STORE PROCEDURE

Kali ini kita akan membahas tentang Stored Procedure (SP).
SP adalah:

- Kumpulan perintah SQL
- Satu kesatuan dalam sebuah object
- Ketika dieksekusi, seluruh perintah dijalankan secara prosedural (sekuensial)
- Dalam SP mengandung deklarasi variabel, logika (IF THEN atau CASE WHEN), dan juga ada LOOPING (WHILE END WHILE)
- Dalam SP tingkat lanjut, terintegrasi object lain yaitu TRIGGER dan CURSOR (pembahasan tersendiri)

DELIMITER

Salah satu hal yang terpenting dalam SP adalah memperhatikan delimiter yang digunakan. Delimiter adalah tanda/simbol bahwa sebuah perintah sudah selesai ditulis dan siap untuk dieksekusi. Dalam MySQL, delimiter default adalah simbol ;
Tapi kita bisa mengganti delimiter dengan simbol yang lain.

Saya akan membahasnya dalam bentuk contoh coding (lihat video).

Ketika saya memberikan perintah DELIMITER menjadi ##, maka DELIMITER ; sudah tidak berlaku lagi sehingga ketika saya mengetik perintah apapun dan diakhiri dengan tanda demiliter ; maka perintah tersebut tidak akan dieksekusi. Perintah hanya bisa dieksekusi jika delimiter yang digunakan adalah ##

Saya bisa mengganti delimiter dengan tanda atau simbol apapun, termasuk mengembalikan delimiter ke nilai default nya yaitu simbol ;

STRUKTUR STORED PROCEDURE

1. Karena SP adalah bagian dari sebuah database, maka SP tidak bisa dilepaskan integrasinya dari sebuah database. Maka langkah awal dari SP adalah membuat database induk terlebih dahulu.

```
drop database if exists dbSP;

create database dbSP;

use dbSP;

create table tblSP
(
nourut int primary key,
datanya varchar(30),
ukuran decimal(8,5)
);
```

2. Langkah berikut adalah setting delimiter sebelum menuliskan coding.

```
DELIMITER ##
```

```
##
DELIMITER ;
```

3. Selalu diawali dengan perintah CREATE PROCEDURE {diikuti nama procedure nya}

```
CREATE PROCEDURE spHello()
```

4. Selalu dimulai dengan perintah BEGIN dan diakhir dengan END

```
BEGIN
END;
```

5. Di antara BEGIN dan END, dapat dituliskan secara berurutan dimulai dengan DECLARE untuk deklarasi variabel yang harus diikuti oleh tipe data. (Tipe data yang digunakan adalah tipe data yang berlaku di MySQL.

```
DECLARE vHello VARCHAR(100);
```

Cara mendeklarasikan variabel:
- Menggunakan perintah SET
- Menggunakan perintah SELECT INTO

```
SET vHello = 'HELLO WORLD';
```

atau

```
SELECT 'HELLO WORLD' INTO vHello;

SELECT vHello AS HASIL;
```

6. Memanggil SP dengan menggunakan perintah CALL {nama SP nya}

```
CALL spHello()
```

Hasil lengkap bisa dilihat di video, bagaimana teknik menulis SP.

STORED PROCEDURE DENGAN PARAMETER

Kita bisa memanfaatkan parameter atau variabel pada sebuah SP. Parameter adalah nilai yang dikirim ke SP supaya dapat diproses oleh SP itu sendiri.

CONTOH 1: memanfaatkan parameter pNama untuk spHai

```
delimiter ##
create procedure spHai(pNama varchar(100))
begin
        select concat('Hai ', pNama, ' Senang berkenalan dengan Anda');
end##
delimiter ;

call spHai('Teknik Informatika');
```

CONTOH 2: memanfaatkan parameter nilai X dan Y untuk melakukan perhitungan

```
delimiter ##
create procedure spHitung(x Int, y Int)
begin
        select x*y as Hasil_Perkalian;
        call spHai('Evander'); /*memanggil sp lain*/
end##
delimiter ;

call spHitung(5, 9);
```

STORED PROCEDURE DENGAN FUNGSI LOGIKA

Kita dapat menggunakan fungsi logika di dalam SP. Terdapat 2 macam fungsi logika yaitu IF THEN (untuk 2 kondisi saja) dan CASE WHEN (beberapa kondisi).

Perhatikan 2 contoh berikut bagaimana menuliskan struktur fungsi logika.

CONTOH 1: menggunakan fungsi logika IF THEN untuk 2 kondisi yaitu membedakan operasi perkalian atau pembagian dari variabel X dan variabel Y

```
delimiter ##
create procedure spHitung2(cek varchar(4), x Int, y Int)
begin
        IF cek='KALI' THEN /*selalu cek 2 kondisi saja*/
                select x*y as HASIL_KALI;
        ELSE
                select x/y as HASIL_BAGI;
        END IF;
end##
delimiter ;

call spHitung2('KALI', 6, 2);
call spHitung2('BAGI', 6, 2);
```

CONTOH 2: menggunakan fungsi logika CASE WHEN untuk 4 kondisi dari parameter X dan Y

```
delimiter ##
create procedure spHitung3(cek varchar(10), x int, y int)
begin
        case
                when cek = 'TAMBAH' then
                        select x+y as HASIL_JUMLAH;
                when cek = 'KURANG' then
                        select x-y as HASIL_KURANG;
                when cek = 'KALI' then
                        select x*y as HASIL_KALI;
                when cek = 'BAGI' then
                        select x/y as HASIL_BAGI;
        end case;
end##
delimiter ;

call spHitung3('TAMBAH', 10, 6);
call spHitung3('KURANG', 10, 6);
call spHitung3('KALI', 10, 6);
call spHitung3('BAGI', 10, 6);
```

PERULANGAN (LOOPING) DALAM STORED PROCEDURE

Ada 3 macam perulangan yang bisa kita gunakan dalam SP, yaitu:

1. REPEAT . . . . UNTIL . . . . END REPEAT
2. LOOP . . . . END LOOP (bisa disertai dengan LEAVE untuk keluar dari loop)
3. WHILE . . . . DO . . . . END WHILE

Ketiga nya dapat kita gunakan untuk proses perulangan dan hasilnya sama, hanya berbeda cara penulisan struktur logika nya.

CONTOH 1: menggunakan perintah WHILE untuk mengisi 1000 data di tblSP dengan nilai Random

```
delimiter ##
create procedure spIsiData(jum INT)
begin
        declare i int default 1;
        while i<jum+1 do
                insert into tblSP values
                (i, concat('data ke ', i), rand()*1000);
                set i:=i+1;
        end while;
end##
delimiter ;

call spIsiData(1000);
```

Kita dapat menggunakan perintah LOOP dengan hasil yang sama dengan perintah di atas.

```
delimiter ##
create procedure spIsiData(jum INT)
begin
      declare i int default 1;
      isidata: LOOP
        IF i>jum THEN
            LEAVE isidata;
        END IF;

        insert into tblSP values
            (i, concat('data ke ', i), rand()*1000);
            set i:=i+1;
      end loop isidata;
end##
delimiter ;

call spIsiData(1000);
```

Atau kita juga bisa menggunakan perintah REPEAT dengan hasil yang juga sama.

```
delimiter ##
create procedure spIsiData(jum INT)
begin
     declare i int default 1;
     REPEAT
        insert into tblSP values
          (i, concat('data ke ', i), rand()*1000);
          set i:=i+1;

        UNTIL i=jum+1
     END REPEAT;
end##
delimiter ;

call spIsiData(1000);
```