

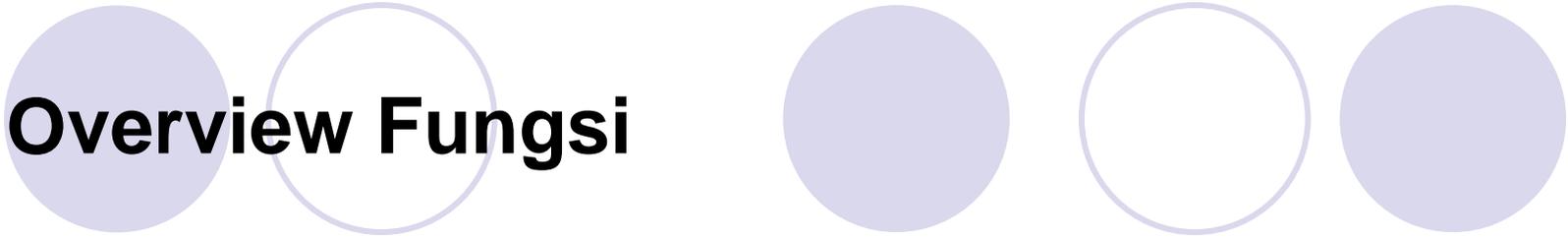
Pertemuan VI

FUNCTION

TUJUAN :

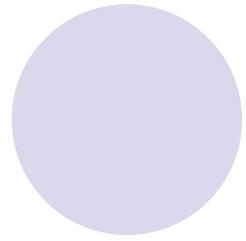
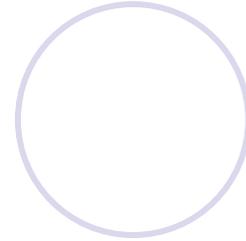
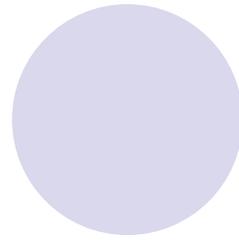
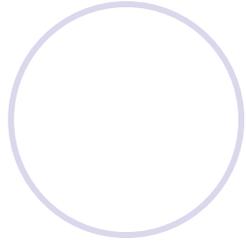
1. Mahasiswa mampu untuk menjelaskan penggunaan fungsi.
2. Mahasiswa mampu untuk membuat store of function
3. Mahasiswa mampu untuk menginvoke sebuah fungsi
4. Mahasiswa mampu untuk meremove sebuah fungsi
5. Mahasiswa mampu untuk membedakan antara fungsi dan prsoedur

1. Overview Fungsi



- Fungsi adalah sebuah blok PL/SQL yang memiliki nama yg mengembalikan sebuah nilai.
- Fungsi dapat disimpan dalam database sebagai sebuah object schema. Sehingga pada saat tertentu dapat diulang kembali proses eksekusinya.
- Fungsi dapat dipanggil sebagai bagian dari sebuah expresi.

Syntaks :



```
CREATE FUNCTION Function_name ([func_parameter [...]])  
RETURN type  
[characteristic ..] routine_body
```

Keterangan :

Parameter	Keterangan
<i>Function_name</i>	Nama dari fungsi
<i>Func_parameter</i>	Nama dari parameter yg nilainya akan dipassingkan ke fungsi, hanya mengenal IN parameter
<i>RETURN datatype</i>	Tipe data dari nilai RETURN yg merupakan output dari fungsi

Keterangan : (*cont*)

<i>Routine_body</i>	Characteristic: LANGUAGE SQL [NOT] DETERMINISTIC {CONTAINS SQL NO SQL READS SQL DATA MODIFIES SQL DATA } SQL SECURITY {DEFINER INVOKER } COMMENT 'string'
<i>Routine_body</i>	Isi dari fungsi

Contoh :

```
DELIMITER $$
DROP FUNCTION IF EXISTS `pbd`.`sfInventoryCustomer` $$
CREATE FUNCTION `pbd`.`sfInventoryCustomer`
  (p_inventory_id INT)
RETURNS INT(11)
BEGIN
  DECLARE v_customer_id INT;
  DECLARE EXIT HANDLER FOR NOT FOUND RETURN NULL;
  SELECT customer_id INTO v_customer_id FROM rental
  WHERE return_date IS NULL AND inventory_id =
  p_inventory_id;
  RETURN v_customer_id;
END $$
DELIMITER ;
select sfInventoryCustomer(2047);
```

2. Keuntungan dari User-Defined Function dalam Ekspresi SQL.

Keuntungan yang dapat diperoleh dari pembuatan user-defined function adalah:

- a) Mengizinkan perhitungan yang kompleks dengan SQL.
- b) Meningkatkan independensi data dengan proses analisis data yang kompleks dalam server daripada hanya menampilkan data ke dalam aplikasi.

2. Keuntungan dari User-Defined Function dalam Ekspresi SQL.(*cont*)

- c) Meningkatkan efisiensi query dengan membuat fungsi dalam query di bandingkan di dalam aplikasi.
- d) Memanipulasi tipe data baru dari data dengan melakukan encoding string dan menggunakan fungsi untuk menjalankan string.

3. Invoke Fungsi dalam ekspresi SQL.

Berikut adalah contoh bagaimana melakukan invoke sebuah fungsi dalam ekspresi SQL:

```
DELIMITER $$  
  
DROP FUNCTION IF EXISTS `pbd`.`spTax`$$  
  
CREATE FUNCTION `pbd`.`spTax` (p_value decimal)  
RETURNS decimal (10,2)  
BEGIN  
RETURN (p_value * 0.10);  
END $$  
  
DELIMITER ;
```

3. **Invoke Fungsi dalam ekspresi SQL. (*cont*)**

Pada contoh diatas, dibuat sebuah fungsi dengan nama spTax. Kemudian fungsi tersebut akan diinvoke pada statement SQL.

Pemanggilan fungsi pada ekspresi SQL dapat dilakukan dengan beberapa cara:

- a. Dipanggil pada perintah SELECT
- b. Dipanggil pada klausa kondisi dari WHERE dan HAVING.
- c. Dipanggil dari clause CONNECT BY, START WITH, ORDER BY dan GROUP BY.
- d. Pada klausa VALUES dari perintah INSERT.
- e. Pada klausa SET dari perintah UPDATE

3. Invoke Fungsi dalam ekspresi SQL. (*cont*)

Contoh pemanggilan fungsi spTAX pada ekspresi SQL:

```
SELECT customer_id, sptax(amount)
FROM payment
WHERE sptax(amount) >= (SELECT MAX
(sptax(amount)) FROM payment where staff_id = 31)
ORDER BY amount DESC;
```

3. **Invoke Fungsi dalam ekspresi SQL. (*cont*)**

Batasan / Restriksi Pemanggilan fungsi dari ekspresi SQL.

Agar sebuah fungsi dapat diakses melalui ekspresi SQL maka:

- a. Harus merupakan store function.
- b. Hanya menerima parameter IN
- c. Menerima hanya tipe data SQL yang valid, bukan spesifik PL/SQL seperti parameter.
- d. Mengembalikan tipe data SQL yang valid, bukan spesifik tipe data PL/SQL.

Catatan : hanya store function yang dapat dipanggil oleh ekspresi SQL, sedangkan store prosedur tidak bisa.

3. Invoke Fungsi dalam ekspresi SQL. (*cont*)

Restriksi pemanggilan fungsi dari ekspresi SQL adalah :

- a. Fungsi yang dipanggil dari ekspresi SQL tidak dapat mengandung statement DDL.
- b. Fungsi yang dipanggil dari statement UPDATE/DELETE pada table T tidak dapat mengandung DDL pada table yang sama dengan T.

3. Invoke Fungsi dalam ekspresi SQL. (*cont*)

- c. Fungsi yang dipanggil dari statement UPDATE/DELETE pada table T tidak boleh mengandung query di table yang sama.
- d. Saat memanggil fungsi dari SELECT, INSERT, UPDATE, DELETE, fungsi tidak boleh mengandung statement SQL untuk control statement (misal COMMIT), session control statement (SET ROLE) atau system control statement (seperti ALTER SYSTEM) dan tidak boleh mengeksekusi DDL statement seperti CREATE karena akan diikuti dengan otomatis commit.

4. Remove Function :

Untuk meremove store function digunakan syntax:

```
DROP FUNCTION function_nama;
```

5. Perbandingan antara Prosedur dan Function

Bagaimana membedakan antara *prosedur* dan *function* ?

a. Prosedur

Prosedur berisi sekumpulan statement yang dapat dieksekusi kemudian. Prosedur dapat tidak berisi parameter atau lebih dari satu parameter yang dapat digunakan untuk mentransfer ke dan dari prosedur, tetapi prosedur tidak mengembalikan nilai.

b. Fungsi

Fungsi dibuat jika hendak menghitung suatu nilai, yang harus dikembalikan ke calling environment. Fungsi dapat terdiri lebih dari satu parameter yang ditransfer dari calling environment. Fungsi harus mengembalikan satu nilai tunggal dan nilai dikembalikan melalui statement RETURN. Fungsi yang digunakan tidak boleh menggunakan parameter mode OUT atau IN OUT.

5. Perbandingan antara Prosedur dan Function (*cont*)

PROCEDURE	FUNCTION
Dijalankan sebagai statement PL/SQL	Diinvoke sebagai bagian dari ekspresi
Tidak mengandung klausa RETURNS pada header	Harus berisi klausa RETURNS pada header
Dapat mengembalikan none, satu atau lebih nilai	Harus mengembalikan sebuah nilai tunggal
Dapat berisi RETURN statement	Harus berisi minimal satu RETURN statement