**Nama : Rima Natalia**

**Kelas : XII RPL 2**

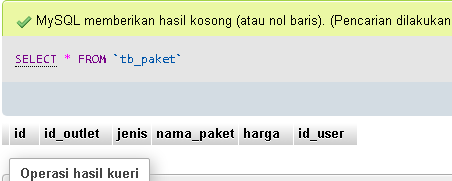
**Materi : Database Laundry TO-2**

**1. Buatlah basis data dengan dengan format db\_laundry\_nis contoh db\_laundry\_192010008 !**

****

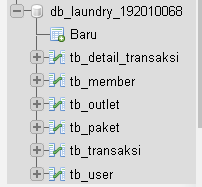
CREATE DATABASE db\_laundry\_192010068;

**2. Tambahkan field id\_user (integer) di tabel paket!**

****

ALTER TABLE tb\_paket ADD id\_user int;

**3. Buatlah tabel di database baru menggunakan query manual dengan nama barang dengan field sebagai berikut: id (int), nama\_barang (varchar), qty(int), harga(double), waktu\_pembelian(date\_time), supplier(varchar), created\_at (datetime), updated\_at (datetime)**

****

CREATE TABLE tb\_member(

id int UNSIGNED PRIMARY KEY AUTO\_INCREMENT NOT NULL,

nama VARCHAR (100),

alamat text,

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

tlp VARCHAR (15)

);

CREATE TABLE tb\_outlet(

id int UNSIGNED PRIMARY KEY AUTO\_INCREMENT NOT NULL,

nama VARCHAR (100),

alamat text,

tlp VARCHAR (15)

);

CREATE TABLE tb\_paket(

id int UNSIGNED PRIMARY KEY AUTO\_INCREMENT NOT NULL,

id\_outlet int,

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

nama\_paket VARCHAR (100),

harga int

);

CREATE TABLE tb\_transaksi(

id int UNSIGNED PRIMARY KEY AUTO\_INCREMENT NOT NULL,

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 UNSIGNED PRIMARY KEY AUTO\_INCREMENT NOT NULL,

id\_transaksi int,

id\_paket int,

qty double,

keterangan text

);

CREATE TABLE tb\_user(

id int UNSIGNED PRIMARY KEY AUTO\_INCREMENT NOT NULL,

nama VARCHAR (100),

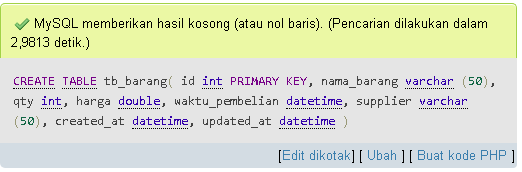
username VARCHAR (30),

password text,

id\_outlet int,

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

);



CREATE TABLE tb\_barang(

id int UNSIGNED PRIMARY KEY AUTO\_INCREMENT NOT NULL,

nama\_barang varchar (50),

qty int,

harga double,

waktu\_pembelian datetime,

supplier varchar (50),

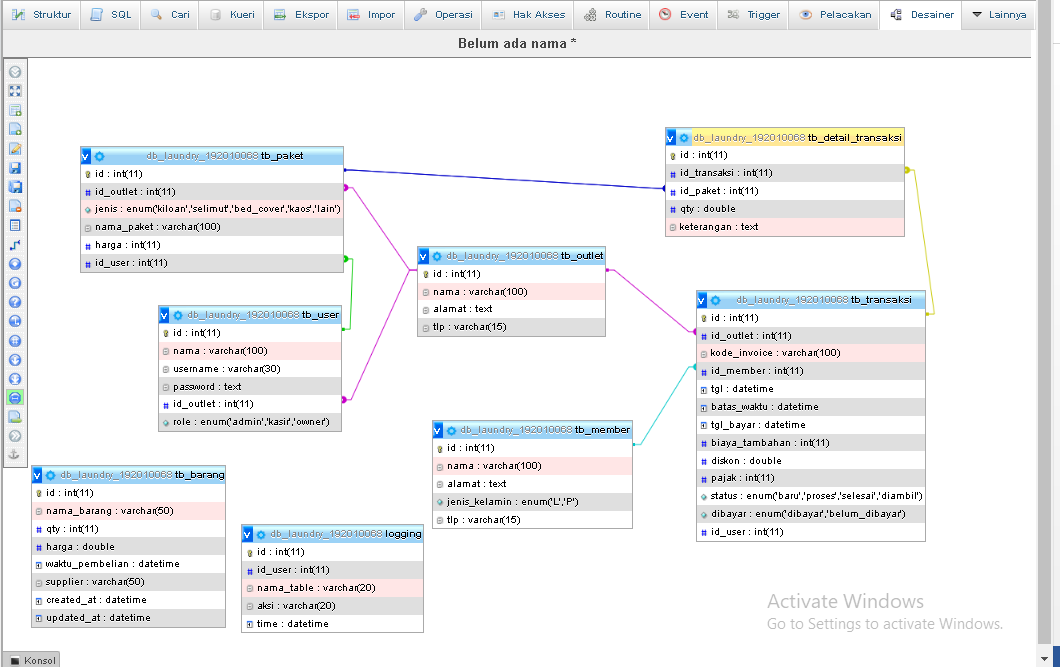
created\_at datetime,

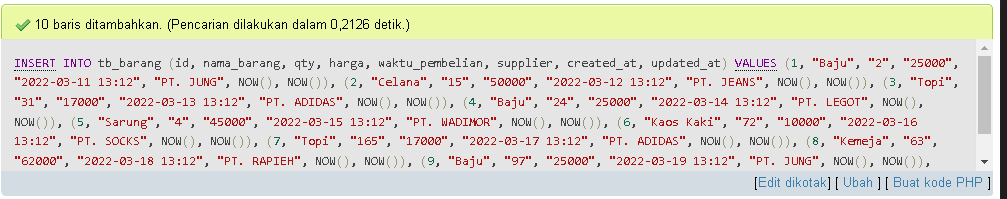
updated\_at datetime

);

**4. Buatlah foreign key (indexing) antar tabel supaya bisa berelasi!**

ALTER TABLE tb\_paket ADD CONSTRAINT FOREIGN KEY (id\_user) REFERENCES tb\_user(id);

****

**5. Lakukan penambahan data pada tabel barang minimal 10 data dengan data yang waktu\_pembelian berisi data dari 11 Maret 2022 – 20 Maret 2022 dengan rentang qty bermacam-macam 5 barang 1-50, 5 barang 60 - 200!** ****

INSERT INTO tb\_barang

(nama\_barang, qty, harga, waktu\_pembelian, supplier, created\_at, updated\_at)

VALUES

( "Baju", "2", "25000", "2022-03-11 13:12", "PT. JUNG", NOW(), NOW()),

( "Topi", "165", "17000", "2022-03-17 13:12", "PT. ADIDAS", NOW(), NOW()),

( "Kemeja", "63", "62000", "2022-03-18 13:12", "PT. RAPIEH", NOW(), NOW()),

( "Baju", "97", "25000", "2022-03-19 13:12", "PT. JUNG", NOW(), NOW()),

( "Celana", "15", "50000", "2022-03-12 13:12", "PT. JEANS", NOW(), NOW()),

( "Topi", "31", "17000", "2022-03-13 13:12", "PT. ADIDAS", NOW(), NOW()),

( "Baju", "24", "25000", "2022-03-14 13:12", "PT. LEGOT", NOW(), NOW()),

( "Sarung", "4", "45000", "2022-03-15 13:12", "PT. WADIMOR", NOW(), NOW()),

( "Kaos Kaki", "72", "10000", "2022-03-16 13:12", "PT. SOCKS", NOW(), NOW()),

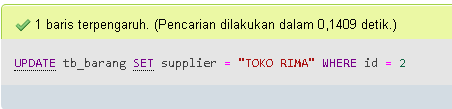
( "Beanie", "110", "35000", "2022-03-20 13:12", "PT. JAMAL", NOW(), NOW());

**6. Buatlah instruksi SQL untuk menampilkan data barang yang qty minimal 50 dan waktu\_pembelian dari 15- 20 Maret 2022**

****

SELECT \* FROM tb\_barang WHERE (waktu\_pembelian BETWEEN '2022-03-11 13:12:00' AND '2022-03-15 13:12:00') AND qty >= 50;

**7. Buatlah satu instruksi SQL untuk mengupdate supplier dengan toko nama kalian, misal “TOKO AZFA” yang waktu\_pembelian antara tgl 11 Maret 2022 – 15 Maret 2022.**

****

UPDATE tb\_barang SET supplier = "TOKO RIMA" WHERE id = 2;

**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 “diambil” dan jenis (paket) “bed\_cover”!**

//

CREATE PROCEDURE getDataStatusTransaksi(p\_status VARCHAR(100), p\_jenis VARCHAR(100))

BEGIN

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 t.id = dt.id\_transaksi

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

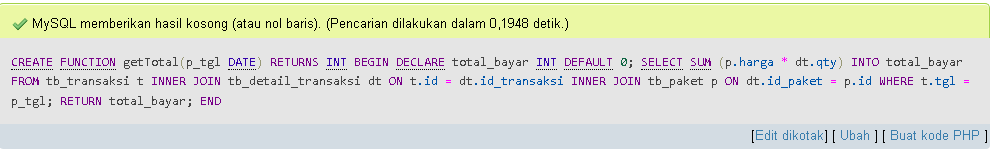
WHERE t.status = p\_status

AND p.jenis = p\_jenis;

END//

CALL getDataStatusTransaksi("diambil", "bed\_cover");

**9. Buatlah function untuk menghasilkan TOTAL\_UANG per tanggal dengan status field dibayar : “dibayar” (tb\_transaksi) dengan harga diambil dari tb\_detail\_transaksi dan tb\_paket dengan parameter tgl (tb\_transaksi)**

****

//

CREATE FUNCTION getTotal(p\_tgl DATE) RETURNS INT

BEGIN

DECLARE total\_bayar INT DEFAULT 0;

SELECT SUM (p.harga \* dt.qty) INTO total\_bayar 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

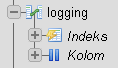
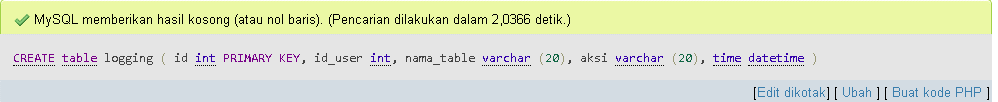
WHERE t.tgl = p\_tgl;

RETURN total\_bayar;

END//

SELECT getTotal("2022-02-27");

**10. Buatlah tabel logging dengan field aksi(varchar), nama\_tabel, waktu (datetime), id\_user(int) Dan Buatlah 3 trigger untuk menginput pada table logging apabila ada kejadian insert, update dan delete pada table barang dengan isi dari field aksi sesuai dengan aksi query!**

**** 

CREATE table logging (

id int PRIMARY KEY,

id\_user int,

nama\_table varchar (20),

aksi varchar (20),

time datetime

);

****

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

BEGIN

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

END//

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

BEGIN

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

END//

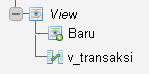
CREATE TRIGGER delete\_barang AFTER DELETE ON tb\_barang FOR EACH ROW

BEGIN

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

END//

**11. Buatlah view dengan nama v\_transaksi untuk menampilkan id, tgl, status (tabel Transaksi), nama, alamat (table member), total\_bayar (dihitung dengan membuat function total bayar per transaksi) dimana data transaksi tersebut memiliki status “proses”!**

****

CREATE VIEW v\_transaksi AS

SELECT t.id, t.tgl, m.nama, m.alamat, SUM(p.harga\*dt.qty)

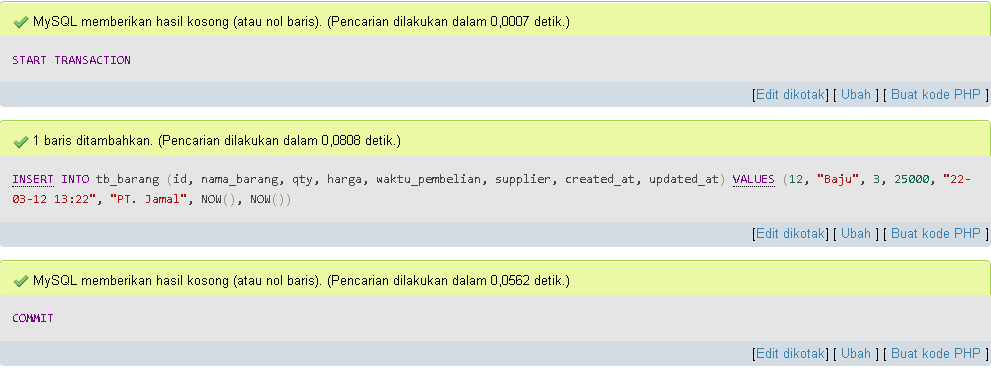
AS total\_bayar FROM tb\_transaksi t

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

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.status = "proses" GROUP BY t.id;

**12. Buatlah perintah commit untuk aksi insert data pada tabel barang **

START TRANSACTION;

INSERT INTO tb\_barang (nama\_barang, qty, harga, waktu\_pembelian, supplier, created\_at, updated\_at)

VALUES ( "Baju", 3, 25000, "22-03-12 13:22", "PT. Jamal", NOW(), NOW());

COMMIT;

**13. Buatlah perintah rollback untuk aksi insert pada tabel barang**

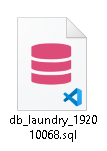
START TRANSACTION;

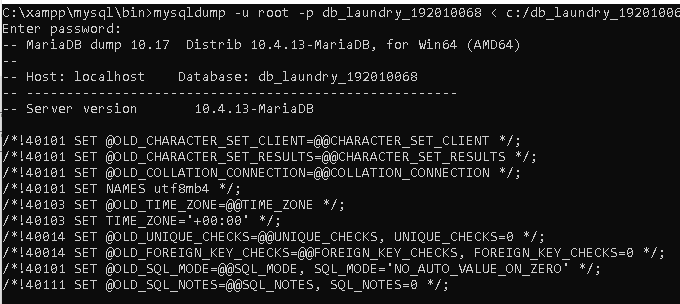
INSERT INTO tb\_barang (nama\_barang, qty, harga, waktu\_pembelian, supplier, created\_at, updated\_at)

VALUES ( "Celana", 2, 48000, "22-03-13 13:22", "PT. JEANS", NOW(), NOW());

ROLLBACK;

**14. Lakukan backup dan restore pada database ini dengan menggunakan dbms**

****

****