**LAPORAN RESMI**

**MODUL III**

**STORED PROCEDURE**

**SISTEM MANAJEMEN BASIS DATA**



**NAMA : Agus Putra Singkih**

**N.R.P : 200442200153**

**DOSEN : Fitri Damayanti, S.Kom., M.Kom.**

**ASISTEN : Muhammad Iqbal Firmansyah**

**TGL PRAKTIKUM : Jum’at, 26 April 2024**

**Disetujui : .. ……… 2024**

**Asisten**

**Muhammad Iqbal Firmansyah**

**21.04.411.00084**

**LABORATORIUM BISNIS INTELIJEN SISTEM**

**PRODI SISTEM INFORMASI**

**JURUSAN TEKNIK INFORMATIKA**

**FAKULTAS TEKNIK**

**UNIVERSITAS TRUNOJOYO MADURA**

# **BAB I PENDAHULUAN**

## **Latar Belakang**

Procedure adalah program yang disimpan dalam database sebagai objek database. Meskipun disimpan sebagai objek database, hal ini sebenarnya cukup tidak umum, karena biasanya yang disimpan dalam database adalah data bukan program. Stored Procedure adalah sebuah kelompok kode SQL yang disimpan di katalog database dan dapat dipanggil kemudian oleh program, trigger, atau bahkan stored procedure lainnya. Sebuah Stored Procedure yang memanggil dirinya sendiri disebut rekursif stored procedure.

Adanya dukungan Stored Procedure akan membuat pengembangan program lebih ringkas dan mudah. Stored procedure juga dapat membantu dalam membuat laporan yang sifatnya analisa data, yang biasanya memerlukan banyak sekali tabel pembantu. Selain itu, stored procedure juga bisa mengefisienkan proses, sehingga semua daya hanya akan dipusatkan di komputer server saja, dan tidak terjadi transfer data yang berlebihan antara server dan client. Oleh karena itu, penggunaan stored procedure sangat direkomendasikan untuk aplikasi atau sistem yang memiliki kompleksitas tinggi dan membutuhkan kinerja yang optimal.

Selain itu, Stored Procedure juga dapat meningkatkan keamanan database. Karena stored procedure hanya dapat diakses oleh pengguna yang memiliki hak akses tertentu, sehingga dapat mengurangi risiko keamanan yang terkait dengan akses langsung ke basis data oleh pengguna yang tidak berwenang.

Selain itu, stored procedure juga dapat membantu dalam memudahkan pemeliharaan sistem, karena perubahan pada kode yang ada hanya perlu dilakukan pada satu tempat saja, yaitu di dalam stored procedure tersebut, tanpa perlu melakukan perubahan pada banyak script atau aplikasi yang menggunakan kode tersebut. Hal ini tentunya akan memudahkan dalam melakukan maintenance pada sistem.

Namun, penggunaan stored procedure juga memiliki beberapa kelemahan, di antaranya adalah sulit dalam debugging dan testing, serta sulit dalam melakukan version control. Oleh karena itu, sebelum menggunakan stored procedure, penting untuk mempertimbangkan dengan matang kelebihan dan kekurangannya terlebih dahulu.

## **Tujuan**

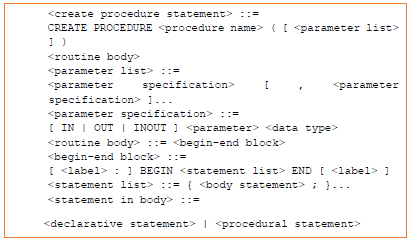
1. Mampu memahami dan membuat procedure dalam basis data dan mampu menggunakan perintah-perintah dalam stored procedure serta menyelesaikan operasi – operasi data spesifik dengan memanfaatkan stored procedure.

# **BAB II DASAR TEORI**

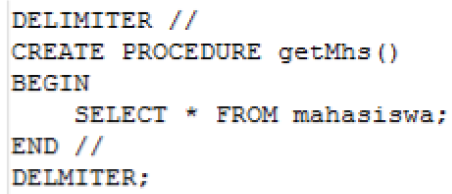
## **Dasar Teori**

### **Stored Procedure**

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 : mengingkatakan performa, mereduksi trafik jaringan, reusable, dan meningkatkan kontrol sekuriti. Di balik kelebihan tersebut, stored procedure juga memiliki kekurangan.

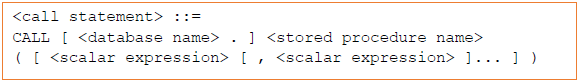
Sintaks stored procedure :

Pernyataan pembuatan stored procedure berikut :

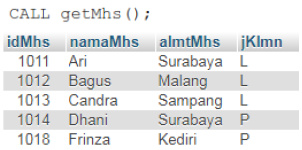


Perintah DELIMITER digunakan untuk mengubah delimiter standar, misalnya di sini dari titik koma (;) menjadi slash ganda (//). Langkah ini umumnya dilakukan ketika isi stored procedure mengandung titik koma – yang merupakan delimiter standar di SQL. Pernyataan di antara BEGIN dan END merupakan badan (body)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

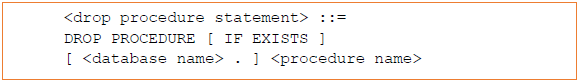
Aktivasi/pemanggilan Stored Procedure :



Eksekusi Query tersebut dengan memanggil procedure getMahasiswa().



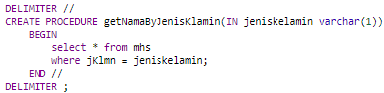
Menghapus Stored Procedure :



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

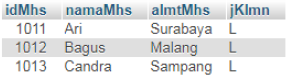
### **Parameter IN**

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.

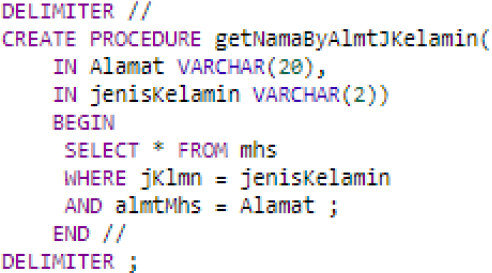


Untuk memanggil stored procedure yang memiliki parameter, maka kita harus menspesifikasikan argumenya. Misalkan kita ingin mendapatkan data mahasiswa dengan jenis kelamin laki-laki.





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 :

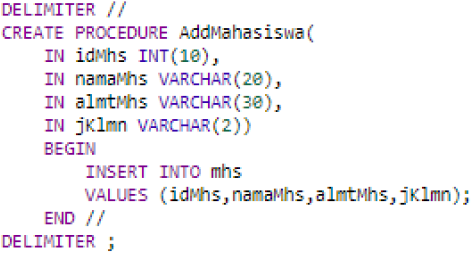


Pemanggilan stored procedure di atas tentunya akan memerlukan dua buah argumen.



Penambahan Data

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



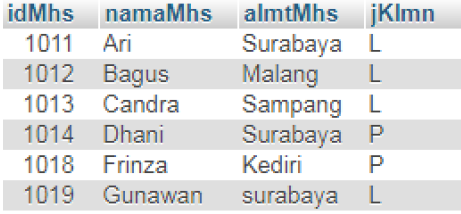
Lakukan eksekusi terhadap procedure tersebut

call AddMahasiswa('1019','Gunawan','surabaya', ‘L’);

Selanjutnya lakukan pengecekan data pada tabel mahasiswa.

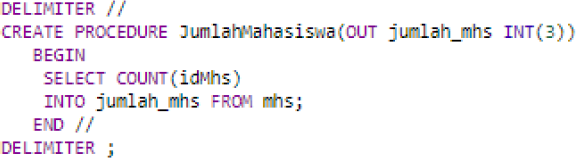
select \* from mhs; atau

select \* getMhs();



### **Parameter 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 diubaholeh stored procedure.



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.

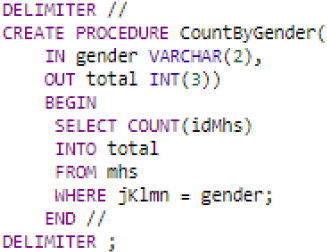
select @jumlah\_mhs;



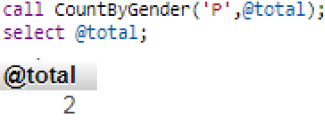
### **Parameter INOUT**

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 :

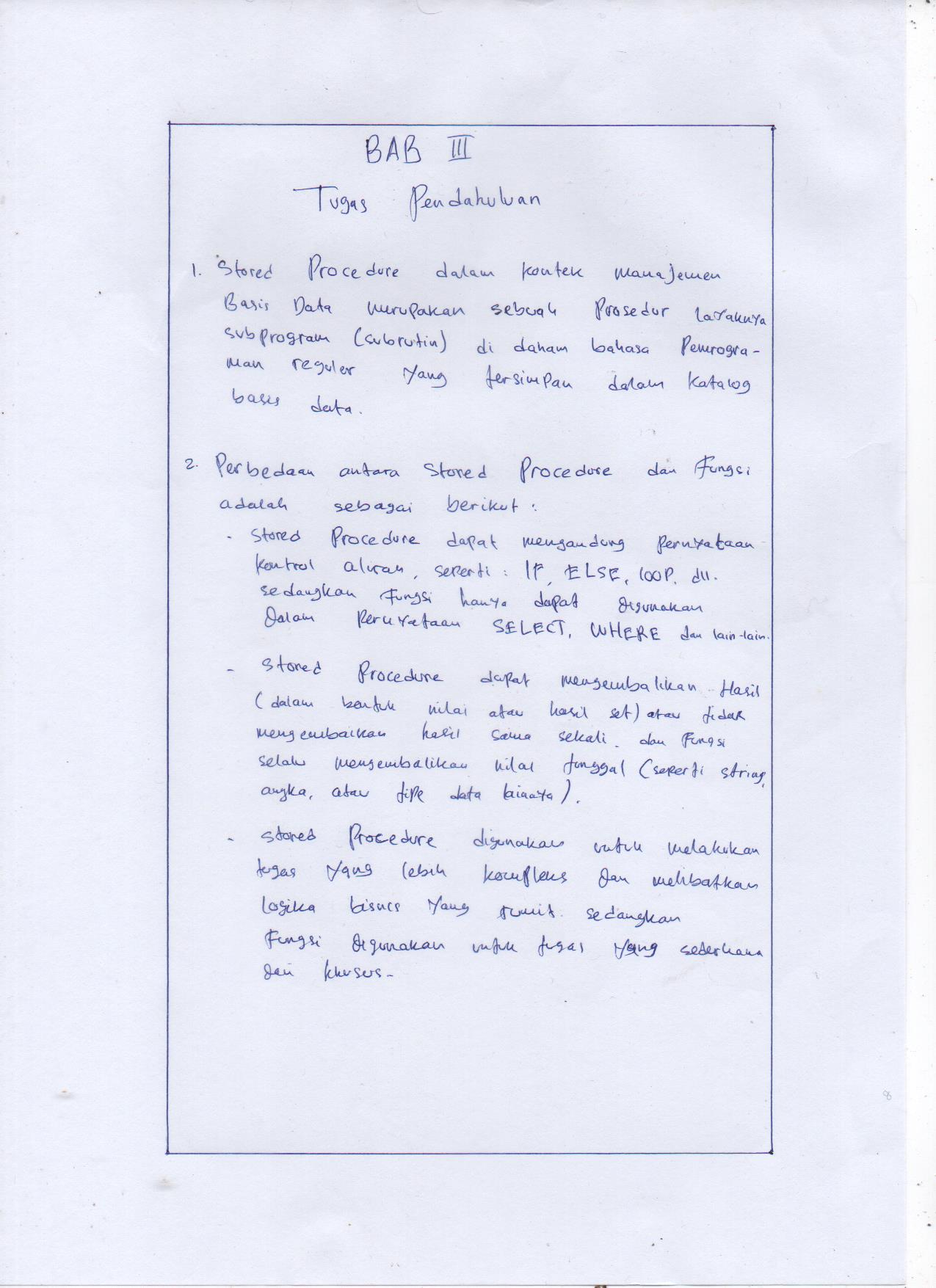


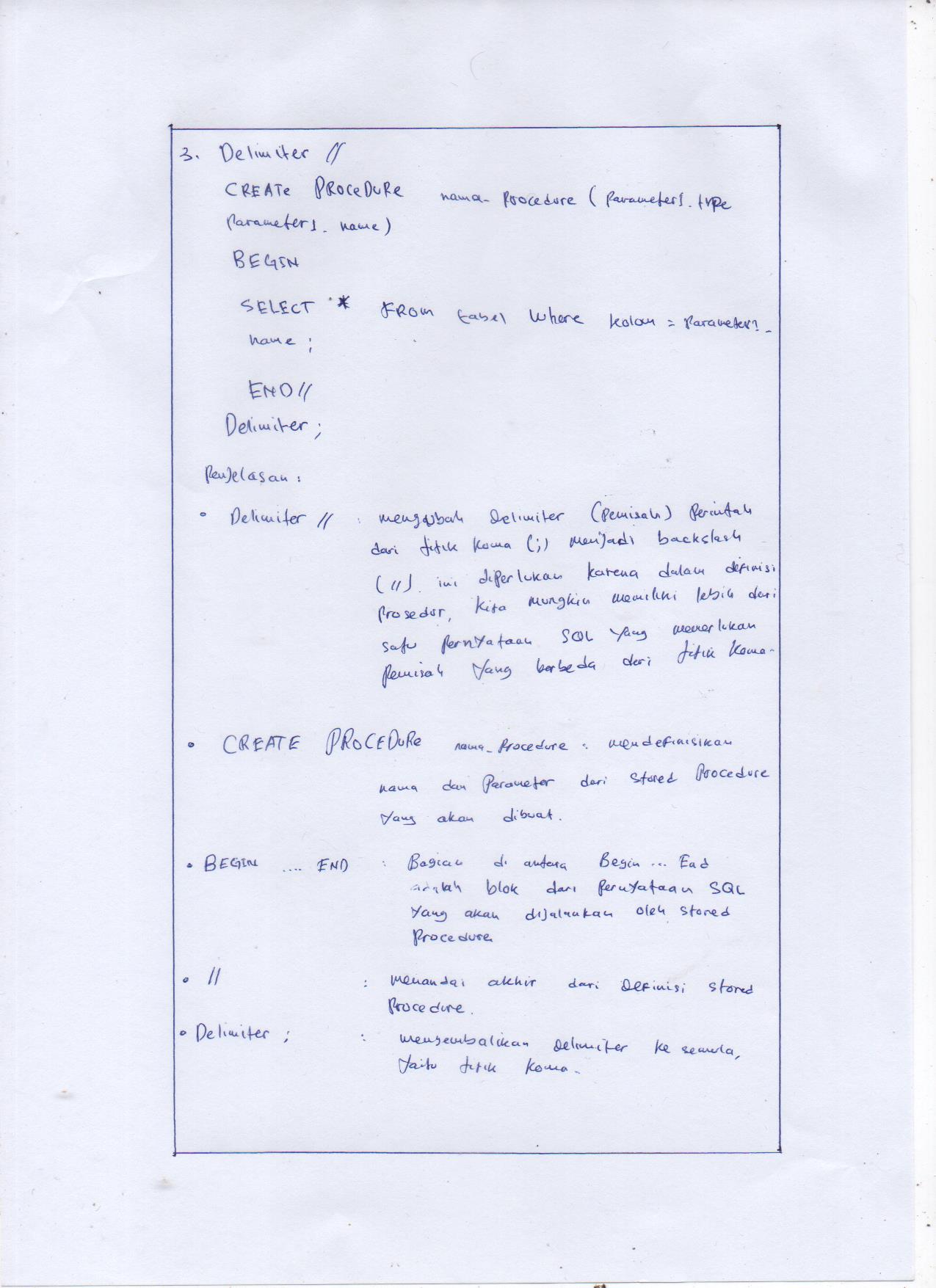
Lakukan eksekusi pada procedure tersebut untuk mencari jumlah mahasiswa yang berjenis kelamin perempuan .

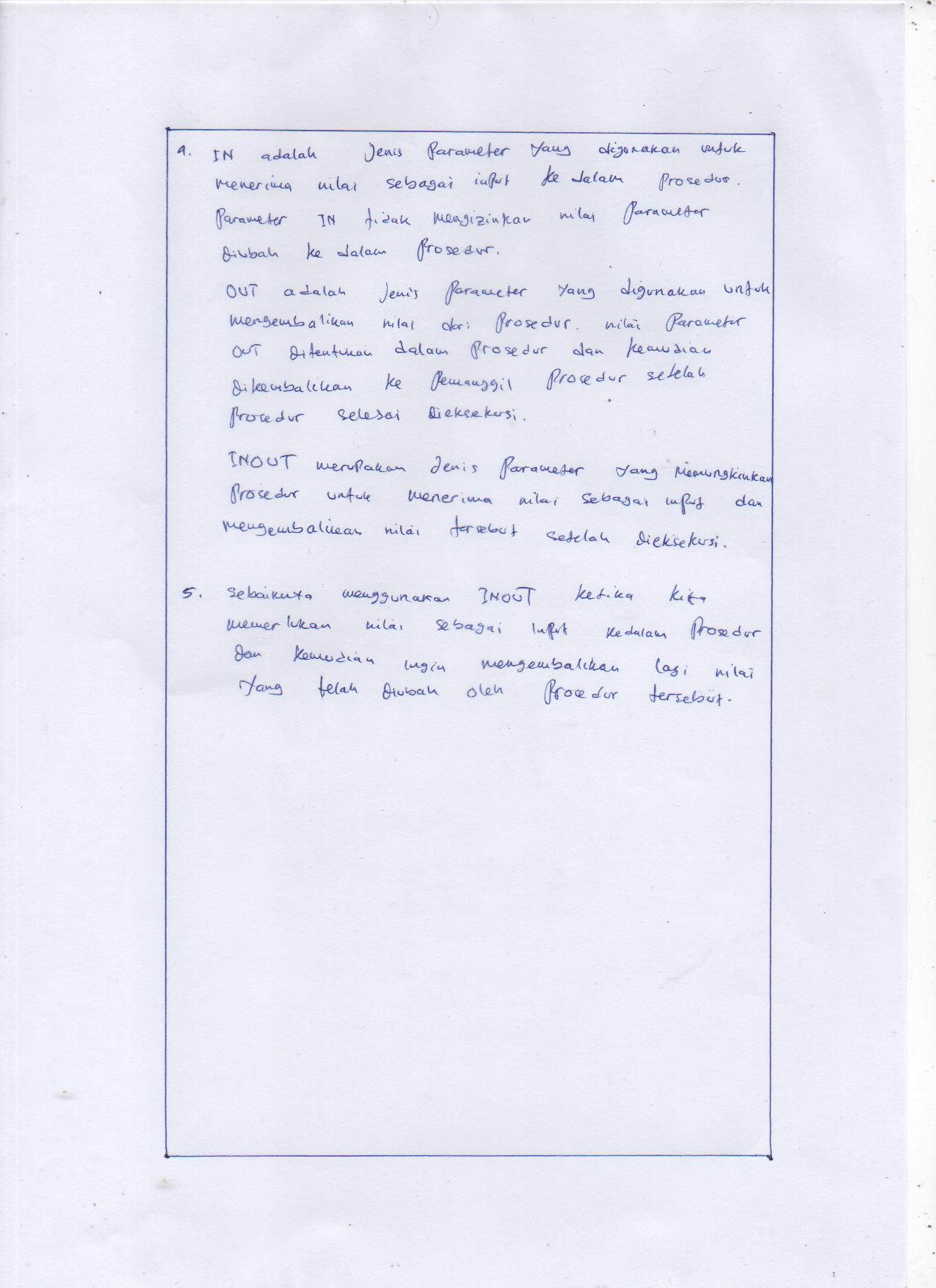


Stored procedure dapat mencerminkan beragam operasi data, misalnya seleksi, penambahan, pengubahan, penghapusan, dan juga operasi – oprasi DDL. Seperti halnya procedure di dalam bahasa pemrograman, stored procedure juga dapat melibatkan variabel, pernyataan kondisional, dan pengulangan.

# **BAB III TUGAS PENDAHULUAN**

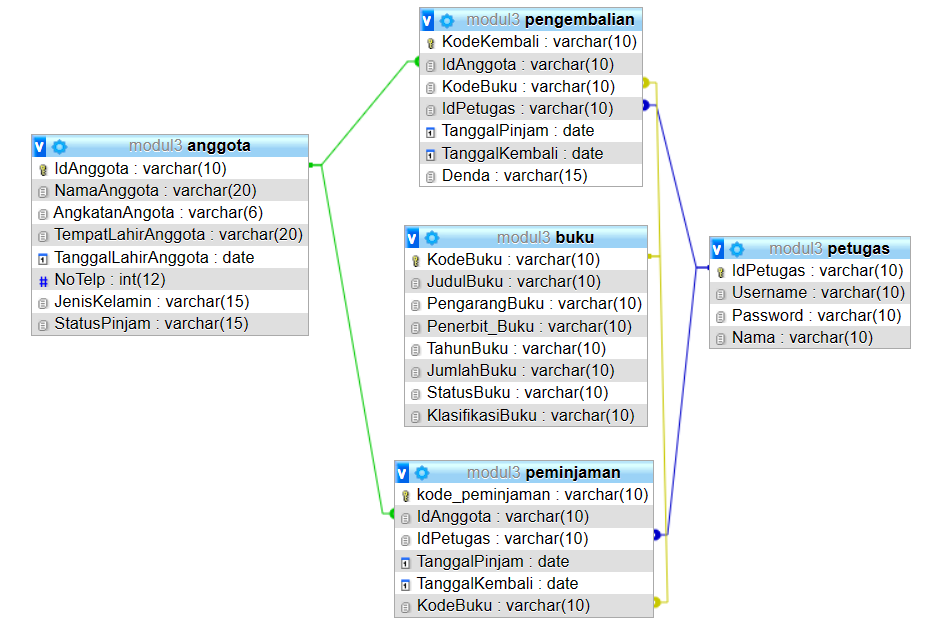
****

****

****

# **BAB IV IMPLEMENTASI**

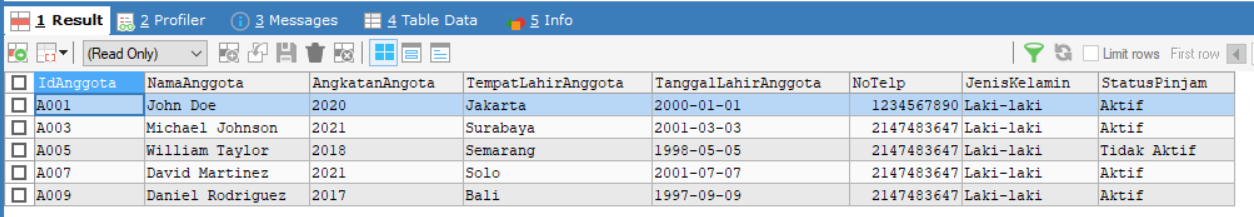
# **Implementasi**



1. Buatlah sebuah prosedur dengan menggunakan parameter IN default!

Query :

|  |
| --- |
| DELIMITER //  CREATE PROCEDURE SearchAnggotaByjkl(  IN p\_JenisKelamin VARCHAR(10)  )  BEGIN  SELECT \* FROM anggota WHERE JenisKelamin = p\_JenisKelamin;  END //  DELIMITER ;  CALL SearchAnggotaByjkl('Laki-laki'); |

Output :

Penjelasan :

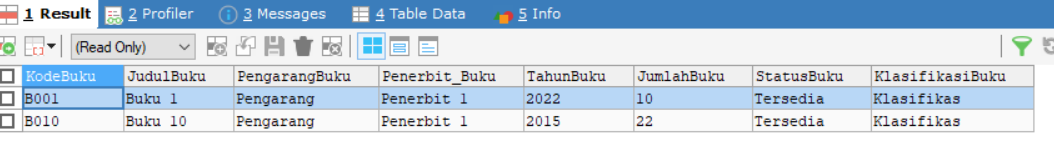
Pada query diatas saya mengawali dengan delimeter, lalu membuat create procedure untuk membuat suatu stored procedure baru dengan nama SearchAnggotaByjkl menggunakan parameter IN bernama p\_JenisKelamin. Kemudian, dibagian stored procedure BEGIN dan END terdapat perintah untuk mencari jenis kelamin berdasarkan jenis\_kelamin. Selanjutnya saya memanggil stored procedure menggunakan CALL dengan nama prosedurnya yaitu SearchAnggotaByjkl dan parameter berisi value yang ingin di cari yaitu ‘Laki-Laki’.

1. Definisikan stored procedure untuk mengetahui data pada table buku dengan berdasarkan pada salah satu atribut/field yaitu penerbit buku!

Query :

|  |
| --- |
| DELIMITER//  CREATE PROCEDURE getBuku (IN  penerbit VARCHAR(20))  BEGIN  SELECT \* FROM buku WHERE Penerbit\_Buku = penerbit;  END//  DELIMITER;  CALL getBuku ('Penerbit 1'); |

Output :



Penjelasan :

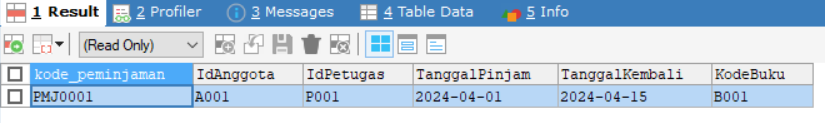
Pada query diatas saya mengawali dengan delimeter, lalu membuat create procedure untuk membuat suatu stored procedure baru dengan nama getBuku menggunakan parameter IN bernama penerbit. Kemudian, dibagian stored procedure BEGIN dan END terdapat perintah untuk mencari nama petugas berdasarkan PenerbitBuku. Selanjutnya saya memanggil stored procedure menggunakan CALL dengan nama prosedurnya yaitu getBuku dan parameter berisi value yang ingin di cari yaitu ‘Penerbit 1.

1. Definisikan stored procedure untuk mengetahui data pada table peminjaman berdasarkan dua field yaitu tanggal pinjam dan kode buku!

Query :

|  |
| --- |
| DELIMITER//  CREATE PROCEDURE getPeminjaman (IN tgl DATE, IN kb VARCHAR(5))  BEGIN  SELECT \* FROM peminjaman  WHERE TanggalPinjam = tgl  AND KodeBuku = kb;  END//  CALL getPeminjaman('2024-04-01', 'B001'); |

Ouput :



Penjelasan :

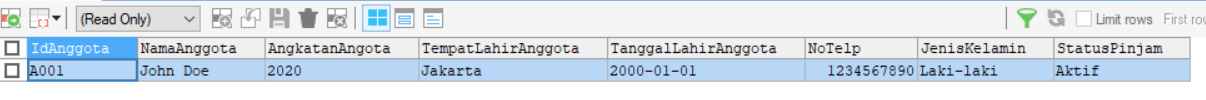
Pada query diatas saya mengawali dengan delimeter, lalu membuat create procedure untuk membuat suatu stored procedure baru dengan nama getPeminjaman menggunakan parameter IN bernama tgl dan kb. Kemudian, dibagian stored procedure BEGIN dan END terdapat perintah untuk mencari anggota berdasarkan TanggalPinjam dan KodeBuku. Selanjutnya saya memanggil stored procedure menggunakan CALL dengan nama prosedurnya yaitu getPeminjaman dan parameter berisi value yang ingin di cari yaitu ‘2024-04-01’, ‘B001’.

1. Buatlah Stored Procedure untuk mengetahui data pada table anggota berdasarkan 3 field yaitu angkatan, tempat lahir, dan jenis kelamin!

Query :

|  |
| --- |
| DELIMITER//  CREATE PROCEDURE getanggota (IN akt VARCHAR(5), IN tl VARCHAR(20),  IN jk VARCHAR (10))  BEGIN  SELECT \* FROM anggota WHERE AngkatanAngota = akt  AND TempatLahirAnggota = tl  AND JenisKelamin = jk;  END//  CALL getanggota ('2020', 'Jakarta', 'Laki-laki'); |

Output :



Penjelasan :

Pada query diatas saya mengawali dengan delimeter, lalu membuat create procedure untuk membuat suatu stored procedure baru dengan nama getanggota menggunakan parameter IN bernama akt, tl dan jk. Kemudian, dibagian stored procedure BEGIN dan END terdapat perintah untuk mencari anggota berdasarkan AngakatanAnggota, TempatLahirAnggota dan JenisKelamin. Selanjutnya saya memanggil stored procedure menggunakan CALL dengan nama prosedurnya yaitu getanggota dan parameter berisi value yang ingin di cari yaitu ‘2020’, ‘Jakarta, ‘laki-laki’.

1. Definisikan stored procedure untuk memasukkan data pada table anggota!

Query :

|  |
| --- |
| DELIMITER //  CREATE PROCEDURE inputAnggota (  IN IdAnggota VARCHAR(10),  IN NamaAnggota VARCHAR(20),  IN AngkatanAngota VARCHAR(6),  IN TempatLahirAnggota VARCHAR(20),  IN TanggalLahirAnggota DATE,  IN NoTelp INT(12),  IN JenisKelamin VARCHAR(15),  IN StatusPinjam VARCHAR(15))  BEGIN  INSERT INTO anggota VALUES(  IdAnggota, NamaAnggota, AngkatanAngota, TempatLahirAnggota, TanggalLahirAnggota,  NoTelp, JenisKelamin, StatusPinjam);  END//  CALL inputAnggota ('A011', 'Agus Putra Singkih', '2020', 'Bangkalan',  '2000-07-07', '087755660522', 'Laki-laki', 'Aktif');  SELECT \* FROM anggota; |

Output :



Penjelasan :

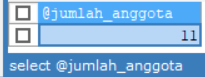
Pada query diatas saya mengawali dengan delimeter, lalu membuat create procedure untuk membuat suatu stored procedure baru dengan nama inputanggota menggunakan parameter IN bernama IdAnggota, NamaAnggota, AngkatanAngota, TempatLahirAnggota, TanggalLahirAnggota, NoTelp, JenisKelamin, dan StatusPinjam. Kemudian, dibagian stored procedure BEGIN dan END terdapat perintah untuk menambah data baru pada tabel anggota. Selanjutnya saya memanggil stored procedure menggunakan CALL dengan nama prosedurnya yaitu inputanggota dan parameter berisi value yang ingin di cari yaitu 'A011', 'Agus Putra Singkih', '2020', 'Bangkalan', '2000-07-07', '087755660522', 'Laki-laki', 'Aktif'’.

1. Definisikan stored procedure untuk mengetahui data jumlah anggota menggunakan parameter OUT!

Query :

|  |
| --- |
| DELIMITER//  CREATE PROCEDURE jumlahanggota(OUT jumlah\_anggota INT(20))  BEGIN  SELECT COUNT(IdAnggota)  INTO jumlah\_anggota FROM anggota;  END//  CALL jumlahanggota(@jumlah\_anggota);  SELECT @jumlah\_anggota |

Output :



Penjelasan :

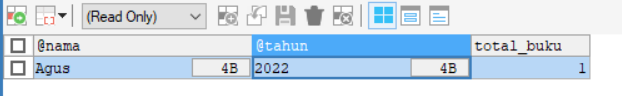
Pada query diatas saya mengawali dengan delimeter, lalu membuat create procedure untuk membuat suatu stored procedure baru dengan nama jumlahanggota menggunakan parameter OUT bernama jumlah\_anggota. Kemudian, dibagian stored procedure BEGIN dan END terdapat perintah untuk menghitung jumlah id\_anggota yang ada d tabel anggota. Selanjutnya saya memanggil stored procedure menggunakan CALL dengan nama prosedurnya yaitu jumlahanggota dengan value nama parameter yaitu @jumlah\_anggota.

1. Definisikan stored procedure untuk mengetahui data jumlah berdasarkan pengarang dan tahun terbit menggunakan parameter INOUT!

Query :

|  |
| --- |
| DELIMITER //  CREATE PROCEDURE jmlbuku (IN penulis VARCHAR(20), IN tahun VARCHAR(5), OUT total INT)  BEGIN  SELECT COUNT(KodeBuku) INTO total  FROM buku  WHERE PengarangBuku = penulis AND TahunBuku = tahun;  END//  DELIMITER ;  SET @nama = 'Agus';  SET @tahun = '2022';  CALL jmlbuku(@nama, @tahun, @total);  SELECT @nama, @tahun, @total AS total\_buku; |

Output :



Penjelasan :

Pada query diatas saya mengawali dengan delimeter, lalu membuat create procedure untuk membuat suatu stored procedure baru dengan nama jmlbuku menggunakan parameter IN bernama penulis, tahun dan OUT bernama total. Kemudian, dibagian stored procedure BEGIN dan END terdapat perintah untuk menghitung jumlah buku berdasarkan pengarang da tahun buku yang ada di tabel buku. Selanjutnya saya memanggil stored procedure menggunakan CALL dengan nama prosedurnya yaitu jmlbuku dengan value nama parameter yaitu @total.

## **Penjelasan Hasil Praktikum**

Berdarsakan hasil praktikum yang dilakukan, Stored Procedure merupakan langkah tepat ketika kita menangani proyek besar, seperti memanggil jumlah data dalam sebuah table. Dengan menggunakan Stored Procedure syntax yang hendak ditulis menjadi lebih singkat dan lebih mudah dipahami. Kemudian dengan mengetahui fungsi parameter IN, OUT, INOUT dapat mempermudah dalam menentukan jenis parameter yang digunakan dalam menyelesaikan tugas praktikum kita.

# **BAB V PENUTUP**

## **Analisa**

Dari hasil praktikum, praktikan menganalisa bahwa Stored Procedure lebih efisien dari perintah biasa, karena procedure disimpan di server pada saat dibuat dan dijalankan di server. Stored Procedure yang berisi sekumpulan perintah Transact-SQL akan dipanggil oleh sebuah perintah Transact-SQL. Sebelum Stored Procedure dibuat, sintaks penulisan perintah dicek kebenarannya. Jika tidak Error, nama procedure disimpan di tabel SysObjects dan teksnya disimpan di tabel SysComments. Pertama kali Stored Procedure dijalankan, rencana eksekusi dibuat dan Stored Procedure dikompilasi. Pemrosesan berikutnya akan lebih cepat, karena SQL Server tidak perlu mengecek sintaks dan membuat rencana eksekusi atau mengkompilasi procedure.

## **Kesimpulan**

Stored Procedure adalah sebuah kelompok kode SQL yang di simpan di katalog database dan dapat di panggil kemudian oleh program. Adanya dukungan Stored Procedure akan membuat program lebih ringkas dan mudah untuk dikembangkan. Stored procedure akan membantu dalam membuat laporan yang sifatnya analisa data, yang biasanya memerlukan banyak sekali tabel pembantu. Stored procedure mengefisienkan proses, sehingga semua daya hanya akan dipusatkan di komputer server saja. Di MySQL, parameter stored procedure dibedakan menjadi tiga : IN, OUT, dan INOUT.