

# PRAKTIKUM SISTEM BASIS DATA

## Modul 14 – Stored Function & Cursor

### Tujuan dan Capaian Pembelajaran Praktikum:

1. Mahasiswa dapat membuat dan menggunakan variabel pada MySQL stored procedure
2. Mahasiswa dapat menggunakan IF pada MySQL stored procedure
3. Mahasiswa dapat menggunakan CASE pada MySQL stored procedure
4. Mahasiswa dapat menggunakan LOOP pada MySQL stored procedure
5. Mahasiswa dapat membuat stored procedure yang menghasilkan lebih dari satu return value.
6. Mahasiswa dapat menggunakan cursor yang digabungkan dengan stored procedure

### Alat dan Bahan:

1. Modul Praktikum
2. Komputer
3. Software MySQL, PHPMyAdmin, DataGrip atau HeidiSQL
4. Gunakan kembali file SQL **db\_sp1.sql** yang tersedia di eClass, pastikan sudah terimpor pada database anda. Bisa melanjutkan database di pertemuan sebelumnya.
5. Viewer LCD / Moodle

### Materi dan Pembagian Waktu:

No	Materi	Waktu
1	Pretest	15 menit
2	Mempelajari teori dan mencoba guided challenges	45 menit
3	Unguided challenges	45 menit
4	Posttest	45 menit
	Total	150 menit

### Materi Praktikum

#### 1. Function pada MySQL

Seperti pada bahasa pemrograman, fungsi adalah program yang **mengembalikan nilai**. Pada MySQL juga dapat dibuat program yang dimasukkan ke dalam fungsi, mirip seperti pada stored procedure, bisa memiliki parameter input, namun di dalam fungsi harus mengembalikan nilai dalam suatu tipe data tertentu. Proses pemanggilannya dapat dilakukan di dalam sintaks statemen lain secara langsung, misalnya pada statemen SELECT.

Contoh:

```
DELIMITER $$

CREATE FUNCTION countCustCountry (countryIn VARCHAR(15)) RETURNS INT
BEGIN
    DECLARE result INT;
    SELECT COUNT(*) INTO result FROM customers WHERE country = countryIn;
    RETURN result;
END$$

DELIMITER;
```

Sedangkan pemanggilannya adalah: `SELECT countCustCountry('USA');`

Contoh lain untuk menghitung komisi 5% berdasarkan batas kredit customer:

```
DELIMITER $$
CREATE FUNCTION addCommission(credLim DECIMAL(10,2))
    RETURNS DECIMAL(10,2)
BEGIN
    DECLARE comm FLOAT DEFAULT 0.05;
    RETURN credLim * comm;
END$$
DELIMITER;
```

Berikut ini cara menampilkan nama lengkap kontak customer dengan komisi credit limitnya dengan format rupiah (asumsi \$1 = Rp 14.500,00):

```
SELECT CONCAT(contactFirstName, ' ', contactLastName) AS contact,
CONCAT('Rp ', FORMAT(addCommission(creditLimit)*14500,2, 'de_DE')) AS
commission FROM customers;
```

## 2. MySQL Cursor

Untuk bisa mengambil data hasil dari stored procedure, kita dapat menggunakan **cursor**. Cursor memungkinkan kita untuk melakukan iterasi ke dalam hasil query SELECT per baris. Sifat cursor pada MySQL adalah:

- **Read only:** tidak bisa digunakan untuk menulis / update / delete, hanya bisa baca data
- **Non scrollable:** tidak bisa mengambil data secara mundur, hanya bisa iterasi maju saja dan tidak bisa melompat ke baris tertentu.
- **Asensitive:** menunjuk ke data aktual, bukan temporary copy/insensitive.

Cursor bisa digunakan di stored procedure, stored function maupun pada trigger.

Tahapan penggunaan kursor:

1. Buat cursor:

```
DECLARE cursor_name CURSOR FOR SELECT_statement;
```

Sebuah kursor selalu berhubungan dengan pernyataan SELECT.

2. Kemudian kursor harus di open dengan statement:

```
OPEN cursor_name;
```

3. Ambil data ke kursor:

```
FETCH cursor_name INTO variables list;
```

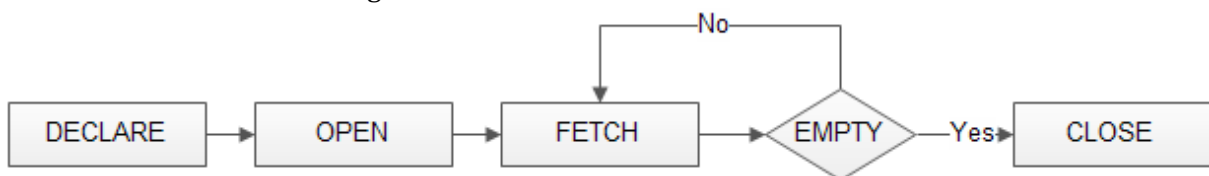
4. Tutup kursor

```
CLOSE cursor_name;
```

Ketika membuat kursor harus mendeklarasikan kondisi saat kursor tidak ketemu data sama sekali, misalnya seperti berikut:

```
DECLARE CONTINUE HANDLER FOR NOT FOUND SET finished = 1;
```

Ilustrasi kursor adalah sebagai berikut:



Contoh: Buatlah procedure berisi kursor untuk menampilkan semua email dari employees.

JAWAB :

```
CREATE PROCEDURE buildEmailList (INOUT emailList VARCHAR(4000))
BEGIN
    DECLARE vFinished INTEGER DEFAULT 0;
    DECLARE vEmail varchar(100) DEFAULT "";

    -- declare cursor for employee email
    DECLARE emailCursor CURSOR FOR
        SELECT email FROM employees;

    -- declare NOT FOUND handler
    DECLARE CONTINUE HANDLER
        FOR NOT FOUND SET vFinished = 1;

    OPEN emailCursor;
    getEmail: LOOP
        FETCH emailCursor INTO vEmail;
        IF vFinished = 1 THEN
            LEAVE getEmail;
        END IF;

        -- build email list
        SET emailList = CONCAT(vEmail,"; ",emailList);

    END LOOP getEmail;
    CLOSE emailCursor;
END;
```

Cara penggunaan :

```
SET @emailList = "";  
CALL buildEmailList(@emailList);  
SELECT @emailList;
```

## LATIHAN UNGUIDED

1. Buatlah function untuk mengambil nama lengkap pegawai/employee yang menjadi Sales Representative untuk customer tertentu. Inputnya adalah customerNumber. Outputnya nama lengkap Sales Representativenya.
2. Dengan menggunakan stored function untuk menghitung rata-rata creditLimit, tampilkan customer yang credit limitnya di atas rata-rata. Tampilan SELECT customers diurutkan berdasarkan creditLimit dari yang terbesar hingga yang terkecil di atas rata-rata hasil perhitungan function.

3. Gunakan cursor untuk menampilkan employee berdasarkan officeCode yang dimasukkan (contoh: CALL getEmployees(@empList1,1);) dengan penyajian PERSIS sebagai berikut:

officeCode	city	employees
1	San Francisco	Thompson; Jennings; Bow; Firrelli; Patterson; Murphy;
2	Boston	Patterson; Firrelli;
3	NYC	Vanauf; Tseng;
4	Paris	Gerard; Castillo; Hernandez; Bondur; Bondur;
dst..		

officeCode	city	@empList1
1	San Francisco	Thompson; Jennings; Bow; Firrelli; Patterson; Murphy;
2	Boston	Patterson; Firrelli;
3	NYC	Vanauf; Tseng;
4	Paris	Gerard; Castillo; Hernandez; Bondur; Bondur;
dst..		

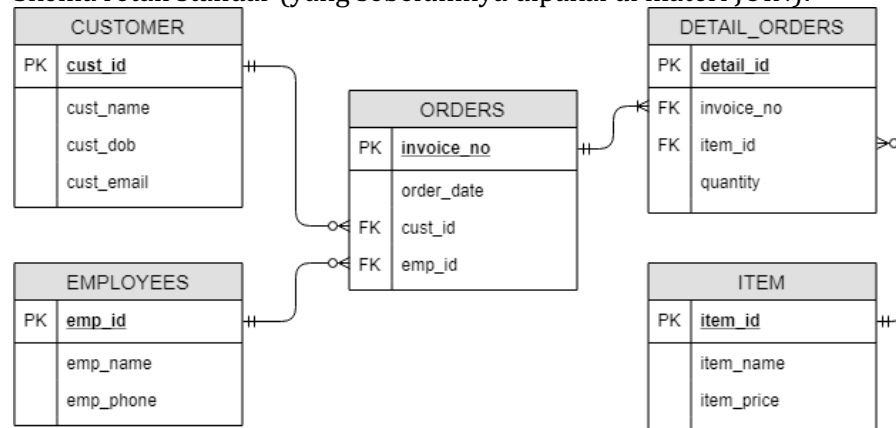
4. Dengan menggunakan cursor, buatlah tabel tampungan **tempOrderList** yang berisi barang yang dibeli sebagai berikut:

cInvoiceNo	cOrderDate	cItemName	cQuantity	cBasePrice	cSubTotal
1	2021-11-18	Apple iPhone 11	1	Rp 17.600.000,00	Rp 17.600.000,00
2	2021-11-18	Xiaomi Mi 9	2	Rp 7.500.000,00	Rp 15.000.000,00

Parameter input di prosedur **getDetailInvoice** hanyalah **invoice\_no**. Prosedur berisi SELECT untuk mengisi cursor, kemudian FETCH isi cursor untuk meng-INSERT ke tabel **tempOrderList**. Berikut DDL untuk tabel **tempOrderList**:

```
CREATE TABLE tempOrderList (  
    cInvoiceNo INT,  
    cOrderDate DATE,  
    cItemName VARCHAR(50),  
    cQuantity INT,  
    cBasePrice VARCHAR(50),  
    cSubTotal VARCHAR(50)  
)
```

Skema retail standar (yang sebelumnya dipakai di materi JOIN):



#### REFERENSI:

MySQL Stored Procedures - <https://www.sitepoint.com/stored-procedures-mysql-php/>

MySQL Tutorial : <http://www.mysqltutorial.org/mysql-stored-procedure-tutorial.aspx>