**Nama : Rima Natalia**

**Kelas : XII RPL2**

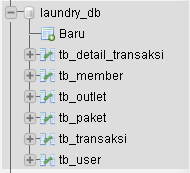
**Materi : Database Laundry**

**1. Buatlah basis data dengan nama laundry\_db !**

****

CREATE DATABASE laundry\_db;

**2. Buatlah semua tabel yang dibutuhkan sesuai rancangan !**

****

CREATE TABLE tb\_member(

id int PRIMARY KEY,

nama VARCHAR (100),

alamat text,

jenis\_kelamin ENUM ('L', 'P'),

tlp VARCHAR (15)

);

CREATE TABLE tb\_outlet(

id int PRIMARY KEY,

nama VARCHAR (100),

alamat text,

tlp VARCHAR (15)

);

CREATE TABLE tb\_paket(

id int PRIMARY KEY,

id\_outlet int,

jenis ENUM ('kiloan', 'selimut', 'bed\_cover','kaos', 'lain' ),

nama\_paket VARCHAR (100),

harga int

);

CREATE TABLE tb\_transaksi(

id int PRIMARY KEY,

id\_outlet int,

kode\_invoice VARCHAR (100),

id\_member int,

tgl datetime,

batas\_waktu datetime,

tgl\_bayar datetime,

biaya\_tambahan int,

diskon double,

pajak int,

status ENUM ('baru', 'proses', 'selesai', 'diambil'),

dibayar ENUM ('dibayar', 'belum\_dibayar'),

id\_user int

);

CREATE TABLE tb\_detail\_transaksi(

id int PRIMARY KEY,

id\_transaksi int,

id\_paket int,

qty double,

keterangan text

);

CREATE TABLE tb\_user(

id int PRIMARY KEY,

nama VARCHAR (100),

username VARCHAR (30),

password text,

id\_outlet int,

role ENUM ('admin', 'kasir', 'owner')

);

**3. Lakukan penambahan data pada semua tabel master minimal 3 data tiap tabel !**

****

INSERT INTO tb\_member

(id, nama, alamat, jenis\_kelamin, tlp)

VALUES

(1, "Jamal", "Jl. Universe", "L", "087355263"),

(2, "Junita", "Jl. Senayan", "P","0863442672"),

(3, "Juniar", "Jl. Pangandaran", "L", "0862543622");

INSERT INTO tb\_outlet

(id, nama, alamat, tlp)

VALUES

(1, "HOTSA", "Cugenang","6247774893"),

(2, "RIDDIN", "Sukajaya","629482764"),

(3, "CHEW", "Majalengka", "629380924");

INSERT INTO tb\_paket

(id, id\_outlet, jenis, nama\_paket, harga)

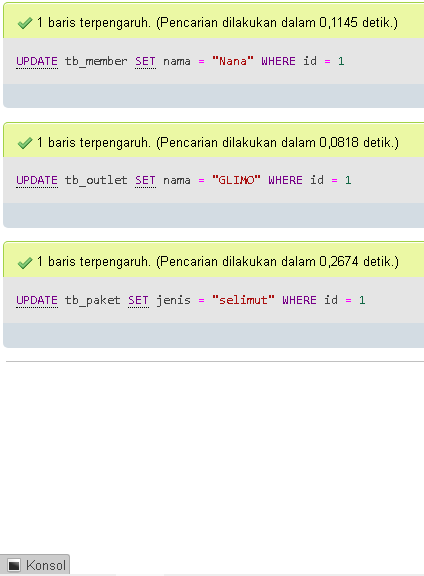
VALUES

(1, "1", "kaos", “paket murah”, “25000”),

(2, "2", "bed\_cover", “paket hemat”, “30000”),

(3, "3", "selimut", “paket royal”, “40000”);

**4. Lakukan perubahan data pada semua tabel master minimal 1 data tiap tabel !**

****

UPDATE tb\_member SET nama = "Nana"

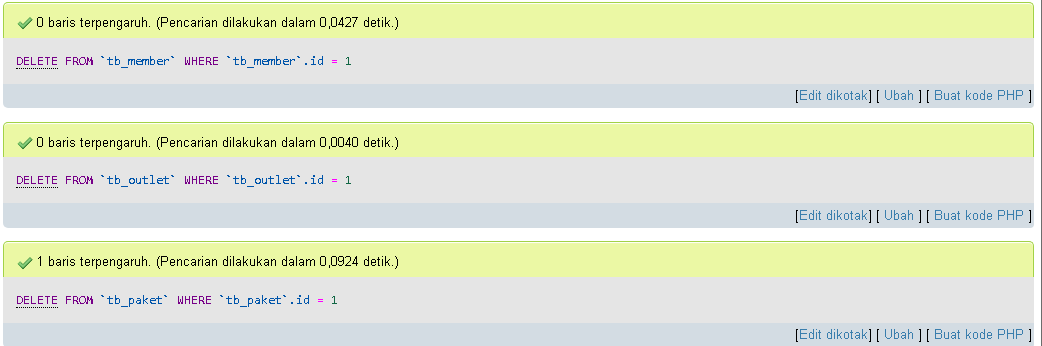
WHERE id = 1;

UPDATE tb\_outlet SET nama = "GLIMO"

WHERE id = 1;

UPDATE tb\_paket SET jenis = "selimut"

WHERE id = 1;

**5. Lakukan penghapusan data pada semua tabel master minimal 1 data tiap tabel !**  ****

DELETE FROM `tb\_member` WHERE `tb\_member`.id = 1;

DELETE FROM `tb\_outlet` WHERE `tb\_outlet`.id = 1;

DELETE FROM `tb\_paket` WHERE `tb\_paket`.id = 1;

**6. Buatlah instruksi SQL untuk menampilkan id, nama\_paket, Jenis, harga pada table paket yang harganya dibawah 50000 !**

SELECT \* FROM `tb\_paket` WHERE harga <= 50000;

****

**7. Buatlah instruksi SQL untuk menampilkan id, tgl, status (tabel Transaksi), nama, alamat (table member), nama\_paket, jenis, harga (table Paket) dimana data transaksi tersebut memiliki status “DIAMBIL”!**

SELECT t.id, t.tgl, t.status, m.nama, m.alamat, p.nama\_paket, p.jenis, p.harga FROM tb\_transaksi t

INNER JOIN tb\_member m ON t.id\_member=m.id

INNER JOIN tb\_detail\_transaksi dt ON dt.id\_transaksi = t.id

INNER JOIN tb\_paket p ON dt.id\_paket = p.id

WHERE t.status = "diambil";

**8. Buatlah Stored Procedure untuk menampilkan id, tgl, status (tabel Transaksi), nama, alamat (table member), nama\_paket, jenis, harga (table Paket) dimana data transaksi tersebut memiliki status “PROSES”!**

SELECT t.id, t.tgl, t.status, m.nama, m.alamat, p.nama\_paket, p.jenis, p.harga FROM tb\_transaksi t

INNER JOIN tb\_member m ON t.id\_member=m.id

INNER JOIN tb\_detail\_transaksi dt ON dt.id\_transaksi = t.id

INNER JOIN tb\_paket p ON dt.id\_paket = p.id

WHERE t.status = "proses";

**9. Buatlah function untuk menghasilkan total transaksi dengan parameter adalah id\_transaksi, dimana alur prosesna adalah dengan menjumlahkan harga per paket di kali dengan jumlah pakaian yang di ambil dari detail transaksi!**

CREATE FUNCTION total\_transaksi (i int) RETURNS int

BEGIN

DECLARE total int

SELECT SUM (p.harga\*dt.qty) INTO total FROM tb\_transaksi t

INNER JOIN tb\_detail\_transaksi dt ON t.id=dt.id\_transaksi

INNER JOIN tb\_paket p ON dt.id\_paket=p.id

WHERE t.id=i;

RETURN total;

END//

**10. Buatlah table logging dengan field aksi typedata varchar(20), waktu typedata datetime **

CREATE table logging (

id int PRIMARY KEY,

aksi varchar (20),

time datetime

);

**11. Buatlah 3 trigger untuk menginput pada table logging apabila ada kejadian insert, update dan delete pada table paket**

****

CREATE TRIGGER insert\_paket AFTER INSERT ON tb\_paket FOR EACH ROW

BEGIN

INSERT INTO logging (`aksi`, `time`) VALUES ("INSERT PAKET", CURRENT\_TIMESTAMP);

END//

CREATE TRIGGER update\_paket AFTER UPDATE ON tb\_paket FOR EACH ROW

BEGIN

INSERT INTO logging (`aksi`, `time`) VALUES ("UPDATE PAKET", CURRENT\_TIMESTAMP);

END//

CREATE TRIGGER delete\_paket AFTER DELETE ON tb\_paket FOR EACH ROW

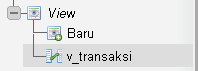
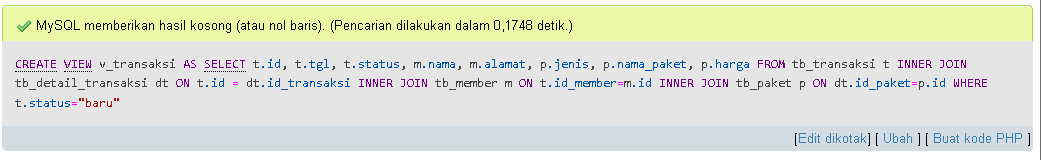
BEGIN

INSERT INTO logging (`aksi`, `time`) VALUES ("DELETE PAKET", CURRENT\_TIMESTAMP);

END//

****

**12. Buatlah view dengan nama v\_transaksi untuk menampilkan id, tgl, status (tabel Transaksi), nama, alamat (table member), nama\_paket, jenis, harga (table Paket) dimana data transaksi tersebut memiliki status “BARU”!**

****

CREATE VIEW v\_transaksi

AS

SELECT t.id, t.tgl, t.status, m.nama, m.alamat, p.jenis, p.nama\_paket, p.harga

FROM tb\_transaksi t INNER JOIN tb\_detail\_transaksi dt ON t.id = dt.id\_transaksi INNER JOIN tb\_member m ON t.id\_member=m.id INNER JOIN tb\_paket p ON dt.id\_paket=p.id WHERE t.status="baru";

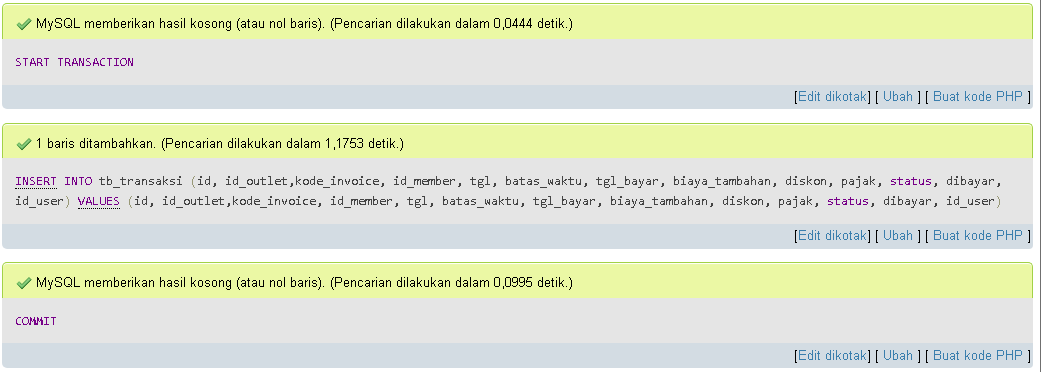
**13. Buatlah perintah commit untuk aksi insert data pada tabel transaksi**

START TRANSACTION;

INSERT INTO tb\_transaksi (id, id\_outlet,kode\_invoice, id\_member, tgl, batas\_waktu, tgl\_bayar, biaya\_tambahan, diskon, pajak, status, dibayar, id\_user)

VALUES (id, id\_outlet,kode\_invoice, id\_member, tgl, batas\_waktu, tgl\_bayar, biaya\_tambahan, diskon, pajak, status, dibayar, id\_user);

COMMIT;

****

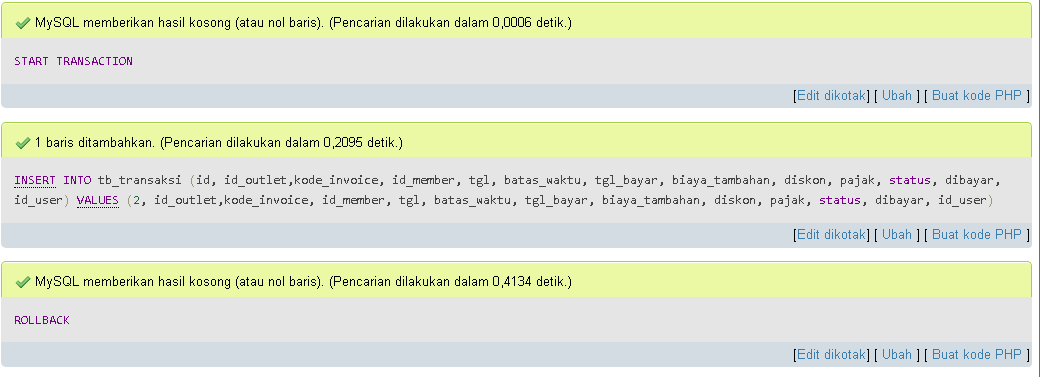
**14. Buatlah perintah rollback untuk aksi insert pada tabel transaksi**

START TRANSACTION;

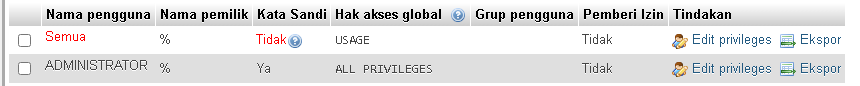
INSERT INTO tb\_transaksi (id, id\_outlet,kode\_invoice, id\_member, tgl, batas\_waktu, tgl\_bayar, biaya\_tambahan, diskon, pajak, status, dibayar, id\_user)

VALUES (2, id\_outlet,kode\_invoice, id\_member, tgl, batas\_waktu, tgl\_bayar, biaya\_tambahan, diskon, pajak, status, dibayar, id\_user);

ROLLBACK;

****

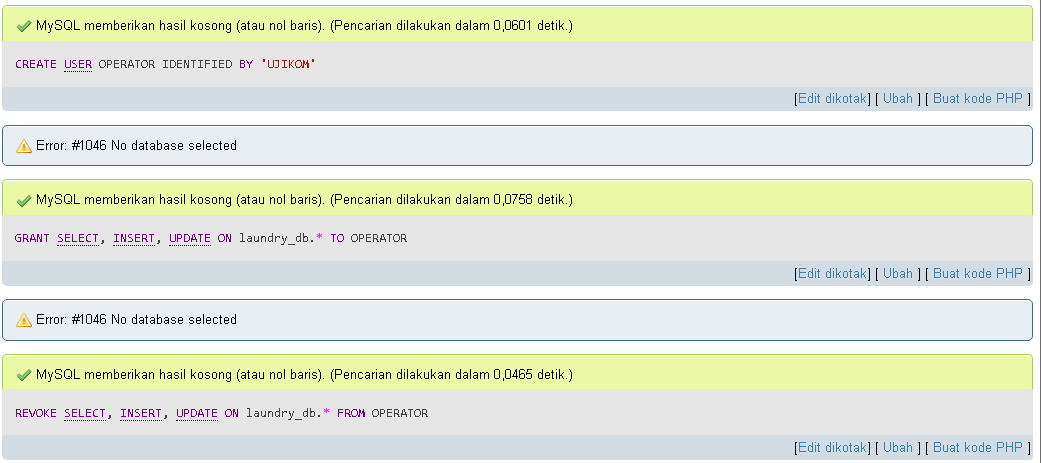
**15. Buatlah user baru dengan nama ADMINISTRATOR dan password UJIKOM, berikan hakakses agar user dapat menggunakan fungsi select,insert,update,delete pada semua table di database laundry\_db**

****

CREATE USER ADMINISTRATOR IDENTIFIED BY 'UJIKOM';

GRANT ALL ON \*.\* TO ADMINISTRATOR;

**16. Buatlah user baru dengan nama OPERATOR dan password UJIKOM, cabut hakakses agar user tidak dapat menggunakan fungsi select,insert,update,delete pada table outlet dan paket di database laundry\_db**

****

CREATE USER OPERATOR IDENTIFIED BY 'UJIKOM';

GRANT SELECT, INSERT, UPDATE ON laundry\_db.\* TO OPERATOR;

REVOKE SELECT, INSERT, UPDATE ON laundry\_db.\* FROM OPERATOR;

**17. Lakukan backup dan restore basis laundry\_db menggunakan dbms**

****

