PRAKTEK DBMS LANJUT
MINGGU 5-STORED PROCEDURE (LANJUTAN)

Gunakan database **db_sp_noabsen_nama** yang anda buat dipertemuan minggu ke-4, praktekkan materi di bawah ini.

Buat table mahasiswa dan isi nilainya

```
CREATE TABLE mahasiswa
(
    nim INT(10),
    nama VARCHAR(100),
    alamat VARCHAR(100),
    PRIMARY KEY(nim)
);

INSERT INTO mahasiswa
VALUES
(21400200,"faqih","bandung"),
(21400201,"ina","jakarta"),
(21400202,"anto","semarang"),
(21400203,"dani","padang"),
(21400204,"jaka","bandung"),
(21400205,"nara","bandung"),
(21400206,"senta","semarang");
```

```
> SELECT * FROM mahasiswa;
+----------+-------+----------+
| nim      | nama  | alamat   |
+----------+-------+----------+
| 21400200 | faqih | bandung  |
| 21400201 | ina   | jakarta  |
| 21400202 | anto  | semarang |
| 21400203 | dani  | padang   |
| 21400204 | jaka  | bandung  |
| 21400205 | nara  | bandung  |
| 21400206 | senta | semarang |
+----------+-------+----------+
7 rows in set (0.00 sec)
```

Membuat Stored Procedure **selectMahasiswa()** untuk mendapatkan seluruh NIM dan NAMA mahasiswa

```
DELIMITER $$

CREATE PROCEDURE selectMahasiswa()
BEGIN
    SELECT nim, nama FROM mahasiswa;
END$$

DELIMITER ;
```

Untuk memanggil Stored Procedure **selectMahasiswa()**

```
CALL selectMahasiswa()
```

Hasilnya adalah

```
> CALL selectMahasiswa();
+----------+-------+
| nim      | nama  |
+----------+-------+
| 21400200 | faqih |
| 21400201 | ina   |
| 21400202 | anto  |
| 21400203 | dani  |
| 21400204 | jaka  |
| 21400205 | nara  |
| 21400206 | senta |
+----------+-------+
7 rows in set (0.00 sec)
```

Jadi, setiap ingin menampilkan NIM dan NAMA mahasiswa kita tidak perlu membuat kode SQL seperti biasanya. Cukup simpan kode SQL di Stored Procedure dan bisa kita panggil dan gunakan berulang-ulang

Stored Procedure dengan Parameter

Kita juga memasukkan parameter di Stored Procedure agar menjadi lebih dinamis
Contoh, kita ingin membuat kode SQL untuk mencari data mahasiswa berdasarkan alamat.

```
DELIMITER $$

CREATE PROCEDURE alamatMahasiswa
(
    alamatMhs VARCHAR(100)
)
BEGIN
    SELECT *
    FROM mahasiswa
    WHERE alamat = alamatMhs;

END$$

DELIMITER ;
```

Di dalam nama Stored Procedure terdapat parameter **alamatMhs** dengan tipe data varchar. Saat masuk ke kode SQL yaitu mencari mahasiswa dengan alamat yang sama dengan parameter yang diinputkan.

Cara memanggilnya adalah

```
CALL alamatMahasiswa("bandung")
```

Hasilnya

```
> CALL alamatMahasiswa("bandung");
+----------+-------+---------+
| nim      | nama  | alamat  |
+----------+-------+---------+
| 21400200 | faqih | bandung |
| 21400204 | jaka  | bandung |
| 21400205 | nara  | bandung |
+----------+-------+---------+
3 rows in set (0.00 sec)
```

Dengan Stored Procedure alamatMahasiswa() kita mencari data mahasiswa yang berasal dari “bandung”.

DML dengan Stored Procedure

Stored Procedure tidak hanya bisa diterapkan di Data Query Language (DQL) tetapi juga Data Manipulation Language (DML)

Misal kita ingin memasukkan data mahasiswa dengan Stored Procedure

```
DELIMITER $$

CREATE PROCEDURE insertMahasiswa
(
    nimMhs INT(10),
    namaMhs VARCHAR(100),
    alamatMhs VARCHAR(100)
)
BEGIN
    INSERT INTO mahasiswa
    VALUES (nimMhs, namaMhs, alamatMhs);

END$$

DELIMITER ;
```

Jika kita ingin memasukkan record baru di table mahasiswa maka akan ada 3 parameter saat memanggil Stored Procedure insertMahasiswa()

```
> CALL insertMahasiswa(21400207,"joni","jakarta");
Query OK, 1 row affected (0.05 sec)

> SELECT * FROM mahasiswa;
+----------+-------+----------+
| nim      | nama  | alamat   |
+----------+-------+----------+
| 21400200 | faqih | bandung  |
| 21400201 | ina   | jakarta  |
| 21400202 | anto  | semarang |
| 21400203 | dani  | padang   |
| 21400204 | jaka  | bandung  |
| 21400205 | nara  | bandung  |
| 21400206 | senta | semarang |
| 21400207 | joni  | jakarta  |
+----------+-------+----------+
8 rows in set (0.00 sec)
```

Selanjutnya jika ingin memasukkan data mahasiswa ke table mahasiswa tidak perlu membuat kode INSERT berkali-kali. Kita bisa gunakan Stored Procedure insertMahasiswa() untuk menggantikan proses INSERT yang biasanya kita gunakan

Jadi Stored Procedure sangat penting dan akan memudahkan dalam menggunakan kode yang ingin dieksekusi secara berulang – ulang.

TUGAS MINGGU 5:

Screenshot setiap tahapan dan hasilnya, upload file dokumentasi ke elearning, nama file: **DBMSL-20211-REG-KELAS-M5-NOABSEN-NAMA.docx**