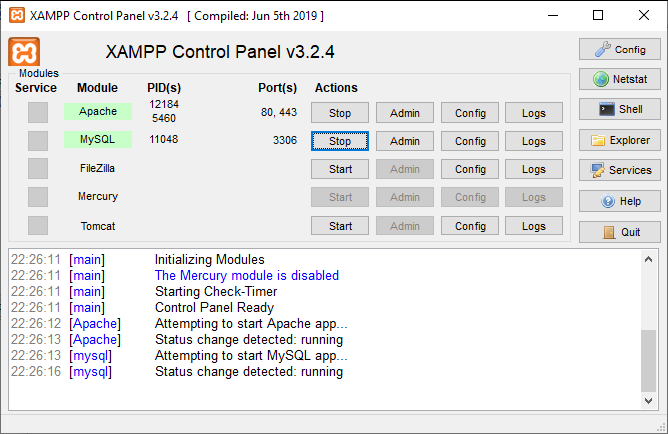
**Form Penjualan (Pertemuan ke 11)**

* Jalankan XAMPP terlebih dahulu

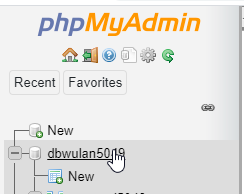


* Buka browser dan jalankan phpmyadmin

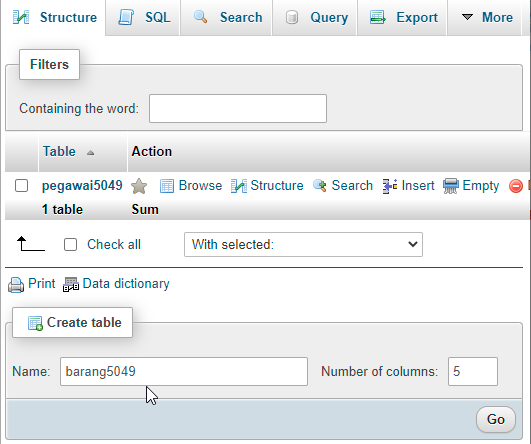


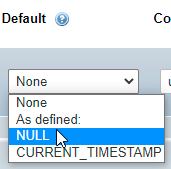
* Buat file tabel pegawai :

Aktifkan terlebih dahulu database yang akan digunakan, dengan mengklik nama databasenya :



Pada tab stucture, isikan nama tabel yg mau dibuat pada Name Create table (jual5049), dan Go



Buat Tabel : jualXXXX (sesuai nim, misal : jual5049)

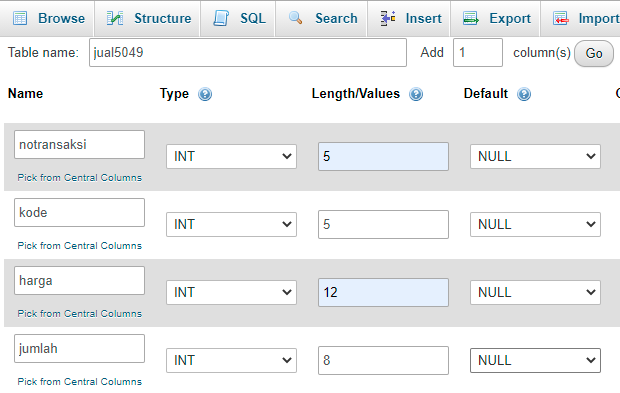
**Name Type Length**

notransaksi int 5

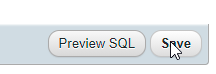
kode varchar 5

harga int 12

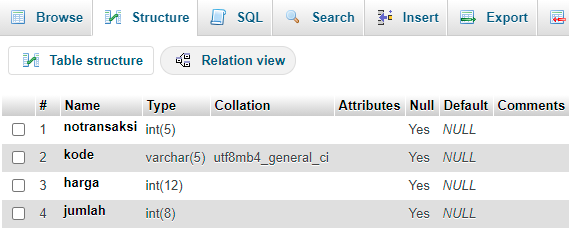
jumlah int 8



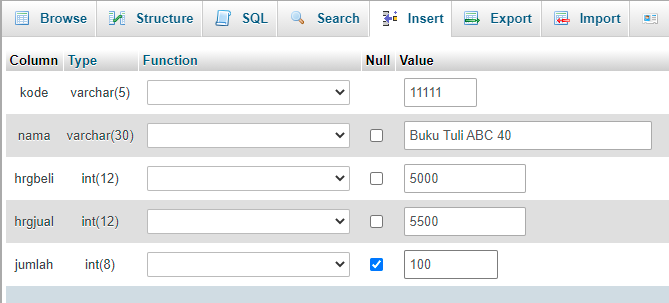
Selanjutnya pilih save



Struktur tabel yang terbentuk sebagai berikut :



Lakukan pengisian data sebagai berikut :



Tambahkan data setidaknya 5 record (setiap data sudah diisikan, klik Go)

Output program :

**Contoh program sebagai berikut :**

Ganti dengan **NIM dan Nama**



Database (tabel jual5049 setelah adanya penjualan)



Kode program :

import java.awt.BorderLayout;

import java.awt.Dimension;

import java.awt.Font;

import java.awt.Color;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import java.awt.event.\*;

import javax.swing.\*;

import java.sql.\*;

import javax.swing.table.DefaultTableModel;

public class FJual06284 extends JFrame implements ActionListener, FocusListener

{

private JPanel pBrg = new JPanel();

private JLabel lNoJual = new JLabel ("No. Transaksi "),

lKode = new JLabel ("Kode "),

lNama = new JLabel ("Nama"),

lHarga = new JLabel ("Harga "),

lJumlah = new JLabel ("Jumlah "),

lBayar = new JLabel ("Bayar "),

lKembali = new JLabel ("Kembali "),

lTotal = new JLabel ("Total "),

lToko1 = new JLabel ("Dewi Putri Ayuningsih"),

lToko2 = new JLabel ("A12.2019.06284");

private JTextField fNoJual = new JTextField (),

fKode = new JTextField (),

fNama = new JTextField (),

fHarga = new JTextField (),

fJumlah = new JTextField (),

fTotal = new JTextField (),

fBayar = new JTextField (),

fKembali = new JTextField ();

private JButton btnJual = new JButton (),

btnDelete = new JButton (),

btnCari = new JButton (),

btnSelesai = new JButton ();

private ResultSet rs;

String[] header = {"Kode","Nama","Harga", "Jumlah", "Total"};

DefaultTableModel tabMode01;

JTable tabel = new JTable();

JScrollPane skrTabel = new JScrollPane();

float total = 0,bayar = 0, kembali = 0;

int xjumlah = 0;

public FJual06284() {

setPreferredSize(new Dimension(580,600));

setTitle("Transaksi Penjualan");

setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

new BorderLayout();

JDesktopPane pBrg = new JDesktopPane();

add(pBrg, BorderLayout.CENTER);

/\* Mengatur letak objek Label Di container\*/

lToko1.setBounds (15, 28, 450, 40);

lToko1.setFont( new Font( "Dialog" , 1 , 36 ));

lToko2.setBounds (15, 58, 450, 40);

lToko2.setFont( new Font( "Dialog" , 1 , 18 ));

lToko1.setForeground(Color.BLUE);

lNoJual.setBounds (15, 100, 120, 20);

lKode.setBounds (15, 400, 60, 20);

lHarga.setBounds (15, 425, 60, 20);

lJumlah.setBounds (15, 450, 60, 20);

lBayar.setBounds (350, 425, 60, 20);

lKembali.setBounds (350, 450, 60, 20);

/\* Mengatur letak objek Text Di Container \*/

fNoJual.setBounds (15, 125, 100, 20);

fKode.setBounds (75, 400, 100, 20);

fNama.setBounds (180, 400, 205, 20);

fTotal.setBounds (400, 400, 150, 20);

fHarga.setBounds (75, 425, 92, 20);

fBayar.setBounds (400, 425, 150, 20);

fKembali.setBounds (400, 450, 150, 20);

fJumlah.setBounds (75, 450, 100, 20);

/\* Mengatur letak objek Button di Container \*/

btnJual.setBounds (165, 500, 85, 25);

btnJual.setLabel("Jual");

btnDelete.setBounds (265, 500, 85, 25);

btnDelete.setLabel("Delete");

btnCari.setBounds (365, 500, 85, 25);

btnCari.setLabel("Cari");

btnSelesai.setBounds (465, 500, 85, 25);

btnSelesai.setLabel("Selesai");

tabMode01 = new DefaultTableModel(null,header);

tabel.setModel(tabMode01);

skrTabel.getViewport().add(tabel);

tabel.setEnabled(true);

skrTabel.setBounds(15,160,540,220);

/\* Mengatur Perataan teks di Text \*/

fHarga.setHorizontalAlignment (JTextField.RIGHT);

fJumlah.setHorizontalAlignment (JTextField.RIGHT);

fBayar.setHorizontalAlignment (JTextField.RIGHT);

fKembali.setHorizontalAlignment (JTextField.RIGHT);

fTotal.setHorizontalAlignment (JTextField.RIGHT);

/\* Objek Button di Non Aktifkan dan di aktifkan \*/

btnJual.setEnabled(true);

btnDelete.setEnabled(false);

btnSelesai.setEnabled(true);

/\* Mengatur objek untuk dapat berinteraksi \*/

fKode.addFocusListener (this);

fJumlah.addFocusListener (this);

fKembali.addFocusListener (this);

btnJual.addActionListener (this);

btnDelete.addActionListener (this);

btnCari.addActionListener (this);

btnSelesai.addActionListener (this);

//tabel.TampungMouseListener(this);

// Meletakkan seluruh kontrol pada objek panel \*/

pBrg.add (skrTabel);

pBrg.add(lToko1);

pBrg.add(lToko2);

pBrg.add(lNoJual);

pBrg.add(fNoJual);

pBrg.add(lKode);

pBrg.add(fKode);

pBrg.add(lNama);

pBrg.add(fNama);

pBrg.add(lHarga);

pBrg.add(fHarga);

pBrg.add(lJumlah);

pBrg.add(fJumlah);

pBrg.add(lTotal);

pBrg.add(fTotal);

pBrg.add(lBayar);

pBrg.add(fBayar);

pBrg.add(lKembali);

pBrg.add(fKembali);

pBrg.add(btnJual);

pBrg.add(btnDelete);

pBrg.add(btnCari);

pBrg.add(btnSelesai);

/\* Menambahkan objek panel (pBrg) ke container frame \*/

getContentPane().add (pBrg);

/\* Menampilkan frame ke layar monitor \*/

pack();

setLocationRelativeTo(null);

CariNoJual();

//tampiltabel();

setVisible (true);

}

public static void main(String[] args) {

new FJual06284().setVisible(true);

}

/\* Fungsi untuk memeriksa kursor saat meninggalkan objek txtKode\_Barang \*/

public void focusGained (FocusEvent fe)

{

if (fBayar.getText().equals (""))

{

}

else

Bayar();

}

public void focusLost (FocusEvent fe)

{

if (fKode.getText().equals (""))

{

}

else

{

Cari();

fJumlah.requestFocus ();

}

if ((fJumlah.getText().equals (""))) { }

else

{

int stat = CekJumlah();

if (stat == 1)

{

fJumlah.requestFocus();

}

else

{

tampiltabel();

Kosong();

}

}

}

/\* Fungsi jika user melakukan action penekanan tombol Button \*/

public void actionPerformed (ActionEvent ae) {

Object obj = ae.getSource();

if (obj == btnJual)

{

SimpanJual();

bersihtabel();

total = 0;

fTotal.setText("");

Kosong();

CariNoJual();

}

if (obj == btnDelete)

{

bersihtabel();

Kosong();

}

if (obj == btnCari)

{

Cari();

}

if (obj == btnSelesai)

{

this.dispose();

}

}

// Fungsi untuk mencari Kode\_Barang ke tabel barang

void CariNoJual(){

try {

KoneksiDBMS CnPenjualan = new KoneksiDBMS();

Connection con = CnPenjualan.BukaCn();

Statement stat = con.createStatement();

String sql1 = "SELECT \* FROM jual06284";

ResultSet rs1 = stat.executeQuery(sql1);

if(rs1.next())

{

rs1.last();

// Jika Kode Barang Ditemukan Di Tabel Barang

int xno = Integer.parseInt(rs1.getString ("notransaksi"));

fNoJual.setText (String.valueOf(xno + 1));

btnJual.setEnabled(false);

fKode.requestFocus ();

}

else

{

// Jika Kode Barang Tidak ketemu di Tabel Barang ..

fNoJual.setText ("21001");

fKode.setText("");

fKode.requestFocus();

}

rs1.close();

con.close();

}

catch(SQLException e)

{

}

}

// Fungsi untuk mencari data Barang di tabel barang ........

void Cari(){

try {

KoneksiDBMS CnPenjualan = new KoneksiDBMS();

Connection con = CnPenjualan.BukaCn();

Statement stat = con.createStatement();

String sql1 = "SELECT \* FROM barang06284 WHERE kode ='" + fKode.getText () + "'";

ResultSet rs1 = stat.executeQuery(sql1);

if(rs1.next())

{

// Jika Kode Barang Ditemukan Di Tabel Barang

fKode.setText (rs1.getString ("kode"));

fNama.setText (rs1.getString ("nama"));

fHarga.setText (rs1.getString ("hrgjual"));

xjumlah = Integer.parseInt(rs1.getString ("jumlah"));

btnJual.setEnabled(false);

btnDelete.setEnabled(true);

fNama.requestFocus ();

}

else

{

// Jika Kode Barang Tidak ketemu di Tabel Barang ..

fKode.setText("");

JOptionPane.showMessageDialog(this, "Kode barang tidak ditemukan !!");

fKode.requestFocus();

}

rs1.close();

con.close();

}

catch(SQLException e)

{

}

}

// Fungsi untuk mencari Kode\_Barang ke tabel barang

int CekJumlah(){

int x = 0;

try {

if (xjumlah < Integer.parseInt(fJumlah.getText()))

{

fJumlah.setText("");

JOptionPane.showMessageDialog(this, "Jumlah Barang : " + xjumlah);

x = 1;

}

}

catch(Exception e)

{

}

return x;

}

//Simpan Data...........................

void SimpanJual()

{

try

{

KoneksiDBMS CnPenjualan = new KoneksiDBMS();

Connection con = CnPenjualan.BukaCn();

Statement stat = con.createStatement();

int brs = tabMode01.getRowCount();

for(int i=0;i<brs;i++)

{

int vnojual = Integer.parseInt(fNoJual.getText());

String vkode = String.valueOf(tabel.getValueAt(i,0));

String vharga = String.valueOf(tabel.getValueAt(i,2));

String vjumlah = String.valueOf(tabel.getValueAt(i,3));

String strsql = "insert into jual06284(notransaksi, kode, harga, jumlah) " +

" values ('"+ vnojual + "', '"+ vkode + "', '"+ vharga + "', '"+ vjumlah + "')";

int stsproses = stat.executeUpdate(strsql);

//update data barang................................

String strsqlupdate = "update barang06284 set jumlah = jumlah - '"+ vjumlah + "' " +

" where kode = '"+ vkode + "'";

int stsprosesupdate = stat.executeUpdate(strsqlupdate);

}

con.close();

}

catch(SQLException e)

{

JOptionPane.showMessageDialog(this, "Ada Kesalahan !!!");

}

}

void Bayar()

{

try {

fBayar.requestFocus();

kembali = Float.parseFloat(fBayar.getText()) -

Float.parseFloat(fTotal.getText());

fKembali.setText (String.valueOf(kembali));

} catch (Exception se) {

System.err.println("Pesan Salah : " + se.getMessage());

}

}

void tampiltabel()

{

try {

String Kode = fKode.getText();

String Nama = fNama.getText();

String Harga = fHarga.getText();

String Jumlah = fJumlah.getText();

String Total = String.valueOf(Float.parseFloat(fHarga.getText()) \*

Float.parseFloat(fJumlah.getText()));

String[] data = {Kode, Nama, Harga, Jumlah, Total};

tabMode01.addRow(data);

//masukkan ke total........

total = total + Float.parseFloat(Total);

fTotal.setText (String.valueOf(total));

} catch (Exception se) {

System.err.println("Pesan Salah : " + se.getMessage());

}

}

void bersihtabel()

{

int brs = tabMode01.getRowCount();

for(int i=0;i<brs;i++)

{

tabMode01.removeRow(0);

}

}

// Fungsi untuk mengkosongkan Objek masukan

void Kosong ()

{

fKode.setText ("");

fNama.setText ("");

fHarga.setText ("");

fJumlah.setText ("");

fBayar.setText ("");

fKembali.setText ("");

btnJual.setEnabled(true);

btnDelete.setEnabled(false);

fKode.requestFocus ();

}

}