

LAPORAN RESMI
MODUL III
(STORED PROCEDURE)
SISTEM MANAJEMEN BASIS DATA



NAMA	: IMAM ARISHANDI IRFANTO
N.R.P	: 220441100034
DOSEN	: FITRI DAMAYANTI, S.Kom,M.Kom
ASISTEN	: M. IQBAL FIRMANSYAH
TGL PRAKTIKUM	: 26 APRIL 2024

Disetujui, Mei 2024

Asisten

M. IOBAL FIRMANSYAH

21.04.411.00084

LABORATORIUM BISNIS INTELIJEN SISTEM



PRODI SISTEM INFORMASI

JURUSAN TEKNIK INFORMATIKA

FAKULTAS TEKNIK

UNIVERSITAS TRUNOJOYO MADURA

BAB I

PENDAHULUAN

1.1 Latar Belakang

Stored procedure merupakan salah satu fitur penting dalam sistem manajemen basis data (DBMS) yang memungkinkan pengguna untuk menyimpan kumpulan instruksi SQL yang telah didefinisikan di dalam database. Latar belakang penggunaan stored procedure sangatlah beragam dan berguna dalam konteks pengembangan aplikasi dan manajemen data.

Selain efisiensi, stored procedure juga menyediakan tingkat abstraksi yang lebih tinggi untuk pengguna dalam mengakses dan memanipulasi data. Hal ini memungkinkan pengembang untuk mengimplementasikan logika bisnis kompleks di dalam database, mengurangi kompleksitas kode aplikasi, dan mempercepat proses pengembangan.

Keuntungan lain dari penggunaan stored procedure adalah kemampuannya untuk mendukung transaksi kompleks. Dalam lingkungan transaksional, stored procedure dapat digunakan untuk mengelola proses transaksi yang melibatkan beberapa operasi SQL.

1.2 Tujuan

1. Mampu memahami dan membuat procedure dalam basis data dan mampu menggunakan perintah-perintah dalam stored procedure serta menyelesaikan operasi-operasi data spesifik dengan memanfaatkan stored procedure.

BAB II

DASAR TEORI

Stored Procedure adalah sebuah prosedur layaknya subprogram (subrutin) di dalam bahasa pemrograman reguler yang tersimpan di dalam katalog basis data. Beberapa kelebihan yang ditawarkan stored procedure antara lain : meningkatkan performa, mereduksi trafik jaringan, reusable, dan meningkatkan kontrol sekuriti. Di balik kelebihan tersebut, stored procedure juga memiliki kekurangan.

2.1 Sintaks stored procedure:

```
<create procedure statement> ::=
CREATE PROCEDURE <procedure name> ( [ <parameter list>
] )
<routine body>
<parameter list> ::=
<parameter specification> [ , <parameter
specification> ]...
<parameter specification> ::=
[ IN | OUT | INOUT ] <parameter> <data type>
<routine body> ::= <begin-end block>
<begin-end block> ::=
```

Pernyataan pembuatan stored procedure berikut :

```
DELIMITER //
CREATE PROCEDURE getMhs()
BEGIN
    SELECT * FROM mahasiswa;
END //
DELIMITER;
```

Perintah DELIMITER digunakan untuk mengubah delimiter standar, misalnya di sini dari titik koma (;) menjadi slash ganda (//). Langkah ini umumnya dilakukan ketika isi stored procedure mengandung titik koma – yang merupakan delimiter standar di SQL. Pernyataan di antara BEGIN dan END merupakan badan (body) stored procedure. Perintah DELIMITER di akhir baris digunakan untuk mengembalikan delimiter ke karakter semula.

2.2 Aktivasi/pemanggilan Stored Procedure

```
<call statement> ::=  
CALL [ <database name> . ] <stored procedure name>  
( [ <scalar expression> [ , <scalar expression> ]... ] )
```

Eksekusi Query tersebut dengan memanggil procedure getMahasiswa().

```
CALL getMhs ();
```

idMhs	namaMhs	almtMhs	jKlmn
1011	Ari	Surabaya	L
1012	Bagus	Malang	L
1013	Candra	Sampang	L
1014	Dhani	Surabaya	P
1018	Frinza	Kediri	P

2.3 Menghapus Stored Procedure

```
<drop procedure statement> ::=  
DROP PROCEDURE [ IF EXISTS ]  
  
[ <database name> . ] <procedure name>
```

Dalam implementasinya, penggunaan stored procedure sering melibatkan parameter. Di MySQL, parameter stored procedure dibedakan menjadi tiga mode: IN, OUT, dan INOUT.

1. 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)

Sebagai contoh, kita bisa mendapatkan semua data matakuliah di semester tertentu.

```
DELIMITER //  
CREATE PROCEDURE getNamaByJenisKlamin(IN jeniskelamin varchar(1))  
BEGIN  
    select * from mhs  
    where jKlmn = jeniskelamin;  
END //  
DELIMITER ;
```

Untuk memanggil stored procedure yang memiliki parameter, maka kita harus menspesifikasikan argumennya. Misalkan kita ingin mendapatkan data mahasiswa dengan jenis kelamin laki-laki.

```
CALL getNamaByJenisKlamin(3);
```

idMhs	namaMhs	almtMhs	jKlmn
1011	Ari	Surabaya	L
1012	Bagus	Malang	L
1013	Candra	Sampang	L

Apabila pemanggilan stored procedure di atas mengabaikan argumen, DBMS akan merespon dengan pesan kesalahan. Bergantung kebutuhan, pendefinisian parameter pada stored procedure juga bisa lebih dari satu. Sebagai contoh, buat stored procedure dengan dua buah parameter seperti berikut :

```
DELIMITER //
CREATE PROCEDURE getNamaByAlmtJKelamin(
    IN Alamat VARCHAR(20),
    IN jeniskelamin VARCHAR(2))
BEGIN
    SELECT * FROM mhs
    WHERE jklmn = jeniskelamin
    AND almtMhs = Alamat ;
END //
DELIMITER ;
```

Pemanggilan stored procedure di atas tentunya akan memerlukan dua buah argumen.

```
CALL getNamaByAlmtJKelamin ('surabaya', 'L');
```

idMhs	namaMhs	almtMhs	jKlmn
1011	Ari	Surabaya	L

Penambahan Data

Pada operasi penambahan, data – data terkait diisikan melauai argumen. Selanjutnya, isi stored procedure akan memasukkan data ke dalam tabel. Berikut adalah contoh stored procedure untuk menambahkan data pada tabel mahasiswa.

```
DELIMITER //
CREATE PROCEDURE AddMahasiswa(
    IN idMhs INT(10),
    IN namaMhs VARCHAR(20),
    IN almtMhs VARCHAR(30),
    IN jklmn VARCHAR(2))
BEGIN
    INSERT INTO mhs
    VALUES (idMhs,namaMhs,almtMhs,jklmn);
END //
DELIMITER ;
```

Lakukan eksekusi terhadap procedure tersebut

```
call AddMahasiswa('1019', 'Gunawan', 'surabaya', 'L');
```

Selanjutnya lakukan pengecekan data pada tabel mahasiswa.

```
select * from mhs; atau
select * getMhs();
```

idMhs	namaMhs	almtMhs	jKlmn
1011	Ari	Surabaya	L
1012	Bagus	Malang	L
1013	Candra	Sampang	L
1014	Dhani	Surabaya	P
1018	Frinza	Kediri	P
1019	Gunawan	surabaya	L

2. OUT

Mode ini mengindikasikan bahwa stored procedure dapat mengubah parameter dan mengirimkan kembali ke program pemanggil. Dalam konteks bahasa pemrograman, parameter OUT analog dengan passing-byreference. Dengan demikian, parameter ini nilainya bisa diubah oleh stored procedure.

```
DELIMITER //
CREATE PROCEDURE JumlahMahasiswa(OUT jumlah_mhs INT(3))
BEGIN
    SELECT COUNT(idMhs)
    INTO jumlah_mhs FROM mhs;
END //
DELIMITER ;
```

Untuk mengeksekusi stored procedure dengan parameter OUT, dibutuhkan argumen yang spesifik.

```
call JumlahMahasiswa(@jumlah_mhs);
```

Perhatikan, argumen harus menggunakan notasi @, yang mengindikasikan sebagai suatu parameter OUT.

Langkah selanjutnya, untuk mendapatkan nilai variabel, gunakan pernyataan SELECT

```
select @jumlah_mhs;
```

```
@jumlah_mhs
6
```

Parameter mode OUT juga bisa dikombinasikan dengan mode IN.

3. OUT

Mode ini pada dasarnya merupakan kombinasi dari mode IN dan OUT. kita bisa mengirimkan parameter kedalam stored procedure dan mendapatkan nilai kembalian yang baru dari stored procedure yang didefinisikan.

Sebagai contoh, definisikan stored procedure seperti berikut :

```

DELIMITER //
CREATE PROCEDURE CountByGender(
    IN gender VARCHAR(2),
    OUT total INT(3))
BEGIN
    SELECT COUNT(idMhs)
    INTO total
    FROM mhs
    WHERE jklmn = gender;
END //
DELIMITER ;

```

Lakukan eksekusi pada procedure tersebut untuk mencari jumlah mahasiswa yang berjenis kelamin perempuan .

```

call CountByGender('P',@total);
select @total;

```

```

@total
2

```

Stored procedure dapat mencerminkan beragam operasi data, misalnya seleksi, penambahan, pengubahan, penghapusan, dan juga operasi – oprasi DDL. Seperti halnya procedure di dalam bahasa pemrograman, stored procedure juga dapat melibatkan variabel, pernyataan kondisional, dan pengulangan.

Jury 02/05/2020

BAB III

TUGAS PENDAHULUAN

3.1 Soal

1. Apa itu stored procedure dalam konteks sistem manajemen basis data?
2. Apa Perbedaan antara stored Procedure dan fungsi dalam database?
3. Bagaimana query untuk membuat stored procedure? jelaskan!
4. Apa yang dimaksud dengan Parameter IN, OUT, INOUT dalam konteks stored procedure?
5. Kapan kita menggunakan parameter INOUT daripada IN atau OUT dalam stored procedure?

3.2 jawaban

1. Stored procedure adalah kumpulan atau kumpulan pernyataan SQL yang telah disimpan dalam basis data dan dapat dipanggil oleh aplikasi basis data untuk menjalankan tugas-tugas tertentu.
2. - Stored procedure dapat mengembalikan nol atau banyak nilai melalui parameter OUT, sedangkan fungsi hanya mengembalikan satu.
- Stored Procedure dapat melakukan perubahan pada data
Contoh: INSERT, UPDATE, DELETE. Sedangkan fungsi tidak dapat melakukan Perubahan data tersebut.
- Stored procedure biasanya digunakan untuk tugas-tugas yang kompleks atau untuk mempercepat kinerja, sedangkan fungsi sering digunakan untuk perhitungan atau operasi sederhana.

3. CREATE PROCEDURE Nama-Prosedur

AS

BEGIN

SELECT * FROM Nama-tabel;

UPDATE Nama-tabel SET kolom : nilai WHERE kondisi;

INSERT INTO Nama-tabel (kolom) VALUES (nilai);

END

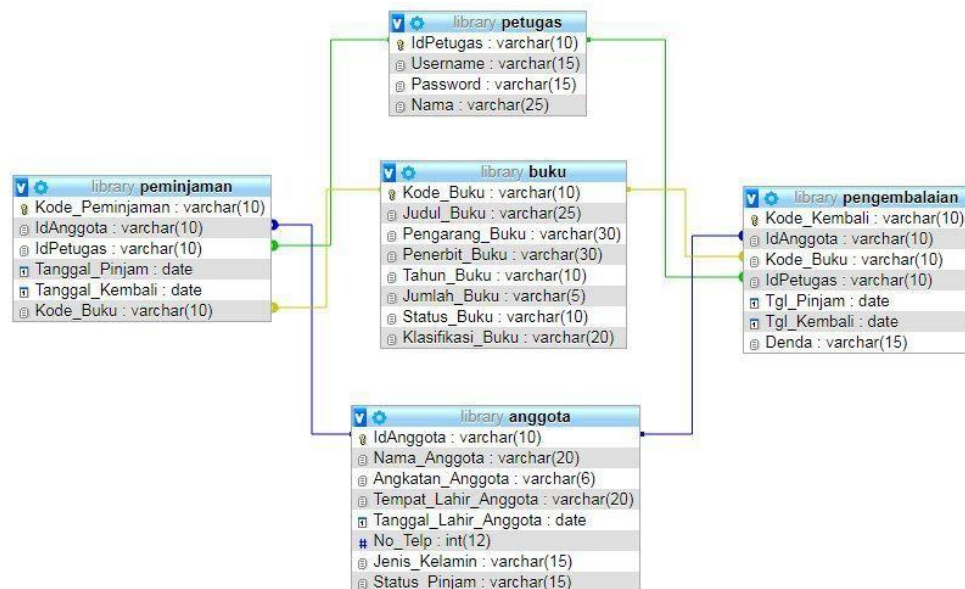
buat create lalu select sesuai dengan tabel yang ingin digunakan, lalu perintah update nilai ke tabel yang sudah dipilih, lalu tutup dengan END

Handwritten signature and date: 02/05/2021

4. • IN = digunakan untuk memberikan nilai input kedalam stored Procedure
 - OUT = digunakan untuk mengembalikan nilai dari stored Procedure
 - INOUT = digunakan untuk memberikan dan mengembalikan nilai dari stored Procedure.
5. Kita menggunakan INOUT dari pada IN atau OUT dalam stored Procedure ketika perlu memberikan nilai awal kedalam parameter dan perlu mendapatkan nilai yang dihasilkan oleh prosedur.

BAB IV IMPLEMENTASI

4.1 Soal



1. Buatlah sebuah prosedur dengan menggunakan parameter IN default
2. Buatlah stored procedure untuk mengetahui data pada table buku berdasarkan pada salah satu field yaitu penerbit buku
3. Butalah stored procedure untuk mengetahui data pada table peminjaman berdasarkan dua field yaitu tanggal peminjaman dan kode buku
4. Buatlah stored procedure untuk mengetahui data pada table anggota berdasarkan pada 3 field yaitu angkatan, tempat lahir dan jenis kelamin
5. Definisikan stored procedure untuk memasukkan data pada table anggota
6. Definisikan stored procedure untuk mengetahui data jumlah anggota menggunakan parameter OUT
7. Definisikan stored procedure untuk mengetahui data jumlah buku berdasarkan pengarang dan tahun terbitnya menggunakan parameter INOUT

4.2 Source code

-1-

```
DELIMITER //
CREATE PROCEDURE search_anggota()
    BEGIN
        SELECT * FROM tb_anggota;
    END //
DELIMITER ;
CALL search_anggota();
```

-2-

```
DELIMITER //
CREATE PROCEDURE search_penerbit_buku(IN PenerbitBuku
VARCHAR(30))
    BEGIN
        SELECT * FROM buku
        WHERE Penerbit_Buku = PenerbitBuku;
    END //
DELIMITER ;
CALL search_penerbit_buku('imam');
```

-3-

```
DELIMITER //
CREATE PROCEDURE search_tgl_pinjam(
    IN TanggalPinjam DATE,
    IN KodeBuku VARCHAR (10))
    BEGIN
        SELECT * FROM peminjaman
        WHERE Tanggal_Pinjam = TanggalPinjam
        AND Kode_Buku = KodeBuku;
    END //
DELIMITER ;
CALL search_tgl_pinjam('2024-04-12','912');
```

-4-

DELIMITER //

```
CREATE PROCEDURE search_angkatan_anggota(  
    IN Angkatan VARCHAR (6),  
    IN TempatLahir VARCHAR (20),  
    IN JenisKelamin VARCHAR (15))  
BEGIN  
    SELECT * FROM tb_anggota  
    WHERE Angkatan_Anggota = Angkatan  
    AND Tempat_Lahir_Anggota = TempatLahir  
    AND Jenis_Kelamin = JenisKelamin;  
END //
```

DELIMITER ;

CALL search_angkatan_anggota('22','Sumenep','Laki-laki');

-5-

DELIMITER //

```
CREATE PROCEDURE input_anggota(  
    idAnggota VARCHAR (10),  
    Nama_Anggota VARCHAR (20),  
    Angkatan_Anggota VARCHAR (6),  
    Tempat_Lahir_Anggota VARCHAR (20),  
    Tanggal_Lahir_Anggota DATE,  
    No_Telp INT (12),  
    Jenis_Kelamin VARCHAR (15),  
    Status_Pinjam VARCHAR (15))  
BEGIN  
    INSERT INTO tb_anggota VALUES  
(idAnggota,Nama_Anggota,Angkatan_Anggota,Tempat_Lahir_Anggota,T  
anggal_Lahir_Anggota,No_Telp,Jenis_Kelamin,Status_Pinjam);  
END //
```

DELIMITER ;

```
CALL input_anggota('14', 'Akhmad Sulthon Rabbani', '22', 'Sumenep',  
'2003-08-04', 088291847647, 'Laki-Laki', 'Sudah');
```

```
SELECT * FROM tb_anggota;
```

-6-

```
DELIMITER //
```

```
CREATE PROCEDURE get_jumlah_anggota(  
    OUT jumlah_anggota INT)
```

```
BEGIN
```

```
    SELECT COUNT(*) INTO jumlah_anggota FROM tb_anggota;
```

```
END //
```

```
DELIMITER ;
```

```
CALL get_jumlah_anggota(@total_anggota);
```

```
SELECT @total_anggota AS TotalAnggota;
```

-7-

```
DELIMITER //
```

```
CREATE PROCEDURE jumlah_buku(  
    IN NamaPengarang VARCHAR (35),
```

```
    INOUT tahunTerbit VARCHAR (10),
```

```
    OUT totalBuku INT)
```

```
BEGIN
```

```
    SELECT SUM(Jumlah_Buku) INTO totalBuku
```

```
    FROM buku
```

```
    WHERE Pengarang_Buku = NamaPengarang
```

```
    AND Tahun_Buku = tahunTerbit;
```

```
    SET tahunTerbit = tahunTerbit + 1;
```

```
END //
```

```
DELIMITER ;
```

```
SET @tahunTerbit = '2019';
```

```
SET @nama = 'imam' ;
```

```
CALL jumlah_buku(@nama, @tahunTerbit, @totalBuku);
```

```
SELECT @nama, @tahunTerbit, @totalBuku AS TotalBuku;
```

4.3 Hasil

1. Soal 1

	idAnggota	Nama_Anggota	Angkatan_Anggota	Tempat_Lahir_Anggota	Tanggal_Lahir_Anggota	No_Telp	Jenis_Kelamin	Status_Pinjam
▶	1	Angga Jelasena Saput	22	Pamekasan	2003-06-04	2147483647	Laki-Laki	Belum
	10	Romauli Napitupulu	22	Medan	2024-03-24	2147483647	Perempuan	Sudah
	11	M. Irfan Faruq Zain	22	Surabaya	2024-03-27	2147483647	Laki-Laki	Sudah
	12	Naela Nahdiah	22	Sidoarjo	2024-03-16	2147483647	Perempuan	Sudah
	13	Oxa Indi Vixzio	22	Sidoarjo	2024-03-19	2147483647	Laki-Laki	Belum
	14	Akhmad Sulthon Rabba	22	Sumenep	2003-08-04	2147483647	Laki-Laki	Sudah
	2	Della Ayu Vernandah	22	Sidoarjo	2024-03-25	2147483647	Perempuan	Sudah
	3	Amelia Zalfa Sahira	22	Sampang	2024-03-17	2147483647	Perempuan	Sudah
	4	Imam Arishandi Irfan	22	Sumenep	2024-03-09	2147483647	Laki-Laki	Belum
	5	Dwi Nurjannah	22	Lumajang	2024-03-02	2147483647	Perempuan	Belum
	6	Viko Alfian P	22	Gresik	2024-03-14	2147483647	Laki-Laki	Belum
	7	Tesa Putri Hutagaol	22	Medan	2024-03-29	2147483647	Perempuan	Sudah
	8	Bintang Wahyu Ariyon	22	Sidoarjo	2024-03-22	2147483647	Laki-Laki	Belum
	9	Khoirotun Nisak	22	Sampang	2024-03-15	2147483647	Perempuan	Belum

2. Soal 2

	Kode_Buku	Judul_Buku	Pengarang_Buku	Penerbit_Buku	Tahun_Buku	Jumlah_Buku	Status_Buku	Klasifikasi_Buku
▶	234	Pejuang Sejati	anqqa	imam	2017	17	Habis	Cinta

3. Soal 3

	Kode_Peminjaman	idAnggota	idPetugas	Tanggal_Pinjam	Tanggal_Kembali	Kode_Buku
▶	1	1	13	2024-04-12	2024-05-22	912

4. Soal 4

	idAnggota	Nama_Anggota	Angkatan_Anggota	Tempat_Lahir_Anggota	Tanggal_Lahir_Anggota	No_Telp	Jenis_Kelamin	Status_Pinjam
▶	14	Akhmad Sulthon Rabba	22	Sumenep	2003-08-04	2147483647	Laki-Laki	Sudah
	4	Imam Arishandi Irfan	22	Sumenep	2024-03-09	2147483647	Laki-Laki	Belum

5. Soal 5

	idAnggota	Nama_Anggota	Angkatan_Anggota	Tempat_Lahir_Anggota	Tanggal_Lahir_Anggota	No_Telp	Jenis_Kelamin	Status_Pinjam
▶	1	Angga Jelasena Saput	22	Pamekasan	2003-06-04	2147483647	Laki-Laki	Belum
	10	Romauli Napitupulu	22	Medan	2024-03-24	2147483647	Perempuan	Sudah
	11	M. Irfan Faruq Zain	22	Surabaya	2024-03-27	2147483647	Laki-Laki	Sudah
	12	Naela Nahdiah	22	Sidoarjo	2024-03-16	2147483647	Perempuan	Sudah
	13	Oxa Indi Vixzio	22	Sidoarjo	2024-03-19	2147483647	Laki-Laki	Belum
	14	Akhmad Sulthon Rabba	22	Sumenep	2003-08-04	2147483647	Laki-Laki	Sudah
	2	Della Ayu Vernandah	22	Sidoarjo	2024-03-25	2147483647	Perempuan	Sudah
	3	Amelia Zalfa Sahira	22	Sampang	2024-03-17	2147483647	Perempuan	Sudah
	4	Imam Arishandi Irfan	22	Sumenep	2024-03-09	2147483647	Laki-Laki	Belum
	5	Dwi Nurjannah	22	Lumajang	2024-03-02	2147483647	Perempuan	Belum
	6	Viko Alfian P	22	Gresik	2024-03-14	2147483647	Laki-Laki	Belum
	7	Tesa Putri Hutagaol	22	Medan	2024-03-29	2147483647	Perempuan	Sudah
	8	Bintang Wahyu Ariyon	22	Sidoarjo	2024-03-22	2147483647	Laki-Laki	Belum
	9	Khoirotun Nisak	22	Sampang	2024-03-15	2147483647	Perempuan	Belum

6. Soal 6

	TotalAnggota
▶	14

7. Soal 7

	@nama	@tahunTerbit	TotalBuku
▶	imam	2020	20

4.4 Penjelasan

Jadi di sini saya membuat beberapa stored procedure sesuai dengan soal yang di berikan, yang Dimana Langkah pertama saya membuat databasenya lalu saya isi, setelahnya saya membuat stored procedure dengan beberapa parameter IN, OUT, INOUT. Lalu setelah saya membuat lalu saya panggil menggunakan sql CALL.

BAB V

PENUTUP

5.1 Analisa

Penggunaan stored procedure dalam sistem manajemen basis data (DBMS) mencakup berbagai aspek, mulai dari keuntungan hingga pertimbangan yang perlu dipertimbangkan. Salah satu keuntungan utama dari penggunaan stored procedure adalah peningkatan kinerja aplikasi. Dengan menjalankan logika bisnis di dalam database, stored procedure mengurangi jumlah data yang harus dikirimkan antara server database dan aplikasi, mengurangi overhead jaringan, dan mempercepat waktu respons.

Namun, ada beberapa pertimbangan yang perlu diperhatikan dalam penggunaan stored procedure. Salah satunya adalah kompleksitas pengembangan dan pemeliharaan. Stored procedure memerlukan pengetahuan yang mendalam tentang bahasa pemrograman yang digunakan di dalam database, serta struktur data dan skema basis data yang kompleks.

5.2 Kesimpulan

Penggunaan stored procedure dalam sistem manajemen basis data (DBMS) menawarkan sejumlah manfaat yang signifikan, termasuk peningkatan kinerja aplikasi, pengurangan lalu lintas jaringan, tingkat abstraksi yang lebih tinggi untuk pengolahan data, dan dukungan terhadap transaksi kompleks.