

**LAPORAN HASIL AKHIR PRAKTIKUM
BASIS DATA II**



Nama : Cristivioni
NIM : 203030503094
Kelas : C
Modul : (I) Stored Procedure dan Function

JURUSAN TEKNIK INFORMATIKA
FAKULTAS TEKNIK
UNIVERSITAS PALANGKA RAYA
2022

Nama : Cristivioni
NIM : 203030503094
Kelas : C
Modul : (I) Stored Procedure dan Function

Komposisi	Nilai	Max
Sampul Depan	5	
Bab I Tujuan dan Landasan Teori	30	
Bab II Langkah Kerja dan Pembahasan	30	
Bab III Kesimpulan	25	
Bab IV Daftar Pustaka	10	
TOTAL	100	

Penilai Asisten Praktikum

Muhammad Arifin

BAB I

TUJUAN DAN LANDASAN TEORI

A. Tujuan

1. Mahasiswa dapat memahami manfaat *Stored Procedure* dan mampu membuat *Stored Procedure*.
2. Mahasiswa memahami manfaat *Functions* dan mampu membuat *Functions*.

A. Landasan Teori

1. Stored Procedure

Stored Procedure merupakan berbagai pernyataan atau prosedur SQL yang tersimpan di dalam database. Stored Procedure bisa menerima parameter sehingga Anda dapat menulis pernyataan IF, melakukan pengaturan pada variabel, dan lain sebagainya. Berikut ini adalah beberapa kelebihanannya:

- **Performa yang Lebih Baik**

Stored Procedure dikompilasi dan kemudian di-cache dan digunakan lagi. Sehingga waktu respon akan menjadi lebih cepat ketika stored procedure yang sama dieksekusi lagi.

- **Modular**

Jika beberapa aplikasi ingin menggunakan query yang sama, maka dengan cara tradisional Anda menduplikasi kode yang tidak perlu pada aplikasi, cara terbaik adalah menempatkan kode ke dalam database.

- **Keamanan Lebih Baik**

Aplikasi yang memanfaatkan pernyataan sql yang dibuat secara dinamis sangat rentan terhadap serangan injeksi SQL, sedangkan stored procedure dapat menghindari serangan injeksi SQL sepenuhnya.

Adapun kelemahan dari Stored Procedure adalah sebagai berikut:

- Peningkatan beban pada database server. Sebagian besar pekerjaan dilakukan di sisi server, dan kurang pada sisi client.
- Anda harus mempelajari sintaks pernyataan MySQL untuk menulis stored procedures.
- Jika Anda mengulangi logika aplikasi Anda di dua tempat yang berbeda: kode bahasa pemrograman dan kode stored procedures, membuat pemeliharaan aplikasi akan menjadi lebih sulit.

- Migrasi ke sistem manajemen basis data yang berbeda (DB2, SQL Server, dll) berpotensi menjadi lebih sulit.

2. Function

Function adalah suatu bagian dari program yang digunakan untuk mengerjakan suatu tugas tertentu yang mengembalikan suatu nilai untuk dikembalikan ke program pemanggil yang letaknya dipisahkan dari bagian program yang menggunakannya. Didalam mysql sudah terdapat beberapa fungsi default yang dapat digunakan untuk mengerjakan tugas tertentu. Misalnya untuk mencari nilai minimal, menghitung rata-rata dan menjumlahkan suatu nilai. Secara umum untuk membuat function formatnya adalah sebagai berikut.

```
DELIMITER $$  
CREATE  
    PROCEDURE `nama database`.`nama_function`()  
BEGIN  
    ... .  
END$$  
DELIMITER ;
```

Return pada function mendefinisikan tipe data yang akan menampung hasil atau nilai yang akan dikembalikan oleh fungsi ke sistem. Adapun statemen yang bisa digunakan dalam function adalah: set, while, if, declare, select, insert, update, dan delete.

Adapun cara pemanggilan function tidak bisa menggunakan perintah CALL seperti ketika memanggil procedure. Secara umum untuk memanggil function dapat menggunakan format berikut.

```
SELECT nama_function();
```

BAB II
LANGKAH KERJA DAN PEMBAHASAN

Pada praktikum ini mahasiswa akan membuat Stored Procedure dan Function pada database toko sepatu. Berikut ini adalah pembahasannya.

1. Membuat database toko sepatu

Pada bagian ini mahasiswa diminta untuk membuat database yang diberi nama toko_sepatu dengan tabel sebagai berikut.

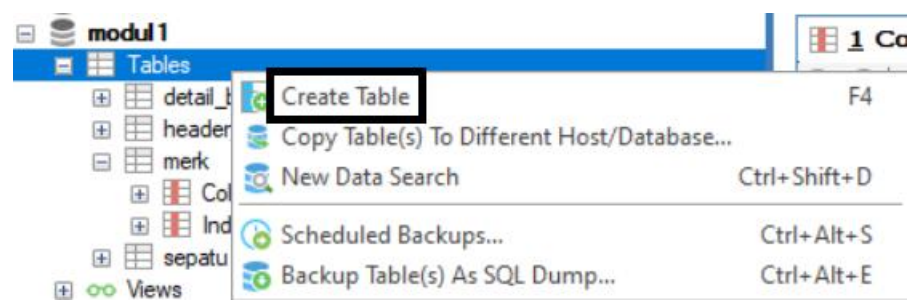
Merk	Sepatu	Detail_bayar	Header_bayar
Id_merk	Id_sepatu	Id_detail	No_nota
Nama_merk	Id_merk	Id_sepatu	Tanggal
Model_sepatu	Ukuran	Jumlah_beli	Id_detail
	Warna		Total_pembelian
	Harga		Bayar
	stok		Sisa_bayar

Dengan ketentuan:

- a. Merk:
 - Id_merk adalahh Primary Key
- b. Sepatu:
 - Id_sepatu adalah Primary Key
 - Id_merk adalah Foreign Key dari tabel merk
 - Stok adalah jumlah sepatu yang tersedia di toko
- c. Detail_bayar: tabel yang mencatat item-item pembelian sepatu oleh pelanggan.
 - Id_detail adalah Primary Key
 - Id_sepatu adalah Foreign Key dari tabel sepatu
 - Jumlah_beli adalah jumlah atau banyaknya sepatu yang dibeli oleh pelanggan
- d. Header_bayar:
 - No_nota adalah Primary Key
 - Tanggal adalah tanggal transaksi dilakukan
 - Id_detail adalah Foreign Key dari tabel detail_bayar
 - Total_pembelian adalah jumlah_beli x harga
 - Bayar adalah sejumlah uang yang dibayarkan oleh pelanggan.
 - Sisa_bayar adalah sisa uang pembayaran pelanggan (bayar-total_pembayaran)

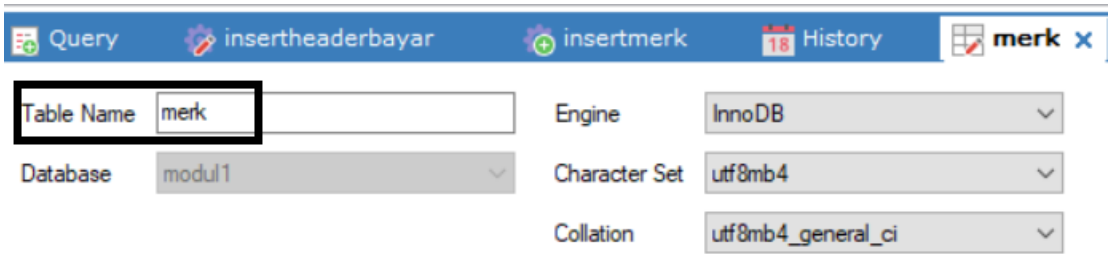
2. Berikut ini langkah kerja pembuatan Tabel pada SQLyog:

- a. Langkah pertama dalam membuat tabel kita adalah klik kanan pada “Tables” lalu, Create Table. Berikut tampilan seperti di bawah ini.



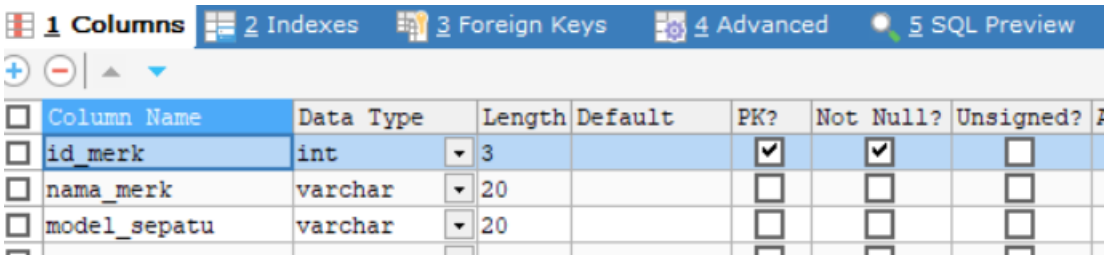
Gambar 2.1. Create Table

- b. Untuk memberi nama tabel bisa dilihat pada tampilan dibawah ini.



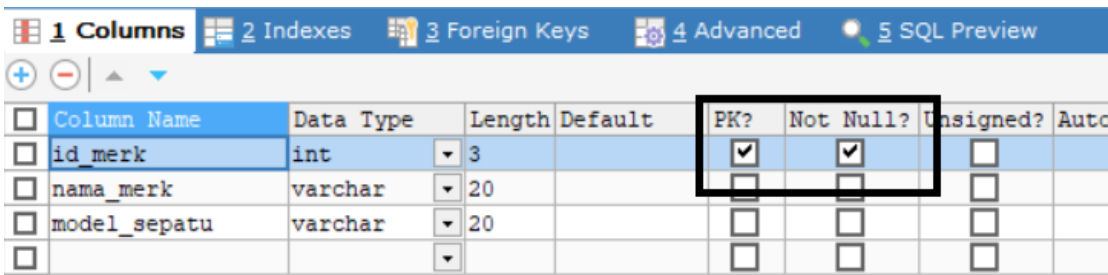
Gambar 2.2. Nama_Tabel

- c. Setelah memberikan nama_tabel kita dapat mengisi isi tabel yang dan menentukan type data yang digunakan serta panjang maximum yang bisa digunakan. Berikut tampilan seperti dibawah ini:



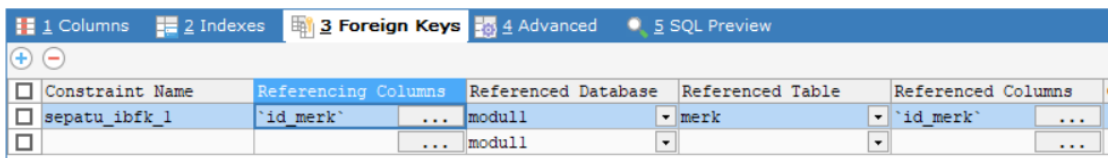
Gambar 2.3. Nama_Column dan Type Data

- d. Untuk menentukan Primary Key, dengan cara klik satu kali centang pada PK yang ada pada Columns. Berikut tampilan seperti dibawah ini:



Gambar 2.4. Primary Key

- e. Untuk menentukan Foreign Key, dengan cara klik opsi 3.Foreign Key kemudian pilih referencing column dan referencing tabel yang ingin dijadikan foreign key. Berikut tampilan seperti dibawah ini:



Gambar 2.5. Foreign Key

3. Query untuk Insert data

Langkah pertama adalah membuat 4 bagian Stored Procedure untuk tabel merk, sepatu, detail_bayar dan header_bayar. Caranya adalah Klik Kanan pada Stored Procedure kemudian pilih Create Stored Procedure, lalu beri nama pada Procedure.

3.1. Insert merk

Berikut dibawah ini adalah Query dari Insert Merk:

```
1 DELIMITER $$
2 USE `modull`$$
3 DROP PROCEDURE IF EXISTS `insertmerk`$$
4 CREATE DEFINER=`root`@`localhost` PROCEDURE `insertmerk`(id_merk INT , nama_merk VARCHAR , model_sepatu VARCHAR)
5 BEGIN
6     INSERT INTO merk VALUES (id_merk, nama_merk, model_sepatu);
7 END$$
8 DELIMITER ;
```

Gambar 2.6. Query Insert Merk

Berikut ini adalah salah satu query untuk melakukan insert data pada tabel merk

```
INSERT INTO merk VALUES (1100,'nike','air jordan');
```

Berikut ini adalah data tabel merk pada SQLyog setelah dilakukan insert data merk.

	id_merk	nama_merk	model_sepatu
	1100	nike	running
	1200	nike	air jordan
	1300	rebbook	olahraga
	1400	tomkins	sepatu sekolah
	1500	adidas	olahraga
*	(NULL)	(NULL)	(NULL)

Gambar 2.7. Data Tabel Merk

3.2. Insert Sepatu

Berikut dibawah ini adalah Query dari Insert Sepatu:

```
1 DELIMITER $$
2 USE `modull`$$
3 DROP PROCEDURE IF EXISTS `insertsepatu`$$
4 CREATE DEFINER=`root`@`localhost` PROCEDURE `insertsepatu`(id_sepatu INT , id_merk INT , ukuran INT , warna VARCHAR (20), harga INT , stok INT )
5 BEGIN
6     INSERT INTO sepatu VALUES (id_sepatu, id_merk, ukuran, warna, harga, stok);
7 END$$
8 DELIMITER ;
```

Gambar 2.8. Query Insert Sepatu

Berikut ini adalah salah satu query untuk melakukan insert data pada tabel sepatu.

```
INSERT INTO sepatu VALUES (2100,1100,36,'ungu'150000,5);
```

Berikut ini adalah data tabel sepatu pada SQLyog setelah dilakukan insert data sepatu.

id_sepatu	id_merk	ukuran	warna	harga	stok
2100	1100	36	ungu	150000	5
2200	1200	37	hitam	200000	5
2300	1300	38	putih	250000	5
2400	1400	39	abu	300000	5
2500	1500	40	merah	350000	5
*	(NULL)	(NULL)	(NULL)	(NULL)	(NULL)

Gambar 2.9. Data Tabel Sepatu

3.3. Insert Detail_Bayar

Berikut dibawah ini adalah Query dari Insert detail bayar:

```
1 DELIMITER $$
2 USE `modull`$$
3 DROP PROCEDURE IF EXISTS `insertdetailbayar`$$
4 CREATE DEFINER=`root`@`localhost` PROCEDURE `insertdetailbayar`(id_detail INT , id_sepatu INT , jumlah_beli INT )
5 BEGIN
6     INSERT INTO detail_bayar VALUES (id_detail, id_sepatu, jumlah_beli);
7 END$$
8 DELIMITER ;
```

Gambar 3.0. Query Insert Detail_bayar

Berikut ini adalah salah satu query untuk melakukan insert data pada tabel detail bayar.

```
INSERT INTO detail_bayar VALUES (3100,2100,2);
```

Berikut ini adalah data tabel detail_bayar pada SQLyog setelah dilakukan insert data detail bayar.

id_detail	id_sepatu	jumlah_beli
3100	2100	2
3200	2200	3
3300	2300	1
3400	2400	2
3500	2500	2
*	(NULL)	(NULL)

Gambar 3.1. Data Tabel Detail_bayar

3.4. Insert Header Bayar

Berikut dibawah ini adalah Query Insert header bayar:

```
1 DELIMITER $$
2 USE `modull`$$
3 DROP PROCEDURE IF EXISTS `insertheaderbayar`$$
4 CREATE DEFINER=`root`@`localhost` PROCEDURE `insertheaderbayar`
5     (no_nota INT ,
6      tanggal DATE ,
7      id_detail INT ,
8      total_pembelian INT ,
9      bayar INT, sisa_bayar INT)
10 BEGIN
11     INSERT INTO header_bayar VALUES
12     (no_nota, tanggal, id_detail, total_pembelian, bayar, sisa_bayar);
13 END$$
14 DELIMITER ;
```

Gambar 3.2. Query Insert Header Bayar

Berikut ini adalah salah satu query untuk melakukan insert data pada tabel header bayar.

```
INSERT INTO header_bayar VALUES
(4100,20220215,3100,300000,300000,0);
```

Berikut ini adalah data tabel headerbayar setelah dilakukan insert data header bayar.

no_nota	tanggal	id_detail	total_pembelian	bayar	sisa_bayar
4100	2022-02-15	3100	300000	300000	0
4200	2022-02-17	3200	600000	600000	0
4300	2022-02-20	3300	250000	300000	50000
4400	2022-02-23	3400	600000	600000	0
4500	2022-02-25	3500	700000	700000	0
(Auto)	(NULL)	(NULL)	(NULL)	(NULL)	(NULL)

Gambar 3.3. Data Tabel Header Bayar

4. Query untuk Delete

4.1. Delete Merk

Berikut dibawah ini adalah Query delete merk:

```
1 DELIMITER $$
2
3 USE `modull`$$
4
5 DROP PROCEDURE IF EXISTS `deletemerk`$$
6
7 CREATE DEFINER=`root`@`localhost` PROCEDURE `deletemerk` (idmerk INT )
8 BEGIN
9     DELETE FROM merk WHERE id_merk = idmerk;
10 END$$
11
12 DELIMITER ;
```

Gambar 3.4. Query Delete Merk

Berikut ini adalah salah satu query untuk melakukan delete data pada tabel merk.

```
CALL `deletemerk` (1500);
```

Berikut ini adalah data setelah dilakukan delete data pada tabel merk.

<input type="checkbox"/>	id_merk	nama_merk	model_sepatu
<input type="checkbox"/>	1100	nike	running
<input type="checkbox"/>	1200	nike	air jordan
<input type="checkbox"/>	1300	rebbook	olahraga
<input type="checkbox"/>	1400	tomkins	sepatu sekolah
*	(NULL)	(NULL)	(NULL)

Gambar 3.5.

4.2. Delete Sepatu

Berikut ini adalah Query delete sepatu:

```
1  DELIMITER $$
2
3  USE `modull`$$
4
5  DROP PROCEDURE IF EXISTS `deletesepatu`$$
6
7  CREATE DEFINER=`root`@`localhost` PROCEDURE `deletesepatu`(idsepatu INT )
8  BEGIN
9      DELETE FROM sepatu WHERE id_sepatu = idsepatu;
10 END$$
11
12 DELIMITER ;
```

Gambar 3.6. Query Delete Sepatu

Berikut ini adalah salah satu query untuk melakukan delete data pada tabel sepatu.

```
CALL `deletesepatu`(2500)
```

Berikut ini adalah data setelah dilakukan delete data pada tabel sepatu.

<input type="checkbox"/>	id_sepatu	id_merk	ukuran	warna	harga	stok
<input type="checkbox"/>	2100	1100	36	ungu	150000	5
<input type="checkbox"/>	2200	1200	37	hitam	200000	5
<input type="checkbox"/>	2300	1300	38	putih	250000	5
<input type="checkbox"/>	2400	1400	39	abu	300000	5
*	(NULL)	(NULL)	(NULL)	(NULL)	(NULL)	(NULL)

Gambar 3.7.

4.3. Delete Detail Bayar

Berikut ini adalah Query delete detail bayar:

```
1  DELIMITER $$
2
3  USE `modull`$$
4
5  DROP PROCEDURE IF EXISTS `deletedetailbayar`$$
6
7  CREATE DEFINER=`root`@`localhost` PROCEDURE `deletedetailbayar`(iddetail INT )
8  BEGIN
9      DELETE FROM detail_bayar
10     WHERE id_detail = iddetail;
11 END$$
12
13 DELIMITER ;
```

Gambar 3.6.

Berikut ini adalah salah satu query untuk melakukan delete pada tabel detail bayar.

```
CALL `deletedetailbayar`(3500);
```

Berikut ini adalah data setelah dilakukan delete data pada tabel detail bayar.

	id_detail	id_sepatu	jumlah_beli
	3100	2100	2
	3200	2200	3
	3300	2300	1
	3400	2400	2
*	(NULL)	(NULL)	(NULL)

Gambar 3.7.

4.4. Delete Header Bayar

Berikut ini adalah Query delete header bayar:

```
1 DELIMITER $$
2
3 USE `modull`$$
4
5 DROP PROCEDURE IF EXISTS `deletedetailbayar`$$
6
7 CREATE DEFINER=`root`@`localhost` PROCEDURE `deletedetailbayar`(iddetail INT )
8 BEGIN
9     DELETE FROM detail_bayar
10    WHERE id_detail = iddetail;
11 END$$
12
13 DELIMITER ;
```

Gambar 3.8. Query Delete Header Bayar

Berikut ini adalah salah satu query untuk melakukan delete pada tabel header bayar.

```
CALL `deleteheaderbayar`(4500);
```

Berikut ini adalah data setelah dilakukan delete pada tabel header bayar.

	no_nota	tanggal	Delete selected row(s) (Alt+Del)		embelian	bayar	sisabayar
	4100	2022-02-15	3100		300000	300000	0
	4200	2022-02-17	3200		600000	600000	0
	4300	2022-02-20	3300		250000	300000	50000
	4400	2022-02-23	3400		600000	600000	0
*	(Auto)	(NULL)	(NULL)		(NULL)	(NULL)	(NULL)

Gambar 3.9.

5. Query untuk Update

5.1. Update Merk

Berikut ini adalah Query update pada tabel merk:

```
1 DELIMITER $$
2
3 USE `modull`$$
4
5 DROP PROCEDURE IF EXISTS `updatemerk`$$
6
7 CREATE DEFINER=`root`@`localhost` PROCEDURE `updatemerk`( idmerk INT, nama_merk VARCHAR (20), model_sepatu VARCHAR (20) )
8 BEGIN
9     UPDATE merk
10    SET nama_merk = nama_merk, model_sepatu = model_sepatu WHERE id_merk = idmerk;
11 END$$
12
13 DELIMITER ;
```

Gambar 4.0. Query Update Merk

Berikut ini adalah salah satu query untuk melakukan update pada tabel merk.

```
call `updatemerk`(1100,'adidas','olahraga');
```

Berikut ini adalah data setelah dilakukan update pada tabel merk.

<input type="checkbox"/>	id_merk	nama_merk	model_sepatu
<input type="checkbox"/>	1100	adidas	olahraga
<input type="checkbox"/>	1200	nike	air jordan
<input type="checkbox"/>	1300	rebbook	olahraga
<input type="checkbox"/>	1400	tomkins	sepatu sekolah
<input type="checkbox"/>	1500	adidas	olahraga
<input type="checkbox"/>	*	(NULL)	(NULL)

Gambar 4.1.

5.2. Update Sepatu

Berikut ini adalah Query update pada tabel sepatu:

```
1 DELIMITER $$
2
3 USE `modull`$$
4
5 DROP PROCEDURE IF EXISTS `updatesepatu`$$
6
7 CREATE DEFINER=`root`@`localhost` PROCEDURE `updatesepatu`(idsepatu INT, ukuran INT, warna VARCHAR (20) )
8 BEGIN
9     UPDATE sepatu
10    SET ukuran = ukuran, warna = warna WHERE id_sepatu = idsepatu;
11 END$$
12
13 DELIMITER ;
```

Gambar 4.2. Query Update Sepatu

Berikut ini adalah salah satu query untuk melakukan update pada tabel sepatu.

```
call `updatesepatu`(2100,38,'hitam putih');
```

Berikut ini adalah data setelah dilakukan update pada tabel sepatu.

<input type="checkbox"/>	id_sepatu	id_merk	ukuran	warna	harga	stok
<input type="checkbox"/>	2100	1100	38	hitam putih	150000	5
<input type="checkbox"/>	2200	1200	37	hitam	200000	5
<input type="checkbox"/>	2300	1300	38	putih	250000	5
<input type="checkbox"/>	2400	1400	39	abu	300000	5
<input type="checkbox"/>	2500	1500	40	merah	350000	5
<input type="checkbox"/>	*	(NULL)	(NULL)	(NULL)	(NULL)	(NULL)

Gambar 4.3.

5.3. Update Detail Bayar

Berikut ini adalah Query update pada tabel detail bayar:

```
1 DELIMITER $$
2
3 USE `modul1`$$
4
5 DROP PROCEDURE IF EXISTS `updatedetailbayar`$$
6
7 CREATE DEFINER=`root`@`localhost` PROCEDURE `updatedetailbayar`(IN id_detail INT (3), IN jumlah_beli VARCHAR (3) )
8 BEGIN
9     UPDATE detail_bayar
10    SET jumlah_beli = jumlah_beli WHERE id_detail = id_detail;
11 END$$
12
13 DELIMITER ;
```

Gambar 4.4. Query Update Detail Bayar

Berikut ini adalah salah satu query untuk melakukan update pada tabel detail bayar.

```
CALL `updatedetailbayar`(3100,3);
```

Berikut ini adalah data setelah dilakukan update pada tabel detail bayar.

id_detail	id_sepatu	jumlah_beli
3100	2100	3
3200	2200	3
3300	2300	1
3400	2400	2
3500	2500	2
(NULL)	(NULL)	(NULL)

Gambar 4.5.

5.4. Update Header Bayar

Berikut ini adalah Query update pada tabel header bayar:

```
1 DELIMITER $$
2
3 USE `modul1`$$
4
5 DROP PROCEDURE IF EXISTS `updateheader_bayar`$$
6
7 CREATE DEFINER=`root`@`localhost` PROCEDURE `updateheader_bayar`(Nonota INT , bayar INT, tanggal DATE ,total_pembelian INT )
8 BEGIN
9     UPDATE header_bayar
10    SET bayar = bayar , tanggal = tanggal , total_pembelian = total_pembelian WHERE no_nota = Nonota;
11 END$$
12
13 DELIMITER ;
```

Gambar 4.6. Query Update Header Bayar

Berikut ini adalah salah satu query untuk melakukan update pada tabel header bayar.

```
call `updateheader_bayar`(4100, 310000,10000);
```

Berikut ini adalah data seteleha dilakukan update pada tabel header bayar.

no_nota	tanggal	id_detail	total_pembelian	bayar	sisabayar
4100	2022-02-15	3100	300000	310000	10000
4200	2022-02-17	3200	600000	600000	0
4300	2022-02-20	3300	250000	300000	50000
4400	2022-02-23	3400	600000	600000	0
(Auto)	(NULL)	(NULL)	(NULL)	(NULL)	(NULL)

Gambar 4.7.

5.5. Laporan Total Pemasukan

Berikut ini adalah Query Function Laporan Total Pemasukan

```
1  DELIMITER $$
2
3  CREATE
4
5      FUNCTION `modul1`.`Laporan_Total` (tanggal DATE )
6      RETURNS VARCHAR (100)
7
8  BEGIN
9      DECLARE
10         total INT(20); SELECT SUM(hb.total_pembelian)
11         FROM header_bayar WHERE hb.tanggal = tanggal
12         INTO total;
13
14         RETURN CONCAT
15         ('Tanggal = ',tanggal,' || Total Pemasukan :',total);
16
17     END$$
18
19 DELIMITER ;
```

Gambar 4.8. Laporan Total Pemasukan

BAB III

KESIMPULAN

Setelah saya mempelajari modul 1, dapat saya simpulkan, yaitu:

- Stored Procedure merupakan berbagai pernyataan atau prosedur SQL yang tersimpan di dalam database.
- Sebuah stored Procedure dikompilasi dan lebih cepat dalam mengeksekusi query atau batch.
- Return merupakan tipe data yang akan menampung hasil atau nilai yang akan dikembalikan oleh fungsi ke sistem.
- Function adalah suatu bagian dari program yang digunakan untuk mengerjakan suatu tugas tertentu yang mengembalikan suatu nilai untuk dikembalikan ke program pemanggil yang letaknya dipisahkan dari bagian program yang menggunakannya.
- Return pada function mendefinisikan tipe data yang akan menampung hasil atau nilai yang akan dikembalikan oleh fungsi ke sistem. Adapun statemen yang bisa digunakan dalam function adalah: set, while, if, declare, select, insert, update, dan delete.

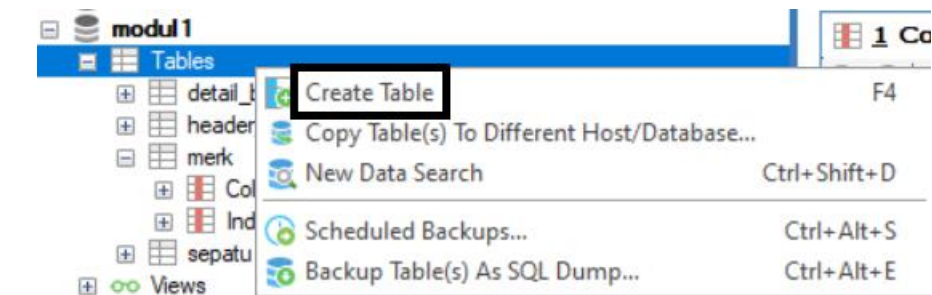
BAB IV

DAFTAR PUSTAKA

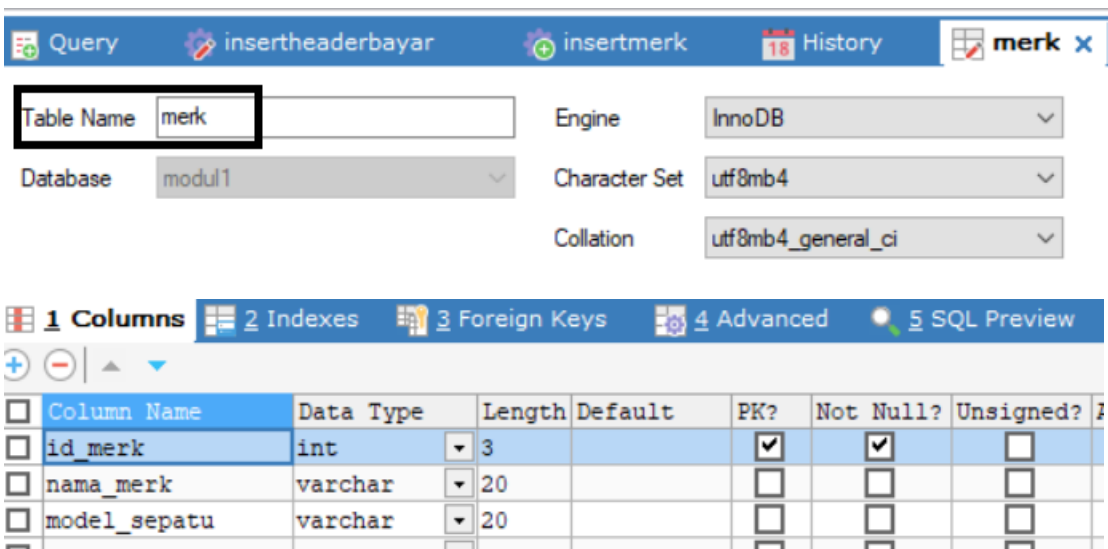
- Modul Praktikum Fakultas Teknik Informatika Universitas Palangka Raya tahun 2022
- Yudana. 2022. *Apa Itu Stored Procedure? Bagaimana Menggunakannya Di MySQL?*. [online] Available at: <<https://www.yudana.id/tutorial-membuat-stored-procedure-dengan-menggunakan-mysql/>> [Accessed 1 March 2022].
- Nesabamedia. 2022. *Apa itu Stored Procedure? Mengenal Stored Procedure Pada Database*. [online] Available at: <<https://www.nesabamedia.com/stored-procedure-pada-database/>> [Accessed 1 March 2022].
- Tutor-All Programming. 2022. *Stored Procedure, Trigger, dan Function dengan MySQL*. [online] Available at: <<https://tutorallprogramming.blogspot.com/2017/11/stored-procedure-trigger-dan-function.html>> [Accessed 1 March 2022].

BAB V

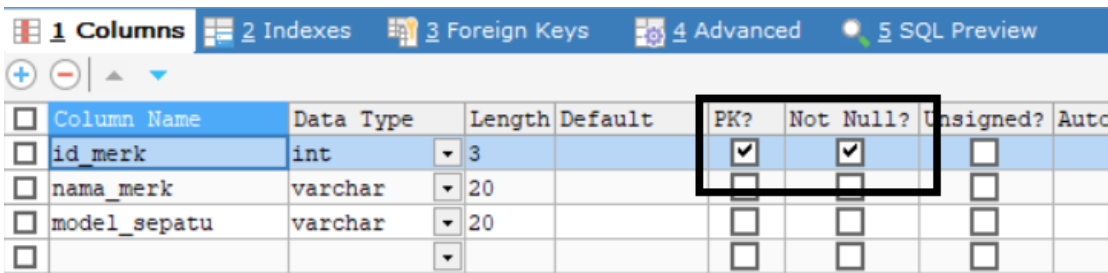
LAMPIRAN



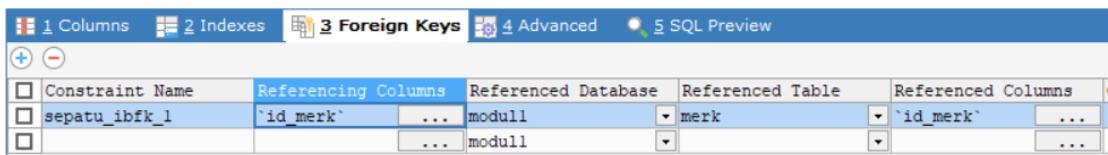
Gambar 2.1. Create Table



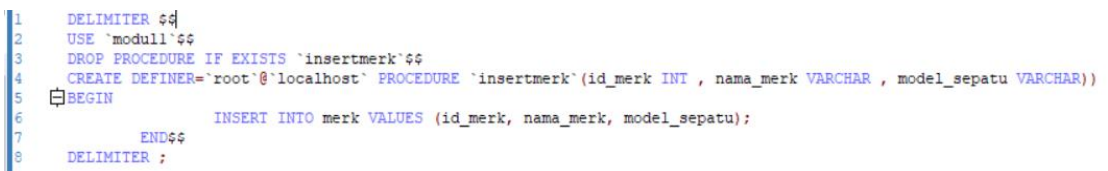
Gambar 2.3. Nama_Column dan Type Data



Gambar 2.4. Primary Key



Gambar 2.5. Foreign Key



Gambar 2.6. Query Insert Merk

id_merk	nama_merk	model_sepatu
1100	nike	running
1200	nike	air jordan
1300	rebbook	olahraga
1400	tomkins	sepatu sekolah
1500	adidas	olahraga
*	(NULL)	(NULL)

Gambar 2.7. Data Tabel Merk

```

1 DELIMITER $$
2 USE `modul1`$$
3 DROP PROCEDURE IF EXISTS `insertsepatu`$$
4 CREATE DEFINER=`root`@`localhost` PROCEDURE `insertsepatu`(id_sepatu INT , id_merk INT , ukuran INT , warna VARCHAR (20), harga INT , stok INT )
5 BEGIN
6     INSERT INTO sepatu VALUES (id_sepatu, id_merk, ukuran, warna, harga, stok);
7 END$$
8 DELIMITER ;

```

Gambar 2.8. Query Insert Sepatu

id_sepatu	id_merk	ukuran	warna	harga	stok
2100	1100	36	ungu	150000	5
2200	1200	37	hitam	200000	5
2300	1300	38	putih	250000	5
2400	1400	39	abu	300000	5
2500	1500	40	merah	350000	5
*	(NULL)	(NULL)	(NULL)	(NULL)	(NULL)

Gambar 2.9. Data Tabel Sepatu

```

1 DELIMITER $$
2 USE `modul1`$$
3 DROP PROCEDURE IF EXISTS `insertdetailbayar`$$
4 CREATE DEFINER=`root`@`localhost` PROCEDURE `insertdetailbayar`(id_detail INT , id_sepatu INT , jumlah_beli INT )
5 BEGIN
6     INSERT INTO detail_bayar VALUES (id_detail, id_sepatu, jumlah_beli);
7 END$$
8 DELIMITER ;

```

Gambar 3.0. Query Insert Detail_bayar

id_detail	id_sepatu	jumlah_beli
3100	2100	2
3200	2200	3
3300	2300	1
3400	2400	2
3500	2500	2
*	(NULL)	(NULL)

Gambar 3.1. Data Tabel Detail_bayar

```

1 DELIMITER $$
2 USE `modul1`$$
3 DROP PROCEDURE IF EXISTS `insertheaderbayar`$$
4 CREATE DEFINER=`root`@`localhost` PROCEDURE `insertheaderbayar`
5 (
6     no_nota INT ,
7     tanggal DATE ,
8     id_detail INT ,
9     total_pembelian INT ,
10    bayar INT, sisa_bayar INT)
11 BEGIN
12     INSERT INTO header_bayar VALUES
13     (no_nota, tanggal, id_detail, total_pembelian, bayar, sisa_bayar);
14 END$$
15 DELIMITER ;

```

Gambar 3.2. Query Insert Header Bayar

<input type="checkbox"/>	no_nota	tanggal	id_detail <input type="checkbox"/>	total_pembelian	bayar	sisabayar
<input type="checkbox"/>	4100	2022-02-15	3100	300000	300000	0
<input type="checkbox"/>	4200	2022-02-17	3200	600000	600000	0
<input type="checkbox"/>	4300	2022-02-20	3300	250000	300000	50000
<input type="checkbox"/>	4400	2022-02-23	3400	600000	600000	0
<input type="checkbox"/>	4500	2022-02-25	3500	700000	700000	0
*	(Auto)	(NULL)	(NULL)	(NULL)	(NULL)	(NULL)

Gambar 3.3. Data Tabel Header Bayar

```

1  DELIMITER $$
2
3  USE `modull`$$
4
5  DROP PROCEDURE IF EXISTS `deletemerk`$$
6
7  CREATE DEFINER=`root`@`localhost` PROCEDURE `deletemerk`(idmerk INT )
8  BEGIN
9      DELETE FROM merk WHERE id_merk = idmerk;
10
11      END$$
12  DELIMITER ;

```

Gambar 3.4. Query Delete Merk

<input type="checkbox"/>	id_merk	nama_merk	model_sepatu
<input type="checkbox"/>	1100	nike	running
<input type="checkbox"/>	1200	nike	air jordan
<input type="checkbox"/>	1300	rebbook	olahraga
<input type="checkbox"/>	1400	tomkins	sepatu sekolah
*	(NULL)	(NULL)	(NULL)

Gambar 3.5.

```

1  DELIMITER $$
2
3  USE `modull`$$
4
5  DROP PROCEDURE IF EXISTS `deletedetailbayar`$$
6
7  CREATE DEFINER=`root`@`localhost` PROCEDURE `deletedetailbayar`(iddetail INT )
8  BEGIN
9      DELETE FROM detail_bayar
10      WHERE id_detail = iddetail;
11
12      END$$
13  DELIMITER ;

```

Gambar 3.6.

<input type="checkbox"/>	id_detail	id_sepatu <input type="checkbox"/>	jumlah_beli
<input type="checkbox"/>	3100	2100	2
<input type="checkbox"/>	3200	2200	3
<input type="checkbox"/>	3300	2300	1
<input type="checkbox"/>	3400	2400	2
*	(NULL)	(NULL)	(NULL)

Gambar 3.7.

```
1 DELIMITER $$
2
3 USE `modull`$$
4
5 DROP PROCEDURE IF EXISTS `deletedetailbayar`$$
6
7 CREATE DEFINER=`root`@`localhost` PROCEDURE `deletedetailbayar`(iddetail INT )
8 BEGIN
9     DELETE FROM detail_bayar
10    WHERE id_detail = iddetail;
11 END$$
12
13 DELIMITER ;
```

Gambar 3.8. Query Delete Header Bayar

no_nota	tanggal	embelian	bayar	sisabayar
4100	2022-02-15	3100	300000	300000
4200	2022-02-17	3200	600000	600000
4300	2022-02-20	3300	250000	300000
4400	2022-02-23	3400	600000	600000
(Auto)	(NULL)	(NULL)	(NULL)	(NULL)

Gambar 3.9.

```
1 DELIMITER $$
2
3 USE `modull`$$
4
5 DROP PROCEDURE IF EXISTS `updatemerk`$$
6
7 CREATE DEFINER=`root`@`localhost` PROCEDURE `updatemerk`( idmerk INT, nama_merk VARCHAR (20), model_sepatu VARCHAR (20) )
8 BEGIN
9     UPDATE merk
10    SET nama_merk = nama_merk, model_sepatu = model_sepatu WHERE id_merk = idmerk;
11 END$$
12
13 DELIMITER ;
```

Gambar 4.0. Query Update Merk

id_merk	nama_merk	model_sepatu
1100	adidas	olahraga
1200	nike	air jordan
1300	rebbook	olahraga
1400	tomkins	sepatu sekolah
1500	adidas	olahraga
(NULL)	(NULL)	(NULL)

Gambar 4.1.

```
1 DELIMITER $$
2
3 USE `modull`$$
4
5 DROP PROCEDURE IF EXISTS `updatesepatu`$$
6
7 CREATE DEFINER=`root`@`localhost` PROCEDURE `updatesepatu`(idsepatu INT, ukuran INT, warna VARCHAR (20) )
8 BEGIN
9     UPDATE sepatu
10    SET ukuran = ukuran, warna = warna WHERE id_sepatu = idsepatu;
11 END$$
12
13 DELIMITER ;
```

Gambar 4.2. Query Update Sepatu

id_sepatu	id_merk	ukuran	warna	harga	stok
2100	1100	38	hitam putih	150000	5
2200	1200	37	hitam	200000	5
2300	1300	38	putih	250000	5
2400	1400	39	abu	300000	5
2500	1500	40	merah	350000	5
(NULL)	(NULL)	(NULL)	(NULL)	(NULL)	(NULL)

Gambar 4.3.

```

1  DELIMITER $$
2
3  USE `modul1`$$
4
5  DROP PROCEDURE IF EXISTS `updatedetailbayar`$$
6
7  CREATE DEFINER=`root`@`localhost` PROCEDURE `updatedetailbayar`(IN id_detail INT (3), IN jumlah_beli VARCHAR (3) )
8  BEGIN
9      UPDATE detail_bayar
10     SET jumlah_beli = jumlah_beli WHERE id_detail = id_detail;
11 END$$
12
13 DELIMITER ;

```

Gambar 4.4. Query Update Detail Bayar

	id_detail	id_sepatu	jumlah_beli
	3100	2100	3
	3200	2200	3
	3300	2300	1
	3400	2400	2
	3500	2500	2
*	(NULL)	(NULL)	(NULL)

Gambar 4.5.

```

1  DELIMITER $$
2
3  USE `modul1`$$
4
5  DROP PROCEDURE IF EXISTS `updateheader_bayar`$$
6
7  CREATE DEFINER=`root`@`localhost` PROCEDURE `updateheader_bayar`(NOnota INT , bayar INT, tanggal DATE ,total_pembelian INT )
8  BEGIN
9      UPDATE header_bayar
10     SET bayar = bayar , tanggal = tanggal , total_pembelian = total_pembelian WHERE no_nota = Nonota;
11 END$$
12
13 DELIMITER ;

```

Gambar 4.6. Query Update Header Bayar

	no_nota	tanggal	id_detail	total_pembelian	bayar	sisabayar
	4100	2022-02-15	3100	300000	310000	10000
	4200	2022-02-17	3200	600000	600000	0
	4300	2022-02-20	3300	250000	300000	50000
	4400	2022-02-23	3400	600000	600000	0
*	(Auto)	(NULL)	(NULL)	(NULL)	(NULL)	(NULL)

Gambar 4.7.

```

1  DELIMITER $$
2
3  CREATE
4
5      FUNCTION `modul1`.`Laporan_Total`(tanggal DATE )
6      RETURNS VARCHAR (100)
7
8  BEGIN
9      DECLARE
10     total INT(20); SELECT SUM(hb.total_pembelian)
11     FROM header_bayar WHERE hb.tanggal = tanggal
12     INTO total;
13
14     RETURN CONCAT
15     ('Tanggal = ',tanggal,' || Total Pemasukan : ',total);
16
17 END$$
18
19 DELIMITER ;

```

Gambar 4.8. Laporan Total Pemasukan

