

MODUL X STORED PROCEDURE (2)

A. Kompetensi Dasar:

1. Mahasiswa mampu memahami implementasi stored procedure di dalam basis data
2. Mahasiswa mampu menyelesaikan operasi - operasi data spesifik dengan memanfaatkan stored procedure

B. Teori Singkat :

Stored Procedure adalah sebuah prosedur layaknya subprogram (subrutin) di dalam bahasa pemrograman reguler yang tersimpan di dalam katalog basis data.

Pada sebuah stored procedure dapat diberikan parameter sesuai dengan kebutuhan, sehingga dengan menerapkan stored procedure yang disimpan di dalam suatu database akan dapat digunakan lebih dinamis berdasarkan parameter yang diberikan, juga stored procedure dapat dikombinasikan dengan fungsi-fungsi pernyataan bersyarat dan berkondisi seperti fungsi IF, While.

Stored procedure dibuat dan disimpan hanya satu kali pada sebuah database, selanjutnya dapat dipanggil kembali, sehingga prosedur dengan nama yang sama dalam program dapat lebih cepat dieksekusi. Jenis perintah-perintah dalam stored procedure meliputi menambah data(insert), mengubah data (update), menghapus data(delete), dan memilih/menyeleksi data (select).

Beberapa kelebihan yang ditawarkan stored procedure antara lain : meningkatkan performa, mereduksi trafik jaringan, reusable, dan meningkatkan kontrol sekuriti.

Sintaks Stored Procedure

```
DELIMITER // CREATE PROCEDURE nama_sp() BEGIN perintah_sql; END // DELIMITER;
```

Untuk memanggil stored procedure, digunakan perintah CALL

```
CALL nama_sp();
```

Dalam Implementasinya, penggunaan stored procedure sering melibatkan parameter.

Di MySQL, parameter stored procedure dibedakan menjadi tiga mode : IN, OUT, dan INOUT.

IN

Parameter yang merupakan mode default ini mengindikasikan bahwa sebuah parameter dapat di-pass ke dalam stored procedure tetapi nilainya tidak dapat diubah dari dalam stored procedure

OUT

Mode ini mengindikasikan bahwa stored procedure dapat mengubah parameter dan mengirimkan kembali ke program pemanggil

INOUT

Mode ini pada dasarnya merupakan kombinasi dari mode IN dan OUT.

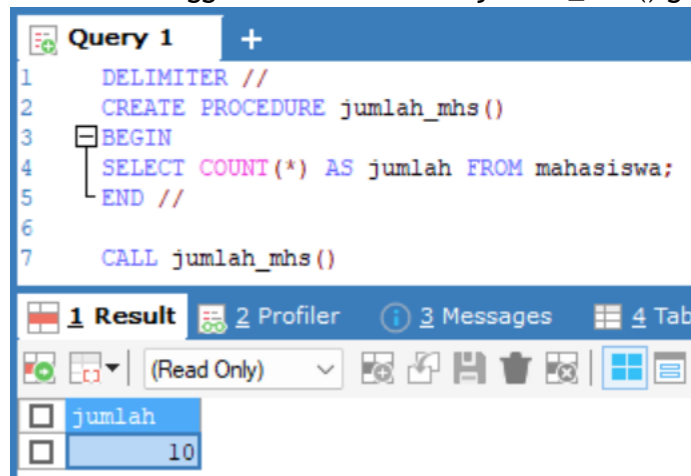
C. Langkah Praktikum:

1. Membuat Stored Procedure jumlah_mhs untuk mendapatkan jumlah mahasiswa yang ada

Sintaks:

```
DELIMITER //  
CREATE PROCEDURE jumlah_mhs()  
BEGIN  
    SELECT COUNT(*) AS jumlah FROM mahasiswa;  
END //  
DELIMITER ;
```

Untuk memanggil Stored Procedure jumlah_mhs() gunakan perintah CALL



2. JOIN dengan Stored Procedure

Misal kita ingin menggabungkan data mahasiswa dan prodi dengan Stored Procedure

Sintaks:

```
DELIMITER //  
CREATE PROCEDURE  
    join_mahasiswa_prodi()  
BEGIN  
    SELECT    mahasiswa.nim,    mahasiswa>Nama,    mahasiswa.kode_prodi,  
            prodi>Nama_prodi  
    FROM mahasiswa
```

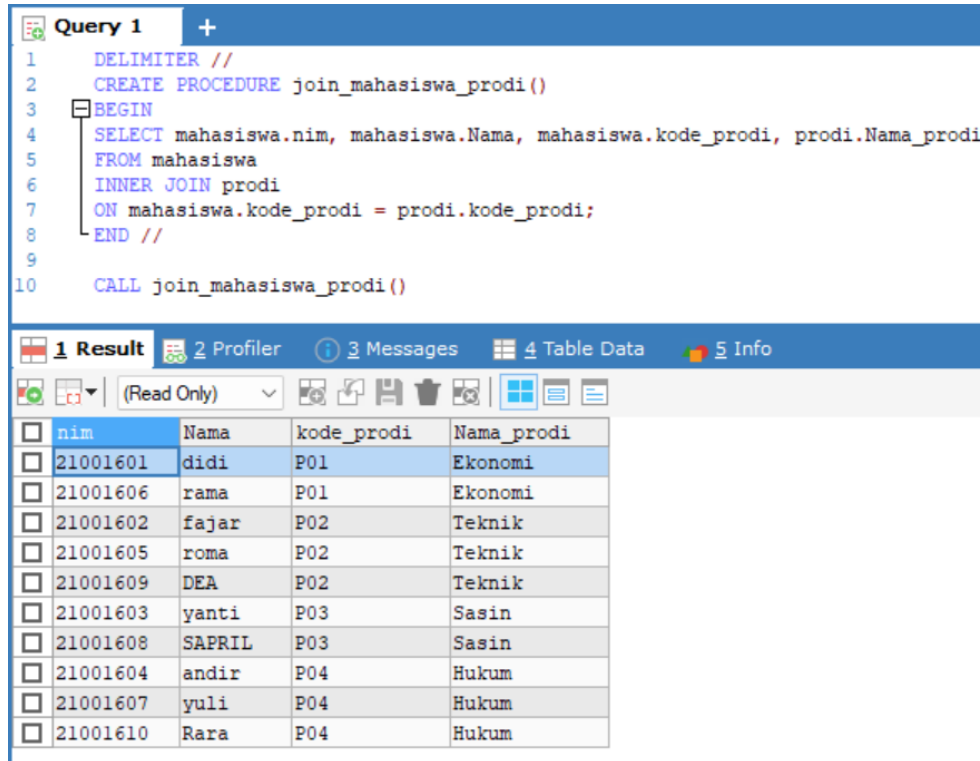
INNER JOIN prodi

ON mahasiswa.kode_prodi = prodi.kode_prodi;

END //

DELIMITER ;

Penggunaan



The screenshot shows a SQL IDE interface. The top pane, titled 'Query 1', contains the following SQL code:

```
1 DELIMITER //  
2 CREATE PROCEDURE join_mahasiswa_prodi()  
3 BEGIN  
4 SELECT mahasiswa.nim, mahasiswa>Nama, mahasiswa.kode_prodi, prodi>Nama_prodi  
5 FROM mahasiswa  
6 INNER JOIN prodi  
7 ON mahasiswa.kode_prodi = prodi.kode_prodi;  
8 END //  
9  
10 CALL join_mahasiswa_prodi();
```

The bottom pane, titled '1 Result', shows the results of the query in a table format. The table has four columns: 'nim', 'Nama', 'kode_prodi', and 'Nama_prodi'. There are 10 rows of data.

nim	Nama	kode_prodi	Nama_prodi
21001601	didi	P01	Ekonomi
21001606	rama	P01	Ekonomi
21001602	fajar	P02	Teknik
21001605	roma	P02	Teknik
21001609	DEA	P02	Teknik
21001603	yanti	P03	Sasin
21001608	SAPRIL	P03	Sasin
21001604	andir	P04	Hukum
21001607	yuli	P04	Hukum
21001610	Rara	P04	Hukum

3. Stored Procedure dengan Parameter

Kita juga memasukkan parameter di Stored Procedure agar menjadi lebih dinamis

Contoh, kita ingin membuat kode SQL untuk mencari data buku berdasarkan kata kunci yang kita inginkan. **sintaks:**

DELIMITER //

CREATE PROCEDURE cari_buku(IN kata_kunci VARCHAR(255))

BEGIN

SELECT * FROM buku

WHERE Judul LIKE CONCAT('%', kata_kunci, '%');

END //

DELIMITER ;

Query 1 x buku +

```
1 DELIMITER //
2 CREATE PROCEDURE cari_buku(
3     IN kata_kunci VARCHAR(255)
4 )
5 BEGIN
6     SELECT * FROM buku
7     WHERE Judul LIKE CONCAT('%', kata_kunci, '%');
8 END //
9
10 CALL cari_buku('organisasi')
```

1 Result

2 Profiler

3 Messages

4 Table Data

5 Info

(Read Only)

<input type="checkbox"/>	kode_buku	Judul	penulis	penerbit	jumlah
<input type="checkbox"/>	B02	Organisasi Komputer	Imam Riadi, SPd, M.Kom	andi	240

D. Tugas

1. Buat stored procedure dimana kita bisa mendapatkan data buku yang belum dipinjam.
2. Buat stored procedure untuk menghapus data di tabel mahasiswa.
3. Buat stored procedure yang bisa mendapatkan semua data prodi, berisi nama fakultas dan lokasi