

Latihan Praktikum Sistem Basis Data

Pertemuan 1

- Instalalasi mySQL
- Sistem interface dengan PHP MyAdmin dan HeidiSQL
- Perintah dasar mySQL
 - ✓ Membuat dan menghapus database (db_akademik)

Pertemuan 2

- Mengetahui tipe data di MySQL
- Membuat dan menghapus tabel
- Membuat tabel relasi
- **Tugas**

Buatlah sebuah database dengan nama database =akademik, yang memuat minimal 3 tabel, Matakuliah, jadwal dan dosen, dan buatlah relasi antar tabelnya Dengan enterprise manager.

Tabel Matakuliah

Field	Tipe data	Allow null	keterangan
Kode_mk	Char(5)	No	Kode matakuliah/primary key
Nama_mk	Varchar(25)	No	Nama Matakuliah
Sks	Int	Yes	sks
Kode_prasyarat	Char(5)	Yes	Kode matakuliah prasyarat

Tabel Dosen

Field	Tipe data	Allow null	keterangan
Kode_dsn	Char(5)	No	Kode dosen/primary key
Nama_dsn	Varchar(25)	No	Nama dosen
Jk	Char (5)	Yes	Jenis kelamin (L/P)
Agama	Varchar(25)	Yes	agama
Nama_jalan	Varchar(50)	Yes	alamat
Kode_pos	Varchar(10)	Yes	Kode pos

Tabel Jadwal

Field	Tipe data	Allow null	keterangan
Kode_mk	Char(5)	No	Kode matakuliah /Foreign key
Kode_dsn	Char(5)	No	Kode dosen/Foreign Key
Hari	Varchar(15)	Yes	hari
Jam	dateTime	Yes	jam
Kelas	Varchar(10)	Yes	Nama kelas

Pertemuan 3

- Dasar-dasar DDL (Dynamic Data Language)
 - Create/ drop Database
 - Create/ drop Table
 - Alter Table

Latihan DDL

1. Buatlah database dengan nama_DB = RUMAH SAKIT
2. Buatlah table table dengan menggunakan sintaks SQL, adapun table yang terdapat didalamnya adalah sebagai berikut :

Tabel Pemeriksaan

No	Nama column	Type data	length	Keterangan
1	No_pendaftaran	Char	5	Primary key
2	No_pasien	Char	5	Foreign key
3	Tgl_pendaftaran	Datetime		
4	Tgl_periksa	datetime		

Tabel Dokter

No	Nama column	Type data	length	Keterangan
1	Kode_dokter	Char	5	Primary key
2	Nama_dokter	Varchar	25	

Tabel Pasien

No	Nama column	Type data	length	Keterangan
1	No_pasien	Char	5	Primary key
2	Nama_pasien	Varchar	25	
3	Alamat	Varchar	50	
4	Telepon	Varchar	12	

Tabel Resep

No	Nama column	Type data	length	Keterangan
1	No_pendaftaran	Char	5	Foreign key
2	Kode_dokter	Char	5	Foreign key
3	Kode_sakit	Char	5	Foreign key
4	Kode_obat	Char	5	Foreign key
5	Dosis	Varchar	25	

Tabel Obat

No	Nama column	Type data	length	Keterangan
1	Kode_obat	Char	5	Primary key
2	Nama_obat	Varchar	25	

Tabel Penyakit

No	Nama column	Type data	length	Keterangan
1	Kode_sakit	Char	5	Primary key
2	Nama_penyakit	Varchar	25	

3. Buatlah Relasi tabel dari tabel di atas

Pertemuan 4.

- DML
- Insert, update, delete

TUGAS DML

- Buatlah database baru dengan nama = DB_BANK , kemudian buat tabel-tabel di bawah ini serta inputkan isi tabelnya dengan menggunakan perintah SQL.
- Diketahui 2 tabel (nasabah dan cabang) beserta isi tabelnya,

Tabel Nasabah

No	Nama column	Type data	length	Keterangan
1	Kd_nasabah	Char	5	Primary key
2	Nama	Varchar	25	
3	Alamat	Varchar	50	
4	Rekening	Varchar	10	
5	Kd_cabang	Char	5	Foreign key
6	Saldo	Float		

Tabel Cabang

No	Nama column	Type data	length	Keterangan
1	Kd_cabang	Char	5	Primary key
2	Nama	Varchar	25	
3	Alamat	Varchar	50	
4	Jml_nasabah	Numeric	Not null	

Inputan data seperti dlm tabel di bawah ini

DATA CABANG

Kd_cabang	Nama	Alamat	Jml_nasabah
KC001	KC AMIKOM	CONDONG CATUR	8
KC002	KC JANTI	JANTI	5
KC003	KC GEJAYAN	GEJAYAN	6
KC004	KC WIROBRAJAN	WIROBRAJAN	8
KC005	KC KAUMAN	KAUMAN	7
KC006	KC IMOGIRI	IMOGIRI	6
KC007	KC BABARSARI	BABARSARI	3

DATA NASABAH

Kd_nasabah	Nama	Alamat	Rekening	Kd_cabang	saldo
N0001	JANUAR	KALASAN	12340001	KC002	150000
N0002	FEBRI	KUNCEN	12340002	KC004	250000
N0003	Maria	paris	12340003	KC006	300000
N0004	Aprillia	samirono	12340004	KC003	200000

N0005	Memey	seyegan	12340005	KC001	120000
N0006	Juno	gamping	12340006	KC004	280000
N0007	Julia	monjali	12340007	KC001	140000
N0008	Agusto	turi	12340008	KC001	220000
N0009	Septia	ngampilan	12340009	KC005	130000
N0010	Okta	sagan	12340010	KC003	270000
N0011	Nova	kotabaru	12340011	KC005	200000
N0012	Desi	terban	12340012	KC003	70000

Pertemuan 5.

Dengan menggunakan data pada pertemuan 4, lakukan hal-hal berikut

1. Buatlah perintah untuk menampilkan semua nama nasabah, alamat dan saldonya
2. Buatlah perintah untuk menampilkan nama nasabah dengan saldo > 200000
3. Buatlah perintah untuk menampilkan nama nasabah dengan alamat gejayan
4. Buatlah perintah untuk menampilkan nama nasabah dengan kode cabang KC003
5. Buatlah perintah untuk menampilkan semua data kantor cabang
6. Buatlah perintah untuk menampilkan nama kantor cabang dengan alamat kauman
7. Buatlah perintah untuk menampilkan nama kantor cabang dengan jumlah nasabah > 6 orang
8. Buatlah perintah untuk mengupdate data kantor cabang dengan kode KC002 menjadi
Nama = KC RSJ pakem
Alamat =pakem
Jumlah nasabah = 7
9. Buatlah perintah untuk menghapus data semua kantor cabang dengan jumlah nasabah kurang dari 5
10. Buatlah perintah untuk mengupdate data nasabah dengan kode N0010 menjadi
Alamat = monjali
Saldo =400000
11. Buatlah perintah untuk menghapus semua data nasabah dengan alamat terban

Pertemuan 6.

- Fungsi (Agregasi)

Buatlah tabel ujian, Atlet dan isikan datanya di Database dengan nama = DB_FUNGSI

Isi Tabel Ujian

No_peserta	Nama_peserta	Nilai_ujian
1	Naufal	80
2	Farman	60
3	Esa	55
4	Nadif	65
5	Rama	70
6	Pandu	50
7	Adit	75
8	Arvi	65

Dengan menggunakan tabel Ujian selesaikan soal berikut ini

1. Buatlah perintah untuk menampilkan nama peserta ujian yang nilainya paling tinggi
2. Buatlah perintah untuk menampilkan nama peserta ujian yang nilainya paling rendah
3. Buatlah perintah untuk menghitung banyaknya peserta ujian yang memperoleh nilai rata-rata
4. Buatlah perintah untuk menampilkan 2 peserta ujian yang nilainya rendah

Isi tabel Atlet

Id_atlet	Nama_atlet	Asal daerah	Gender
1	Budi Riandi	DIY	L
2	Susi susanti	DKI Jakarta	P
3	Febrianti	Surabaya	P
4	Perdana Anwar	Sumbar	L
5	Ani Wartiningsih	DIY	P
6	Firmansyah	Surabaya	L
7	Cahyo Hidayat	DIY	L
8	Adi Putra	DIY	L
9	Okta Ashriza	Sumbar	P

Menggunakan tabel Atlet untuk menyelesaikan soal

1. Buatlah perintah untuk menampilkan nama daerah yang mempunyai atlet paling banyak
2. Buatlah perintah untuk menghitung banyaknya daerah yang merupakan asal atlet
3. Menghitung banyaknya peserta per daerah asal , khusus untuk yang banyaknya lebih dari satu
4. Buatlah perintah untuk menghitung banyaknya atlet laki-laki dan perempuan setiap daerah asal
5. Buatlah perintah untuk mencari atlet tersedikit setiap daerah asal, apakah laki-laki atau perempuan
6. Buatlah perintah untuk menampilkan nama daerah beserta banyaknya atlet masing-masing daerah, dimana hasilnya akan ditampilkan menurut urutan jumlah atlet

Pertemuan 7. Penggunaan Power Designer untuk Perancangan Database

1. Instalasi Power Designer
2. Buatlah rancangan disain konseptual (ERD) dari sistem yang akan dijadikan tugas akhir mata kuliah

Pertemuan 8. Penggunaan Power Designer untuk Mengenerate Database

1. Dari hasil ERD (disain konseptual) buatlah fisik database yang anda inginkan.
2. Lengkapi database hasil generate tersebut dengan data riil minimal 10 item per tabel.

Pertemuan 9.

- Fungsi Lanjutan (Grouping)

Tugas

I. Dengan menggunakan tabel mata_kuliah untuk menyelesaikan soal berikut :

Tabel Matakuliah

Kode_mk	Nama_mk	Sks	Kode_prasyarat
DT005	Kalkulus	4	Null
DT006	Logika dan Algoritma	4	Null
DT007	Pemrograman Komputer I	2	Null
DT010	Pengantar Ilmu Komputer	2	Null
DT015	Struktur Data	4	DT007
DT018	Organisasi Komputer	4	DT010
DT019	Sistem Operasi	4	DT010
DT022	Pengolahan Basis Data	4	NULL
DT023	Matematika Diskret	2	DT005
DT028	Jaringan Komputer I	2	DT018
DT030	Pemrograman Komputer II	4	DT015
DT034	Pemrograman Basis Data	4	DT022
DT037	Jaringan Komputer II	4	DT028
DT041	Pemrograman Client	2	DT037
ST011	Manajemen Umum	2	NULL
ST017	Teknik Digital	2	NULL
ST025	Mikroprosesor	2	ST017
ST026	Praktikum Hardware	2	ST017
ST027	Praktikum Elektronik	2	ST017
ST031	Kepemimpinan	2	ST011
ST035	Sistem Informasi mana	4	ST011

1. Buatlah perintah untuk menghitung banyaknya matakuliah per sks per kuliah prasyarat
2. Buatlah perintah untuk menghitung banyaknya matakuliah, masing-masing yang mempunyai kuliah prasyarat dan yang tidak mempunyai kuliah prasyarat
3. Buatlah perintah untuk menghitung banyaknya matakuliah berbobot 4 sks, masing-masing yg tidak mempunyai kuliah prasyarat
4. Buatlah perintah untuk menghitung banyaknya matakuliah yang memuat kata 'pemrograman'
5. Buatlah perintah untuk menghitung banyaknya matakuliah yang memuat kata 'komputer' dan 'basis data' serta mempunyai kuliah prasyarat.
6. Buatlah perintah untuk memperoleh sks terkecil untuk matakuliah yang mempunyai prasyarat
7. Buatlah perintah untuk menampilkan nama matakuliah yang menjadi huruf besar semua

II. Dengan menggunakan **tabel pinjaman** berikut ini, untuk menyelesaikan soal berikut ini :

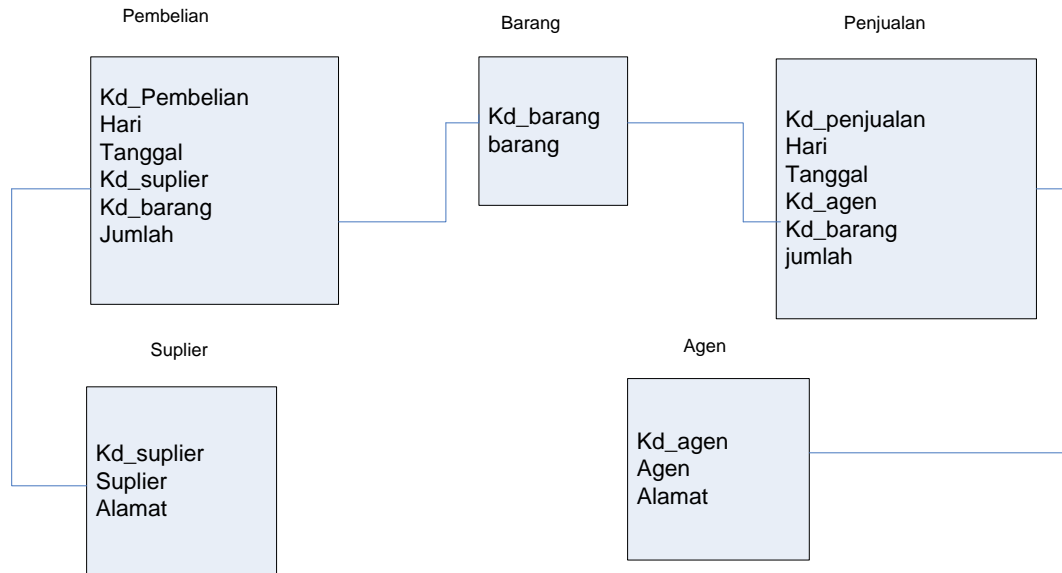
Tgl_pinjaman	Nama	Jml_pinjaman	Bunga (dlm %)
02/14/2005	Naufal Rasendriya	2000.000	4
06/26/2005	Najwa Rashika	2500.000	5
10/23/2005	Ema Utami	1000.000	2
11/19/2005	Suwanto Raharjo	3.500.000	7

1. Buatlah perintah untuk menghitung hutang yang harus dibayarkan untuk masing-masing anggota koperasi dengan ketentuan bunga seperti data di atas.
2. Buatlah perintah untuk menghitung keuntungan yang didapatkan koperasi
3. Jika setiap anggota dikenakan kewajiban membayar hutang tersebut dengan jangka waktu 60 hari, buatlah perintah untuk menghitung kapan hutang-hutang tersebut jatuh tempo

Pertemuan 10. Join

Tugas Praktikum

Berikut ini diberikan database market dealer



Relasi antar Tabel Market dealer

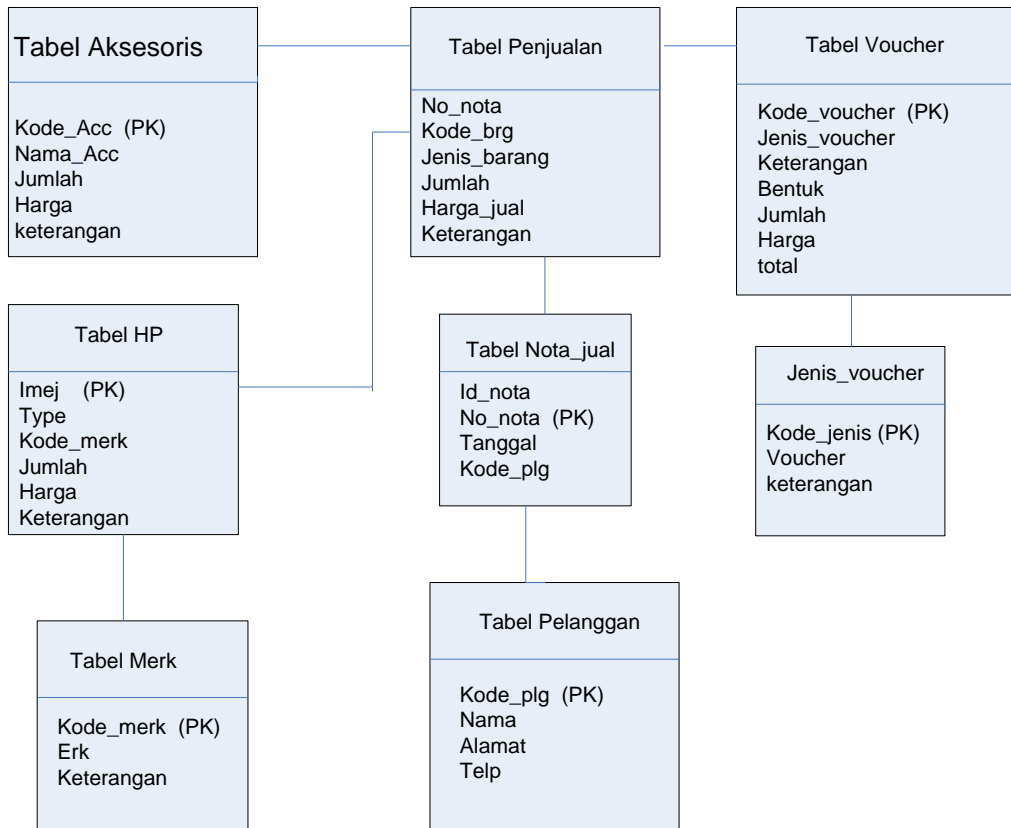
Berdasarkan Relasi Tabel di Atas, Buatlah perintah SQL untuk perintah dibawah ini:

1. Tampilkan data pembelian (kode pembelian, hari, tanggal, nama supplier, nama barang, jumlah)
2. Tampilkan data penjualan (kode penjualan, hari, tanggal, nama agen, nama barang, jumlah)
3. Tampilkan nama barang yang terjual dengan jumlah > 20
4. Tampilkan nama barang dan jumlah barang yang terjual pada tgl 1 september 2008, dikelompokkan berdasarkan nama barang
5. Tampilkan nama dan alamat supplier yang menyeluplai sabun
6. Tampilkan nama dan alamat agen yang membeli sampo
7. Tampilkan nama barang yang terbeli dengan agen yang beralamat Yogyakarta
8. Tampilkan nama supplier yang menyuplai barangnya pada tgl 2 september 2008

Pertemuan 11. View

Latihan VIEW di LAB

Perhatikan database diagram counter untuk pengolahan data counter HP Sell Bandung berikut ini :



Gambar : Database Diagram Counter

Isi Tabel Pelanggan

Kode_plg	Nama	Alamat	Telp
P001	Dwi	Jl, Babarsari	081328341367
P002	Uswan	Godean	081328341368
P003	Sukrisno	Jl Tanjung 11 Concat YK	081328341365
P005	Della	Slokan Mataram	081328341369
P006	Rendra	Jl Magelang Km.7	081328341378
P007	Eko	Mancasan	081328341314
P0024	Endik	Bantul	081328341348

Isi Tabel Merk

Kode_Merk	Merk	Keterangan
MT	MOTOROLLA	
NK	NOKIA	LKP MLS
PL	PHILIPS	
PN	PANASONIC	
SG	SAGEM	
SM	SAMSUNG	
SMN	SIEMENS	
SN	SONY ERICSON	BAGUS

Isi Tabel HP

Imej	Type	Kode_merk	Jumlah	Harga	Keterangan
HP-1	Null	Null	0	5000000	NULL
HP-2	8250	Nk	0	550000	
HP-3	N-Gage	Nk	0	1250000	
HP-4	N-Gage 21	Nk	0	2350000	
HP-5	A	Nk	1	100000	
HP-6	3310	Nk	0	300000	I KP
HP-7	V	Nk	0	2000000	
HP-8	C35	SMN	1	574000	
HP-9	C50	SMN	1	450000	
HP-10	C100	NK	1	560000	
HP-11	9500	NK	0	7000000	

Isi Tabel Aksesories

Kode_acc	Nama_Acc	JUmlah	HArga	Keterangan
12345	Casing Nokia 8250	7	120000	
SIM-IM3	Im3	0	25000	
SIM-MENT	Simcard Mentari	0	12000	0815123456768
SIM-SIMPT	Simcard Simpati	0	12500	

Isi Tabel Jenis_Voucher

Kode_jenis	Voucher	Keterangan
AS	KArtu As	
IM3	IM3	
MT	MENTARI	
SMP	SIMPATI	
XL	PRO XL	

Isi Tabel Voucher

Kode_voucher	Jenis_voucher	Keterangan	Bentuk	Jumlah	Harga	Total
IM3E20	IM3	IM3 Elektronik 20000	Elektronik	0	20000	0
MTE20	MT	Mintari Elektrik 20000	Elektronik	0	20000	0
SME100	SMP	Simpati elektrik 100000	Elektronik	0	100000	0
SME20	SMP	Simpati elektrik 20000	Elektronik	0	20000	0
SME50	SMP	Simpati elektrik 50000	Elektronik	0	50000	0
SMF20	SMP	Simpati Fisik 20000	fisik	7	20000	140000
SMF50	SMP	Simpati Fisik 50000	fisik	10	50000	500000
XJE20	XL	XI Jempol Elektronik 20000	fisik	0	20000	0

Isi Tabel Nota_jual

Id_nota	No_nota	Tanggal	Kode_plg
182	000001	2008-07-19	Null
184	000002	2008-07-23	P003
185	000003	2008-07-23	P003
186	000004	2008-07-23	NULL
187	000005	2008-07-23	NULL
188	000006	2008-07-27	NULL

189	000007	2008-07-28	P003
-----	--------	------------	------

Isi Tabel_penjualan

No_nota	Kode_brg	Jenis_barang	Jumlah	Harga_jual	Keterangan
000001	1234	AC	1	12000	NULL
000001	SIM-MENT	AC	1	13000	NULL
000002	HP-11	HP	1	7650000	
000003	HP-4	HP	1	550000	
000004	IM3E20	VC	1	21500	08564300909
000005	12345	AC	2	12500	NULL
000005	SIM-IM3	AC	1	15000	NULL
000006	12345	AC	1	20000	NULL
000007	HP-4	HP	1	450000	

Dengan menggunakan relasi table di atas, buatlah view-view berikut ini :

1. Menampilkan data HP yang telah terjual, kolom yang ditampilkan adalah imej, type, merk dan harga jual
2. Menampilkan data HP yang masih tersedia yang merknya adalah 'NOKIA', kolom yang ditampilkan adalah imej, type, merk, jumlah dan harga
3. Menampilkan jumlah barang yang terjual dan total harga penjualan pada tanggal 19 July 2008
4. Menampilkan voucher yang paling banyak terjual , kolom yang ditampilkan adalah voucher dan total dari harga jual
5. Menampilkan stok voucher yang masih tersisa paling banyak , kolom yang ditampilkan adalah voucher dan jumlah total
6. Menampilkan harga aksesories yang paling mahal, kolom yang ditampilkan adalah semua kolom dalam tabel aksesories

Pertemuan 12. Trigger

Buatlah tabel-tabel berikut:

tabel_barang

id_brg	Nama_brg	stok
A10	Mouse	10
A11	Keyboard	15
A12	DVD-RW	19

tabel_pembelian

id_pem	id_brg	jml_beli
1	A10	5

Ketikkan trigger berikut:

```
DELIMITER //
CREATE TRIGGER inkremenStok
BEFORE INSERT ON barang
FOR EACH ROW BEGIN
SET NEW.stok = NEW.stok + 1;
END //
```

Untuk pengecekan apakah trigger tersebut sudah berjalan sebagaimana mestinya, tambahkan 1 contoh transaksi ke tabel_pembelian seperti berikut ini

```
INSERT INTO barang
VALUES ('A13', 'Modem', 5);
```

Periksa hasilnya

```
mysql> SELECT * FROM barang;
+-----+-----+-----+
| kodebarang | nama_barang | stok |
+-----+-----+-----+
| A10        | Mouse       | 10   |
| A11        | Keyboard    | 15   |
| A12        | DVD-RW     | 10   |
| A13        | Modem       | 6    |
+-----+-----+-----+
4 rows in set (0.00 sec)
```

Untuk mendapatkan informasi mengenai daftar trigger yang telah terdefinisi, gunakan perintah SHOW TRIGGERS (tambahkan \G untuk menampilkan data secara vertikal)

Sebagaimana objek-objek database lainnya, kita menghapus trigger dengan menggunakan perintah DROP

```
mysql> DROP TRIGGER inkremenStok;
Query OK, 0 rows affected (0.00 sec)
```

Untuk memastikan bahwa operasi penghapusan sudah berhasil, periksa kembali eksistensi trigger.

```
mysql> SHOW TRIGGERS\G
Empty set (0.02 sec)
```


Keyword OLD dan NEW

Buatlah trigger seperti berikut

```
DELIMITER //
CREATE TRIGGER updateStok
AFTER INSERT ON pembelian

FOR EACH ROW BEGIN

    UPDATE barang
    SET stok = stok + NEW.jml_beli
    WHERE id_brg = NEW.id_brg;

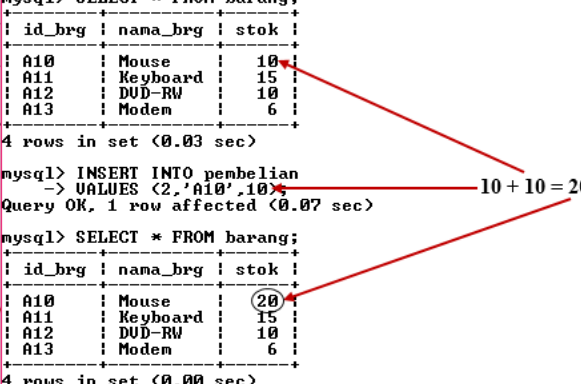
END//
```

Pada contoh diatas, penambahan data pembelian akan mengakibatkan nilai stok barang berubah menyesuaikan banyaknya nilai jumlah pembelian.

```
mysql> SELECT * FROM barang;
+----+-----+-----+
| id_brg | nama_brg | stok |
+----+-----+-----+
| A10    | Mouse    | 10   |
| A11    | Keyboard | 15   |
| A12    | DVD-RW   | 10   |
| A13    | Modem    | 6    |
+----+-----+-----+
4 rows in set (0.03 sec)

mysql> INSERT INTO pembelian
-> VALUES (2,'A10',10);
Query OK, 1 row affected (0.07 sec)

mysql> SELECT * FROM barang;
+----+-----+-----+
| id_brg | nama_brg | stok |
+----+-----+-----+
| A10    | Mouse    | 20   |
| A11    | Keyboard | 15   |
| A12    | DVD-RW   | 10   |
| A13    | Modem    | 6    |
+----+-----+-----+
4 rows in set (0.00 sec)
```



Untuk kasus trigger DELETE, keyword yang bisa digunakan hanya OLD. Misalkan kita ingin mendefinisikan trigger untuk menghapus semua data pembelian manakala data barang yang sesuai diindikasikan melalui *primary key* dan *foreign key* dihapus.

```
DELIMITER //

CREATE TRIGGER deleteChild
AFTER DELETE ON barang

FOR EACH ROW BEGIN

DELETE FROM pembelian
WHERE id_brg = OLD.id_brg;

END//
```

untuk operasi UPDATE, kita bisa memanfaatkan keyword NEW maupun OLD.

```
DELIMITER //

CREATE TRIGGER updateStokEdit
AFTER UPDATE ON pembelian

FOR EACH ROW BEGIN
UPDATE barang
```

```
SET stok = stok + (NEW.jml_beli - OLD.jml_beli)
WHERE id_brg = NEW.id_brg;
```

```
END//
```

```
mysql> UPDATE pembelian
-> SET jml_beli = 20
-> WHERE id_pem = 3;
```

```
mysql> SELECT * FROM pembelian;
+----+-----+-----+
| id_pem | id_brg | jml_beli |
+----+-----+-----+
| 3 | A11 | 20 |
+----+-----+-----+
1 row in set (0.00 sec)

mysql> SELECT * FROM barang;
+----+-----+-----+
| id_brg | nama_brg | stok |
+----+-----+-----+
| A11 | Keyboard | 35 |
| A12 | DUD-RW | 10 |
| A13 | Modem | 6 |
+----+-----+-----+
3 rows in set (0.00 sec)
```

Latihan

Buatlah trigger untuk memeriksa operasi penambahan data **barang**. Skenarionya, jika data sudah ada, berikan status barang sudah ada; sebaliknya, data bisa langsung dimasukkan.

TUGAS PRAKTIKUM

1. Modifikasi trigger INSERT pembelian untuk menambahkan fitur pemberian bonus. Skenarionya, jika pembelian lebih dari 150 dan kurang dari 250, bonus 15; pembelian lebih dari 250 dan kurang dari 350, bonus 25; pembelian lebih dari 350, bonus 50. Ingat, untuk aturan penyesuaian stok barang juga masih berlaku !
2. Buat tabel **pembayaran** dengan *field* **id_pem** dan **jumlah**, kemudian tambahkan *field* **harga** di tabel barang. Modifikasi trigger INSERT pembelian untuk menambahkan fitur penghitungan nota pembayaran. Misalkan harga satuan barang A adalah Rp. 2000, dan terjadi pembelian 15 barang, maka jumlah pembayaran tercatat Rp. 30000. Ingat, aturan penyesuaian stok barang dan juga pemberian bonus masih berlaku di sini.
3. Buat tabel **log_pembelian** dengan *field* **waktu** dan **operasi**. Selanjutnya, definisikan trigger di tabel pembelian untuk merekam operasi INSERT, UPDATE, dan DELETE, dan kemudian menyimpannya sebagai bukti transaksi di tabel **log_pembelian**.

Note :

Untuk mencegah duplikasi pendefinisian trigger, terlebih dahulu hapus semua trigger di tabel pembelian

Pertemuan 13. Store Procedure

Format umum untuk membuat *Stored Procedure* adalah sebagai berikut :

```
CREATE PROCEDURE procedure1 /* name */
(IN parameter1 INTEGER) /* parameters */
BEGIN /* start of block *002F
DECLARE variable1 CHAR(10); /* variables */
IF parameter1 = 17 THEN /* start of IF */
SET variable1 = 'birds'; /* assignment */
ELSE
SET variable1 = 'beasts'; /* assignment */
END IF; /* end of IF */
INSERT INTO table1 VALUES (variable1);/* statement */
END /* end of block */
```

MySQL mendukung *Stored Procedure* untuk versi 5.0 atau setelahnya. Untuk memeriksa apakah versi database MySQL yang digunakan telah mendukung *Stored Procedure* dapat dilakukan dengan cara memberikan perintah berikut :

```
mysql> SHOW VARIABLES LIKE 'version';
```

atau

```
mysql> SELECT VERSION();
```

A. Memulai Stored Procedure

Sebuah procedure dapat dibuat dengan perintah sebagaimana contoh berikut :

```
mysql> CREATE PROCEDURE p1 () SELECT * FROM t_sdana;//
Query OK, 0 rows affected (0.47 sec)
```

Dari perintah diatas tampak bahwa struktur pembentuk *Stored Procedure* di MySQL terdiri dari :

- SQL *statement* untuk membuat *procedure* : CREATE PROCEDURE;
- Nama *procedure* : p1;
- *Parameter list*, ditandai dengan (), diperlukan untuk *procedure* yang membutuhkan parameter;
- SQL *statement*, yang merupakan isi *procedure*, pada contoh diatas adalah perintah "SELECT * FROM t_sdana;"

SQL *statement* yang dapat digunakan pada bagian isi *procedure* meliputi :

- Standard SQL, seperti : INSERT, UPDATE, DELETE, SELECT, DROP, CREATE, REPLACE, SET, COMMIT, ROLLBACK.
- MySQL *Extra feature*, berupa *statement Data Definition Language (DDL)*, seperti : DROP table.
- MySQL *Extension*, berupa *direct select*, Contoh : SELECT 'A'.

Adapun SQL *statement* yang tidak dapat digunakan pada bagian isi procedure adalah CREATE PROCEDURE, ALTER PROCEDURE, DROP PROCEDURE, CREATE FUNCTION, CREATE TRIGGER, CREATE EVENT, USE database, LOAD DATA INFILE, LOCK TABLES, CHECK.

Untuk memanggil atau menjalankan *procedure* dilakukan dengan memberikan perintah CALL nama_procedure diikuti dengan *delimiter*.

Menjalankan dan hasil *procedure* p1 adalah sebagai berikut :

```
mysql> CALL p1 //
+-----+-----+-----+
| kdsdana | nmsdana | nmsdana2 |
+-----+-----+-----+
| 01      | RM      | RUPIAH MURNI |
| 02      | PLN     | PINJAMAN LUAR NEGERI |
| 03      | RMP     | RUPIAH MURNI PENDAMPING |
| 04      | PNP     | PNBP |
| 05      | PDM     | PINJAMAN DALAM NEGERI |
| 06      | BLU     | BADAN LAYANAN UMUM |
| 07      | STM     | STIMULUS |
| 08      | HDN     | HIBAH DALAM NEGERI |
| 09      | HLN     | HIBAH LUAR NEGERI |
+-----+-----+-----+
9 rows in set (0.00 sec)
Query OK, 0 rows affected (0.05 sec)
```

perintah diatas adalah untuk memanggil *procedure* p1 yang berisi perintah : `select * from t_sdana.`

Pertemuan 14. Review

Buat sebuah database yang bernama "mycomp" dengan skema sebagai berikut:

1. Buat sql script yang membangun tabel berikut :
 - ✓ Karyawan (nopeg, nama, gaji, pendapatan)
 - ✓ Nasabah (nonasabah, namanasabah)
 - ✓ Pesanan (nopesanan, nopeg, nonasabah, tglpesanan)
 - ✓ DtlPesanan (nopesanan, nourut, nobarang, jumlah)
 - ✓ Barang (nobarang, deskripsi, qty, harga)
2. Buat sql script yang memasukkan minimal 3 karyawan, 2 nasabah dan 5 jenis barang
3. Buat view pesanan vpesanan yang menampilkan DtlPesanan dan Barang dengan format sebagai berikut : *nopesanan, nourut, deskripsi, jumlah, harga, total*
4. Buat prosedur yang menampilkan view no 3 diatas, berdasarkan nopesanan sebagai parameter.
5. Buat trigger pada DtlPesanan, setiap kali nobarang ditambahkan, dihapus atau diubah, maka perubahan juga mempengaruhi qty pada Barang
6. Buat beberapa transaksi Pesanan dan DtlPesanan
7. Buat sebuah view yang menampilkan pendapatan setiap salesman yang berada pada tabel Karyawan. Perhitungan berdasarkan DtlPesanan, jumlah barang dan harga barang.