

MODUL V

PROCEDURE

TUJUAN

Setelah menyelesaikan modul ini, anda diharapkan dapat :

1. Mengenalkan mahasiswa tentang konsep *procedure*/prosedur.
2. Mengenalkan mahasiswa tentang penggunaan prosedur dalam basis data

DASAR TEORI

Procedure

Prosedur dapat berisi statement SQL (INSERT, DELETE, SELECT) atau operasi lain yang disimpan dalam database. Sebuah procedure dapat dipanggil menggunakan statement CALL nama_prosedur disertai parameter yang diperlukan.

Membuat procedure

Sintaks untuk membuat procedure adalah sebagai berikut :

```
CREATE PROCEDURE procedure_name ([proc_parameter[,...]])
Routine_body
```

Berikut adalah contoh pembuatan *stored procedure*:

1. Menggunakan table employee. Data yang terdapat pada table beli :

```
MariaDB [test]> SELECT * FROM employee;
```

id	first_name	last_name	start_date	end_date	salary	city	description
1	Jason	Martin	1996-07-25	2006-07-25	1234.56	Toronto	Programmer
2	Alison	Mathews	1976-03-21	1986-02-21	6661.78	Vancouver	Tester
3	James	Smith	1978-12-12	1990-03-15	6544.78	Vancouver	Tester
4	Celia	Rice	1982-04-21	1999-04-21	2344.78	Vancouver	Manager
5	Robert	Black	1984-01-15	1998-08-08	2334.78	Vancouver	Tester
6	Linda	Green	1987-07-30	1996-01-04	4322.78	New York	Tester
7	David	Larry	1990-12-31	1998-02-12	7897.78	New York	Manager
8	James	Cat	1996-09-17	2002-04-15	1232.78	Vancouver	Tester
9	Caroline	Grass	1978-04-23	2003-03-26	7897.78	New York	Tester
10	Doe	NULL	NULL	NULL	3500.00	Indonesia	NULL
11	John	Lenon	1996-07-25	2006-07-25	1000.00	Canberra	Programmer

```
11 rows in set (0.00 sec)
```

2. Membuat prosedur dengan nama `show_employee()`, dengan perintah sebagai berikut:

```
MariaDB [test]> DELIMITER &&
MariaDB [test]> CREATE PROCEDURE show_employees()
  -> BEGIN
  -> SELECT * FROM employee;
  -> END &&
Query OK, 0 rows affected (0.08 sec)
```

3. Memanggil prosedur `show_employees()` yang telah dibuat sebelumnya. Dengan perintah `CALL`, sebagai berikut:

```
MariaDB [test]> DELIMITER ;
MariaDB [test]> CALL show_employees();
```

id	first_name	last_name	start_date	end_date	salary	city	description
1	Jason	Martin	1996-07-25	2006-07-25	1234.56	Toronto	Programmer
2	Alison	Mathews	1976-03-21	1986-02-21	6661.78	Vancouver	Tester
3	James	Smith	1978-12-12	1990-03-15	6544.78	Vancouver	Tester
4	Celia	Rice	1982-04-21	1999-04-21	2344.78	Vancouver	Manager
5	Robert	Black	1984-01-15	1998-08-08	2334.78	Vancouver	Tester
6	Linda	Green	1987-07-30	1996-01-04	4322.78	New York	Tester
7	David	Larry	1990-12-31	1998-02-12	7897.78	New York	Manager
8	James	Cat	1996-09-17	2002-04-15	1232.78	Vancouver	Tester
9	Caroline	Grass	1978-04-23	2003-03-26	7897.78	New York	Tester
10	Doe	NULL	NULL	NULL	3500.00	Indonesia	NULL
11	John	Lenon	1996-07-25	2006-07-25	1000.00	Canberra	Programmer

```
11 rows in set (0.00 sec)
```

Tampilan hasil dari perintah `CALL show_employees()`, memiliki hasil yang sama dengan hasil dari perintah `SELECT * FROM employee;`

Parameter dalam Procedure

Procedure menerima parameter (`proc_parameter`) yang didefinisikan sebagai berikut :

```
Proc_parameter :
    [IN | OUT | INOUT]param_name type
```

Dalam parameter procedure, didefinisikan jenis parameter, yaitu IN, OUT, atau INOUT. Bila tidak dispesifikasika saat pembuatan, maka default jenis parameter yang dipilih adalah IN.

- Parameter IN mengirimkan sebuah nilai ke dalam procedure. Rovedure bisa saja merubah nilai parameter ini, namun parameter tersebut tidakvisibel terhadap pemanggilan procedure ketika procedure tersebut selesai.
- Parameter OUT mengirimkan nilai dari procedure ke pemanggil. Nilai inisial dari parameter ini adalah NULL dan nilainya visible terhadap pemanggil.
- Parameter INOUT diinisialisasi oleh pemanggil, dapat dimodifikasi oleh procedure, dan perubahan nilai parameter visible terhadap pemanggil ketika procedure selesai.

Parameter IN

Berikut adalah contoh penggunaan parameter IN :

```
MariaDB [test]> DELIMITER &&
MariaDB [test]> CREATE PROCEDURE getEmployeeByCity (IN cityName VARCHAR(255))
-> BEGIN
-> SELECT * FROM employee WHERE city LIKE cityName;
-> END &&
Query OK, 0 rows affected (0.07 sec)
```

Pada contoh di atas, dibuat procedure `getEmployeeByCity` dengan satu parameter masukan berjenis IN bernama `cityName`. Procedure ini digunakan untuk menampilkan data pada table `employee` dengan nama kota sesuai dengan parameter masukan.

Pemanggilan procedure ini dapat dilihat pada contoh di bawah ini. Pada contoh ini, kita memasukkan nama kota 'Vancouver' sebagai parameter masukan pada prosedur tersebut.

```
MariaDB [test]> DELIMITER ;
MariaDB [test]> CALL getEmployeeByCity("Vancouver");
```

id	first_name	last_name	start_date	end_date	salary	city	description
2	Alison	Mathews	1976-03-21	1986-02-21	6661.78	Vancouver	Tester
3	James	Smith	1978-12-12	1990-03-15	6544.78	Vancouver	Tester
4	Celia	Rice	1982-04-21	1999-04-21	2344.78	Vancouver	Manager
5	Robert	Black	1984-01-15	1998-08-08	2334.78	Vancouver	Tester
8	James	Cat	1996-09-17	2002-04-15	1232.78	Vancouver	Tester

```
5 rows in set (0.00 sec)
```

Parameter OUT

Berikut adalah contoh penggunaan parameter OUT.

```
MariaDB [test]> DELIMITER :)
MariaDB [test]> CREATE PROCEDURE getNumEmployee (OUT numEmployee INT)
-> BEGIN
-> SELECT COUNT(*) INTO numEmployee FROM employee;
-> END :)
Query OK, 0 rows affected (0.11 sec)
```

Pada contoh di atas, dibuat procedure untuk menampilkan jumlah employee dalam tabel. Hasil query tersebut disimpan dalam variabel `numEmployee` dengan statement `INTO numEmployee`. Pemanggilan employee dengan parameter OUT dilakukan dengan menggunakan variabel session yang diawali dengan karakter `@`. Pemanggilan procedure `getNumEmployee` ditunjukkan sebagai berikut:

```
MariaDB [test]> DELIMITER ;
MariaDB [test]> CALL getNumEmployee(@num);
Query OK, 1 row affected (0.04 sec)

MariaDB [test]> SELECT @num;
+-----+
| @num |
+-----+
|    11 |
+-----+
1 row in set (0.00 sec)
```

Pada hasil pemanggilan prosedur `getNumEmployee` tersebut terdapat 11 data employee.

Pada parameter dengan mode INOUT ini, kita bisa mengirimkan parameter kedalam stored procedure dan mendapatkan nilai kembalian yang baru dari stored procedure yang didefinisikan

LATIHAN

1. Buatlah procedure untuk menambahkan data pada tabel employee
2. Buatlah procedure untuk mengupdate data pada tabel employee
3. Buatlah procedure untuk menghitung jumlah karyawan yang tinggal di city Vancouver