure successfully completed.

```

7. Edit the lab\_02\_04\_soln.sql script.

a. Add code to create two bind variables.

Create bind variables basic\_percent and pf\_percent of type NUMBER.

b. In the executable section of the PL/SQL block assign the values 45 and 12 to

basic\_percent and pf\_percent respectively.

c. Terminate the PL/SQL block with “/” and display the value of the bind variables

by using the PRINT command.

d. Execute and save your script file as lab\_02\_05\_soln.sql. Sample output is shown below.

```
Hello World
TODAY IS : 12-JAN-04
TOMORROW IS : 13-JAN-04
PL/SQL procedure successfully completed.
```

## Jawaban dan Solusi

1a.

```
1 v BEGIN
2 END;
```

```
ORA-06550: line 2, column 1:
PLS-00103: Encountered the symbol "END" when expecting one of the following:
```

```
( begin case declare exit for goto if loop mod null pragma
raise return select update while with <an identifier>
<a double-quoted delimited-identifier> <a bind variable> <<
continue close current delete fetch lock insert open rollback
savepoint set sql execute commit forall merge pipe purge
json_exists json_value json_query json_object json_array
```

Untuk bagian yang a saat dijalankan menghasilkan menghasilkan error karena didalam blok tidak ada yang dapat dilakukan.

1b.

```

1 v DECLARE
2   amount INTEGER(10);
3   END;

```

ORA-06550: line 3, column 1:

PLS-00103: Encountered the symbol "END" when expecting one of the following:

```

begin function pragma procedure subtype type <an identifier>
<a double-quoted delimited-identifier> current cursor delete
exists prior

```

Untuk bagian b menghasilkan error karena query tidak ada BEGIN dan variable yang telah dideklarasikan tidak ditambahkan dengan logika yang dijalankan.

1c.

```

1 v DECLARE
2   BEGIN
3   END;

```

ORA-06550: line 3, column 1:

PLS-00103: Encountered the symbol "END" when expecting one of the following:

```

( begin case declare exit for goto if loop mod null pragma
raise return select update while with <an identifier>
<a double-quoted delimited-identifier> <a bind variable> <<
continue close current delete fetch lock insert open rollback
savepoint set sql execute commit forall merge pipe purge
json_exists json_value json_query json_object json_array

```

Untuk bagian c menghasilkan error karena tidak ada deklarasi variable dalam DECLARE dan tidak ditambah dengan logika yang dapat dijalankan

1d.

```

1 v DECLARE
2   amount INTEGER(10);
3 v BEGIN
4   DBMS_OUTPUT.PUT_LINE(amount);
5   END;|

```

Statement processed.

Untuk bagian d tidak menghasilkan error karena bagian DECLARE telah mendeklarasikan variable dan bagian BEGIN memberikan perintah untuk mencetak.

Dalam query tersebut tetapi kurang benar dalam inisialisasi variable yang benar adalah:

```
amount INTEGER := 10;
```

Maka dapat menampilkan hasil output

```

1 v DECLARE
2   amount INTEGER := 10;
3 v BEGIN
4   DBMS_OUTPUT.PUT_LINE(amount);
5   END;|

```

Statement processed.

10

2.

```

1 v DECLARE
2   nrp CHAR(10) := '3123500056';
3   nama VARCHAR2(50) := 'Muhammad Raihan';
4 v BEGIN
5   DBMS_OUTPUT.PUT_LINE('Hello World ' || nrp || ' ' || nama);
6   END;

```

Statement processed.

Hello World 3123500056 Muhammad Raihan

Pertama menginisialisasi variable nrp dan nama dalam bagian DECLARE. Lalu menampilkan output tersebut dengan perintah DBMS\_OUTPUT.PUT.LINE()

3.

- A. Today = valid, dapat digunakan sebagai variable
- B. Last\_name = valid, dapat digunakan sebagai variable
- C. today's\_date = invalid, karena ada tanda petik satu ( ' )
- D. Number\_of\_days\_in\_February\_this\_year = valid
- E. Isleap\$year = valid
- F. #number = invalid, karena ada tanda pagar (#) yang dimulai pada awal variable
- G. NUMBER# = valid
- H. number1to7 = valid

4.

- A. number\_of\_copies PLS\_INTEGER; = valid
- B. printer\_name constant VARCHAR2(10); = invalid, karena printer\_name di deklarasikan sebagai constant dan perlu diberikan inisialisasi
- C. deliver\_to VARCHAR2(10):=Johnson; = invalid, karena isi dari variabel "Johnson" tidak di masukkan dalam tanda petik 2
- D. by\_when DATE:= SYSDATE+1; = valid

5.

```
SET SERVEROUTPUT ON
DECLARE
fname VARCHAR2(20);
lname VARCHAR2(15) DEFAULT 'fernandez';
BEGIN
DBMS_OUTPUT.PUT_LINE( FNAME || ' ' ||lname);
END;
```

```

1 v SET SERVEROUTPUT ON
2 DECLARE
3     fname VARCHAR2(20);
4     lname VARCHAR2(15) DEFAULT 'fernandez';
5 v BEGIN
6     DBMS_OUTPUT.PUT_LINE( FNAME || ' ' || lname);
7 END;|

```

Unsupported Command

Statement processed.  
fernandez

- A. The block will execute successfully and print 'fernandez' = Iya, berhasil mencetak nama 'fernandez'
- B. The block will give an error because the fname variable is used without Initializing. = Tidak, variable hanya dibuat tetapi tidak menyebabkan error
- C. The block will execute successfully and print 'null fernandez' = tidak ada saat dilakukan running karena variable fname dalam bentuk string
- D. The block will give an error because you cannot use the DEFAULT keyword to initialize a variable of type VARCHAR2 = tidak, menggunakan default tidak menghasilkan error
- E. The block will give an error because the variable FNAME is not declared = tidak, FNAME sudah di deklarasikan dalam variable fname karena tidak sensitive pada capslock

6.

```

1 v DECLARE
2     v_today date := SYSDATE;
3     v_tomorrow v_today%TYPE := v_today + 1;
4 v BEGIN
5     DBMS_OUTPUT.PUT_LINE('Hello world');
6     DBMS_OUTPUT.PUT_LINE('TODAY IS: ' || v_today);
7     DBMS_OUTPUT.PUT_LINE('TOMORROW IS: ' || v_tomorrow);
8 END;

```

Statement processed.  
Hello world  
TODAY IS: 14-SEP-24  
TOMORROW IS: 15-SEP-24



7.

```
VARIABLE basic_percent NUMBER;  
VARIABLE pf_percent NUMBER;  
SET AUTOPRINT ON  
DECLARE  
    v_today date := SYSDATE;  
    v_tomorrow v_today%TYPE := v_today + 1;  
BEGIN  
    :basic_percent := 45;  
    :pf_percent := 12;  
    DBMS_OUTPUT.PUT_LINE('Hello world');  
    DBMS_OUTPUT.PUT_LINE('TODAY IS: '|| v_today);  
    DBMS_OUTPUT.PUT_LINE('TOMORROW IS: '|| v_tomorrow);  
END;
```

#### B. Analisa

- Pada nomer 5 terdapat error dari code yang dijalankan. Penyebab error berasal dari (SET SERVEROUTPUT ON).
- Pada soal no 7 tidak dapat dilakukan dalam Live sql oracle karena adanya perintah yang membutuhkan SQL PLUS

#### **Kesimpulan**

- DECLARE digunakan untuk mendeklarasikan variable
- BEGIN digunakan untuk memulai sebuah perintah dalam blok
- END digunakan sebagai penutup dari BEGIN
- := itu sama dengan =
- || itu sama dengan +