

LAPORAN PRAKTIKUM
Manajemen dan Administrasi Basis Data
Week 8 : PL/SQL (Procedure)

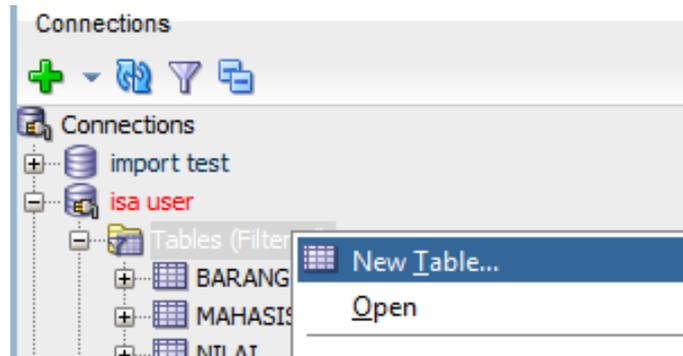


MUHAMMAD FARIS ISA
D4LJ – Teknik Informatika
3122640005

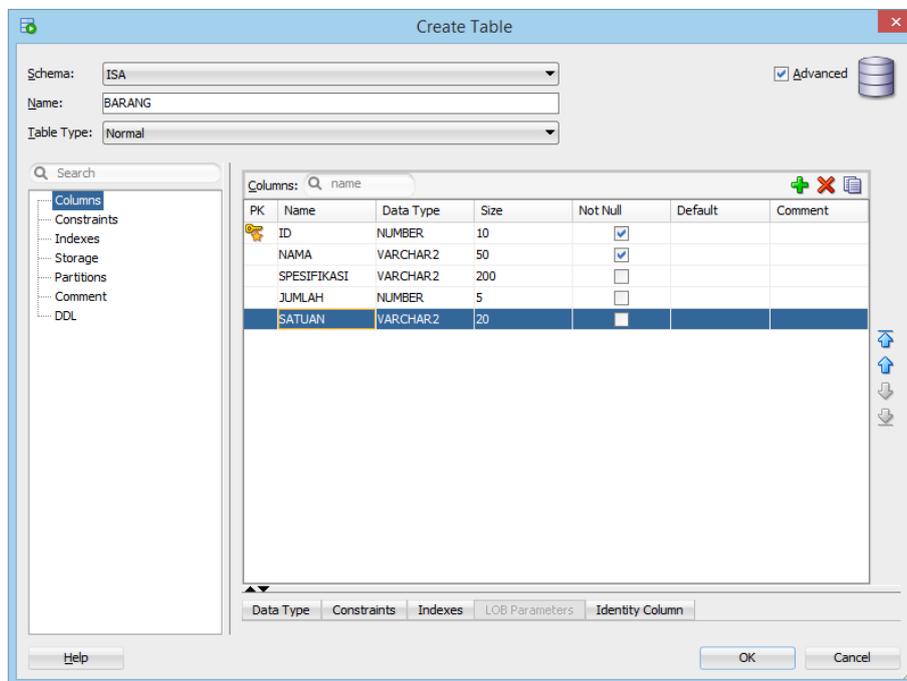
POLITEKNIK ELEKTRONIKA NEGERI SURABAYA
2022

1. Membuat Tabel Barang

- Pembuatan table berdasarkan yang diminta dengan menggunakan klik kanan pada table. Lalu klik Create New.



- Setelah itu masukkan field seperti yang diminta.



- Jika sudah berhasil bisa dilihat pada dibawah bahwa tabel telah dibuat.

Oracle SQL Developer : Table ISA.BARANG@isa user

Help

Columns | Data | Model | Constraints | Grants | Statistics | Triggers | Flashback | Dependencies | Details | Partitions | Indexes | SQL

Actions...

COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1 ID	NUMBER (10, 0)	No	(null)	1 (null)	
2 NAMA	VARCHAR2 (50 BYTE)	No	(null)	2 (null)	
3 SPESIFIKASI	VARCHAR2 (200 BYTE)	Yes	(null)	3 (null)	
4 JUMLAH	NUMBER (5, 0)	Yes	(null)	4 (null)	
5 SATUAN	VARCHAR2 (20 BYTE)	Yes	(null)	5 (null)	

2. Membuat script procedure untuk menambahkan, mengubah dan menghapus data pada tabel Barang.

- Membuat script untuk procedure.

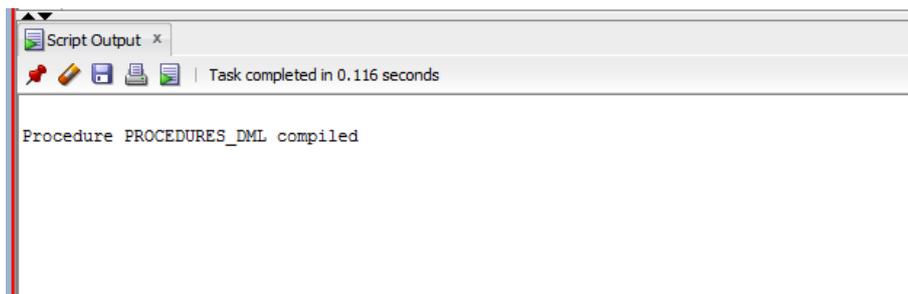
```
CREATE OR REPLACE PROCEDURE PROCEDURES_DML
(
  PROCEDURE_TYPE IN VARCHAR2
, BARANG_ID IN BARANG.ID%TYPE
, BARANG_NAMA IN BARANG.NAMA%TYPE DEFAULT ''
, BARANG_SPESIFIKASI IN BARANG.SPESIFIKASI%TYPE DEFAULT NULL
, BARANG_JUMLAH IN BARANG.JUMLAH%TYPE DEFAULT NULL
, BARANG_SATUAN IN BARANG.SATUAN%TYPE DEFAULT NULL
) AS
BEGIN
  CASE LOWER(PROCEDURE_TYPE)
    WHEN 'insert' THEN
      INSERT INTO
        BARANG
      VALUES (
        BARANG_ID,
        BARANG_NAMA,
        BARANG_SPESIFIKASI,
        BARANG_JUMLAH,
        BARANG_SATUAN
      );
    WHEN 'update' THEN
      UPDATE
```

```

        BARANG
    SET
        NAMA = BARANG_NAMA,
        SPESIFIKASI = BARANG_SPESIFIKASI,
        JUMLAH = BARANG_JUMLAH,
        SATUAN = BARANG_SATUAN
    WHERE
        ID = BARANG_ID;
    WHEN 'delete' THEN
        DELETE FROM
            BARANG
        WHERE
            ID = BARANG_ID;
    ELSE
        DBMS_OUTPUT.PUT_LINE('PROCEDURE Value Not Found.');
```

END CASE;
END PROCEDURES_DML;

- Selanjutnya lakukan run script dan lihat apakah berhasil, jika berhasil maka akan muncul tampilan seperti dibawah.



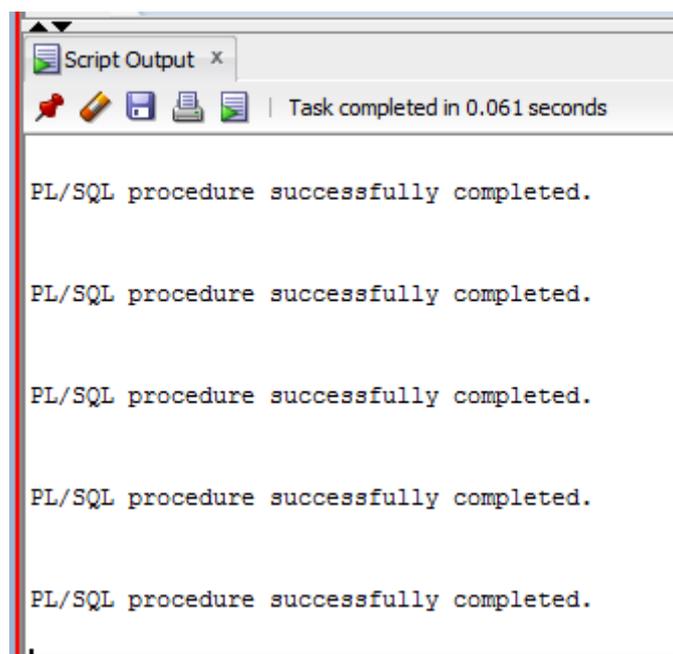
- Dan bisa dilihat juga pada bagian kanan Connections, bahwa terdapat Procedure yang sebelumnya telah dibuat.



- Untuk melakukan pengujian pada procedure dapat melakukan seperti dibawah ini.

```
EXECUTE PROCEDURES_DML('insert', 1, 'ThinkPad X1 Nano',  
'Processor Up to 11th Gen Intel®Core™ i7', 5, 'pcs');  
EXECUTE PROCEDURES_DML('insert', 2, 'ThinkPad L13 Yoga Gen 3',  
'Processor : Up to Intel vPro® with 12th Gen Intel® Core™ i7 U15  
processors', 2, 'pcs');  
EXECUTE PROCEDURES_DML('insert', 3, 'ThinkBook 14p Gen 3',  
'Processor : Up to AMD Ryzen™ 5 6000 Series Mobile Processors', 2,  
'pcs');  
EXECUTE PROCEDURES_DML('insert', 4, 'ROG Zephyrus G15',  
'Processor : AMD Ryzen™ 7 6800HS Mobile Processor', 9, 'pcs');  
EXECUTE PROCEDURES_DML('insert', 5, 'ROG Flow X16', 'Processor :  
AMD Ryzen™ 9 6900HS Mobile Processor', 1, 'pcs');
```

- Lakukan run script, jika berhasil maka data akan bertambah kedalam table dan bisa dilihat bahwa hasil run berhasil.



The screenshot shows a 'Script Output' window with a title bar 'Script Output x'. Below the title bar, there are icons for a red pushpin, a pencil, a save icon, a print icon, and a refresh icon. To the right of these icons, it says 'Task completed in 0.061 seconds'. The main area of the window contains five lines of text, each stating 'PL/SQL procedure successfully completed.'.

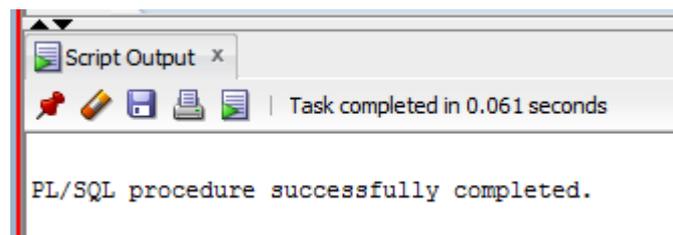
- Dapat dilihat pada tabel bahwa terisi seperti berikut ini.

	NAMA	SPESIFIKASI	JUMLAH	SATUAN
1	1 ThinkPad X1 Nano	ProcessorUp to 11th Gen Intel®Core™ i7	5	pcs
2	2 ThinkPad L13 Yoga Gen 3	Processor : Up to Intel vPro® with 12th Gen I...	2	pcs
3	3 ThinkBook 14p Gen 3	Processor : Up to AMD Ryzen™ 5 6000 Series Mo...	2	pcs
4	4 ROG Zephyrus G15	Processor : AMD Ryzen™ 7 6800HS Mobile Processor	9	pcs
5	5 ROG Flow X16	Processor : AMD Ryzen™ 9 6900HS Mobile Processor	1	pcs

- Jika ingin melakukan pengujian mengubah data table maka coba dengan script ini.

```
EXECUTE PROCEDURES_DML('update', 5, 'ROG Flow X16', 'Processor :
AMD Ryzen™ 7 6800HS Mobile Processor', 5, 'pcs');
```

- Jika berhasil maka dapat dilihat pada run bahwa berhasil serta data pada tabel berubah.

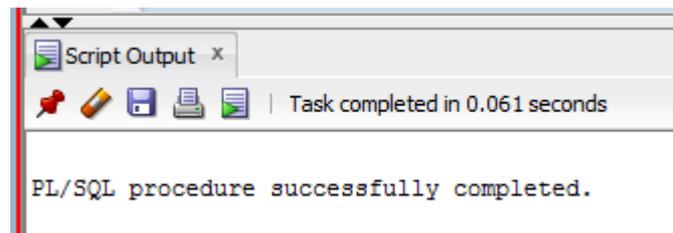
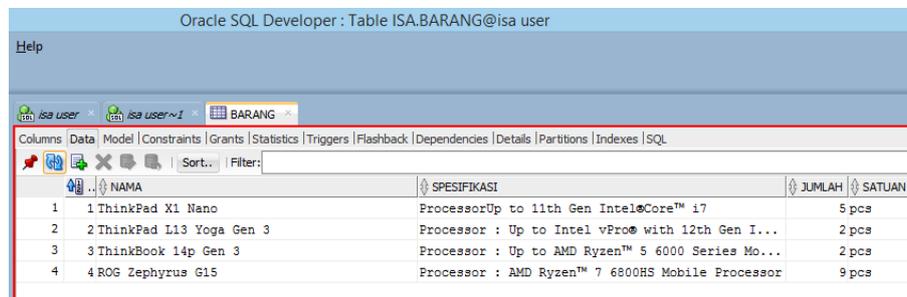


	NAMA	SPESIFIKASI	JUMLAH	SATUAN
1	1 ThinkPad X1 Nano	ProcessorUp to 11th Gen Intel®Core™ i7	5	pcs
2	2 ThinkPad L13 Yoga Gen 3	Processor : Up to Intel vPro® with 12th Gen I...	2	pcs
3	3 ThinkBook 14p Gen 3	Processor : Up to AMD Ryzen™ 5 6000 Series Mo...	2	pcs
4	4 ROG Zephyrus G15	Processor : AMD Ryzen™ 7 6800HS Mobile Processor	9	pcs
5	5 ROG Flow X16	Processor : AMD Ryzen™ 7 6800HS Mobile Processor	5	pcs

- Jika ingin melakukan pengujian hapus data table dapat menggunakan script dibawah ini.

```
EXECUTE PROCEDURES_DML('delete', 5);
```

- Untuk mengecek apakah berhasil atau tidak, dapat melihat pada Script Output dan pada tabel apakah berhasil terhapus.

The image shows a screenshot of the Oracle SQL Developer interface. The title bar indicates 'Oracle SQL Developer: Table ISA.BARANG@isa user'. The main window displays a table named 'BARANG' with the following data:

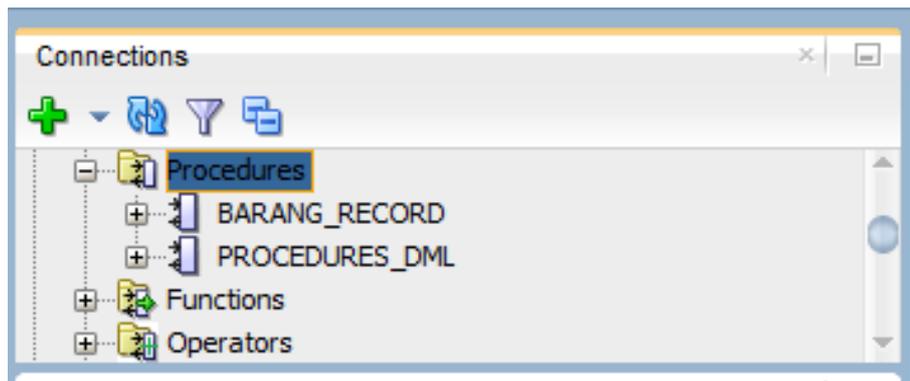
ID	NAMA	SPESIFIKASI	JUMLAH	SATUAN
1	1 ThinkPad X1 Nano	ProcessorUp to 11th Gen Intel®Core™ i7	5	pcs
2	2 ThinkPad L13 Yoga Gen 3	Processor : Up to Intel vPro® with 12th Gen I...	2	pcs
3	3 ThinkBook 14p Gen 3	Processor : Up to AMD Ryzen™ 5 6000 Series Mo...	2	pcs
4	4 ROG Zephyrus G15	Processor : AMD Ryzen™ 7 6800HS Mobile Processor	9	pcs

3. Membuat script procedure untuk menampilkan semua data pada tabel Barang.

- Membuat script seperti dibawah

```
create or replace PROCEDURE BARANG_RECORD
(
    RESULT OUT SYS_REFCURSOR
)
AS
BEGIN
    OPEN RESULT FOR
    SELECT ID, NAMA, SPESIFIKASI, JUMLAH, SATUAN
    FROM BARANG;
END BARANG_RECORD;
```

- Untuk mengetes apakah berhasil atau tidak scriptnya dapat menjalankan perintah run script dan lihat pada Script Output apakah berhasil atau tidak.

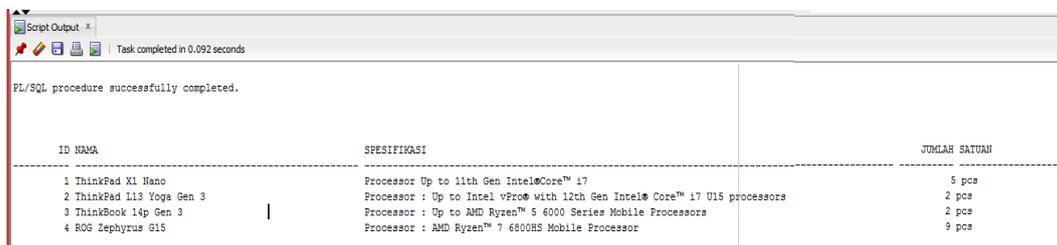


Dilihat bahwa pada prosedur terdapat prosedur baru yaitu BARANG_RECORD yang sebelumnya telah dibuat.

- Untuk menjalankan prosedur BARANG_RECORD, dapat menggunakan script dibawah ini :

```
VARIABLE X REFCURSOR;
EXECUTE BARANG_RECORD(:X);
PRINT X;
```

- Pengujian script diatas dapat dilakukan menggunakan perintah run script dan bisa dilihat pada Script Output hasilnya :

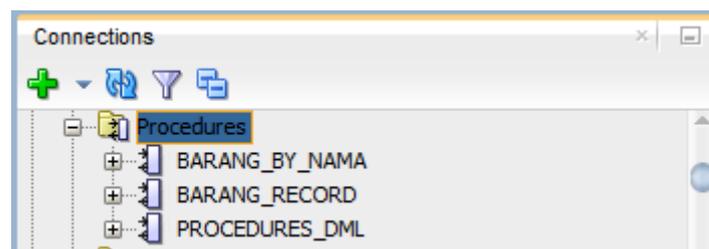
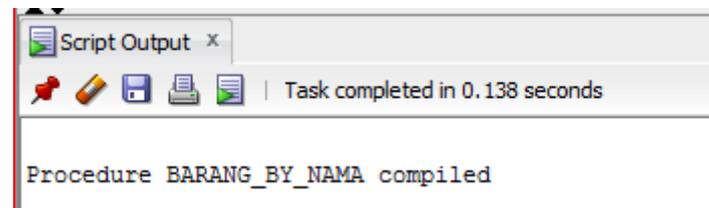


4. Membuat script procedure untuk untuk menampilkan semua data pada table barang dengan input kriteria pada field nama (input nama boleh sepenggal kata).

- Membuat script seperti dibawah

```
create or replace PROCEDURE BARANG_BY_NAMA
(
  NAMA_SEARCH IN BARANG.NAMA%TYPE,
  RESULT OUT SYS_REFCURSOR
)
AS
BEGIN
  OPEN RESULT FOR
  SELECT ID, NAMA, SPESIFIKASI, JUMLAH, SATUAN
  FROM BARANG
  WHERE NAMA LIKE '%' || NAMA_SEARCH || '%';
END BARANG_BY_NAMA;
```

- Untuk mengetes apakah berhasil atau tidak scriptnya dapat menjalankan perintah run script dan lihat pada Script Output apakah berhasil atau tidak.

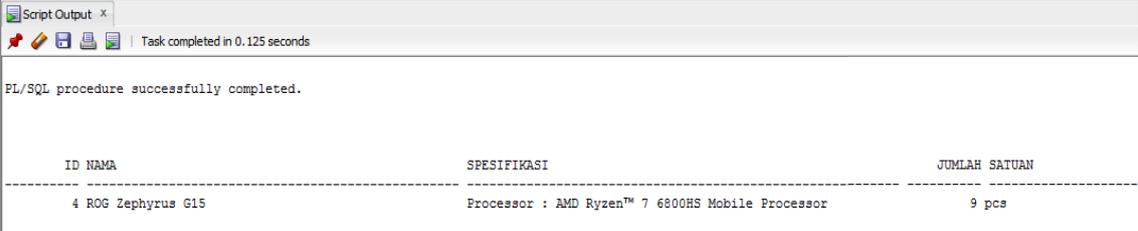


Dilihat bahwa pada procedur terdapat procedur baru yaitu BARANG_BY_NAMA yang sebelumnya telah dibuat.

- Untuk menjalankan procedur BARANG_BY_NAMA, dapat menggunakan script dibawah ini :

```
VARIABLE X REFCURSOR;
EXECUTE BARANG_BY_NAMA('ROG', :X);
PRINT X;
```

- Pengujian script diatas dapat dilakukan menggunakan perintah run script dan bisa dilihat pada Script Output hasilnya :



```
Script Output x
Task completed in 0.125 seconds

PL/SQL procedure successfully completed.

-----
ID NAMA                                SPESIFIKASI                                JUMLAH SATUAN
-----
4 ROG Zephyrus G15                      Processor : AMD Ryzen™ 7 6800HS Mobile Processor          9 pcs
-----
```

Kesimpulan

1. Menggunakan prosedur dapat membuat pengerjaan input, update, dan delete menjadi lebih cepat dan mudah.
2. Penggunaan Prosedur dapat memperingkas pemanggilan query, karena dapat dilakukan secara berulang.