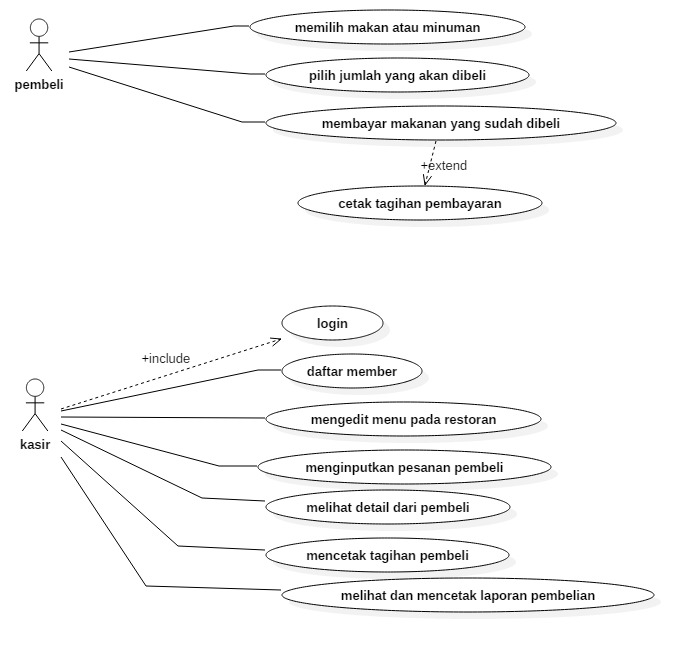
Nama : muhammad deny zulyan aly maksum

Nim : 18050623006

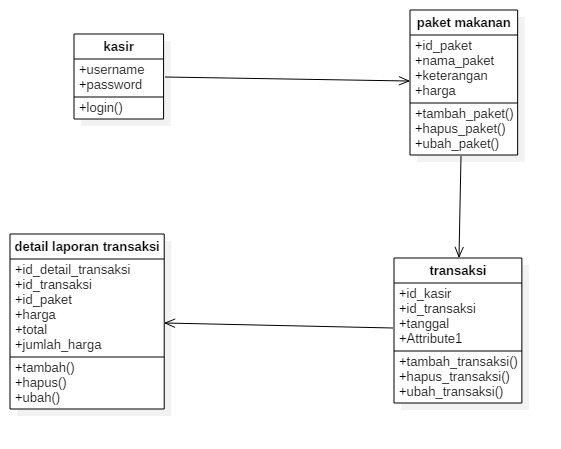
Prodi : D3 manajemen informatika

Tugas membuat kasir restoran menggunakan bahasa pemrogaman java

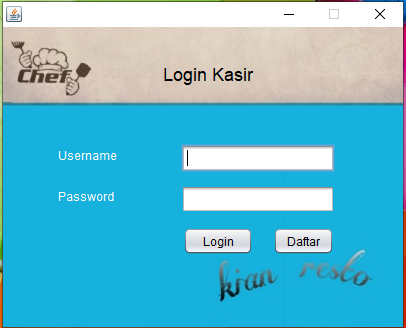
1. Use case dari program kasir



1. Class diagram dari program kasir



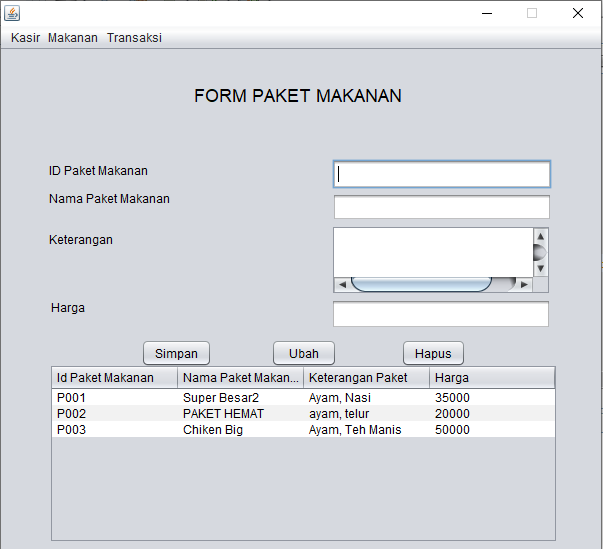
1. Hasil dari program kasir
2. Gambar login, login digunakan oleh admin untuk masuk kedalam programnya.



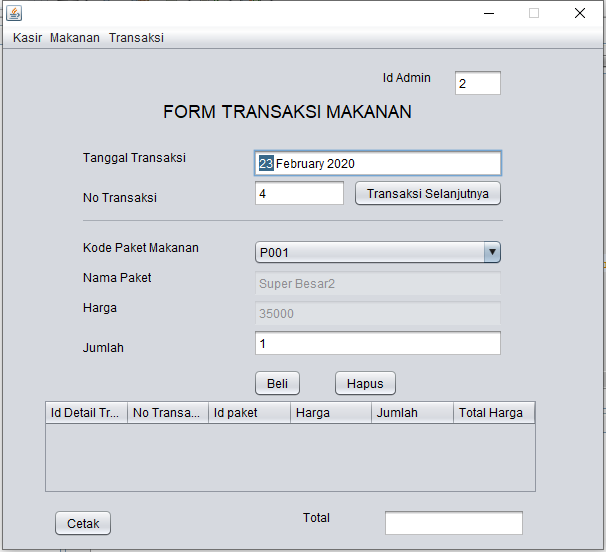
1. Gambar daftar kasir, dimana digunakan untuk menambahkan admin kasir agar dapat melakukan login



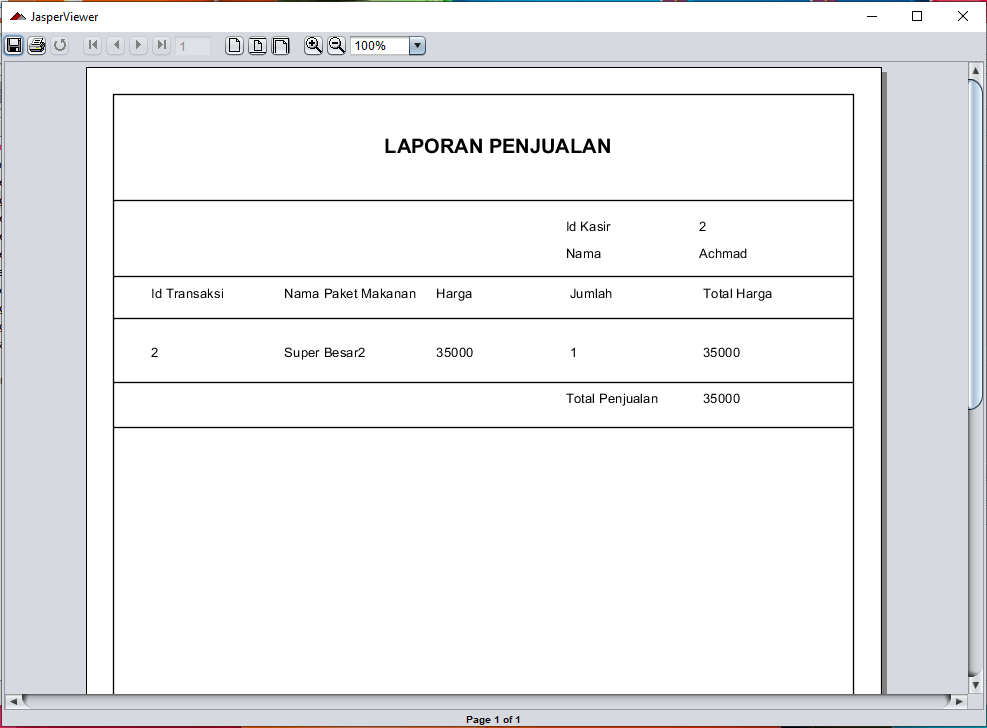
1. Gambar tambah makanan atau minuman, untuk menambahkan daftar paket pada menu restoran



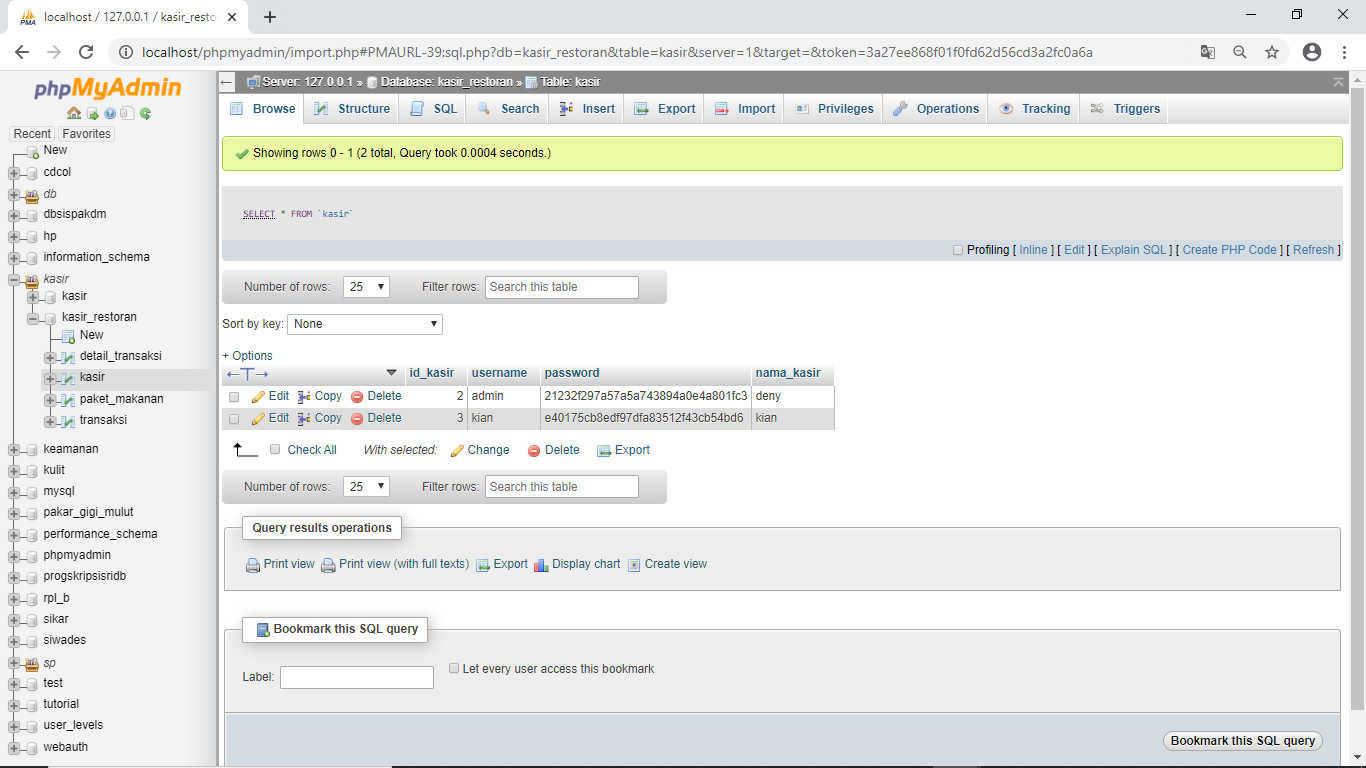
1. Tampilan pemesanan pembeli, admin kasir menginputkan pesanan pembeli dan harga yang harus dibayar oleh pembeli



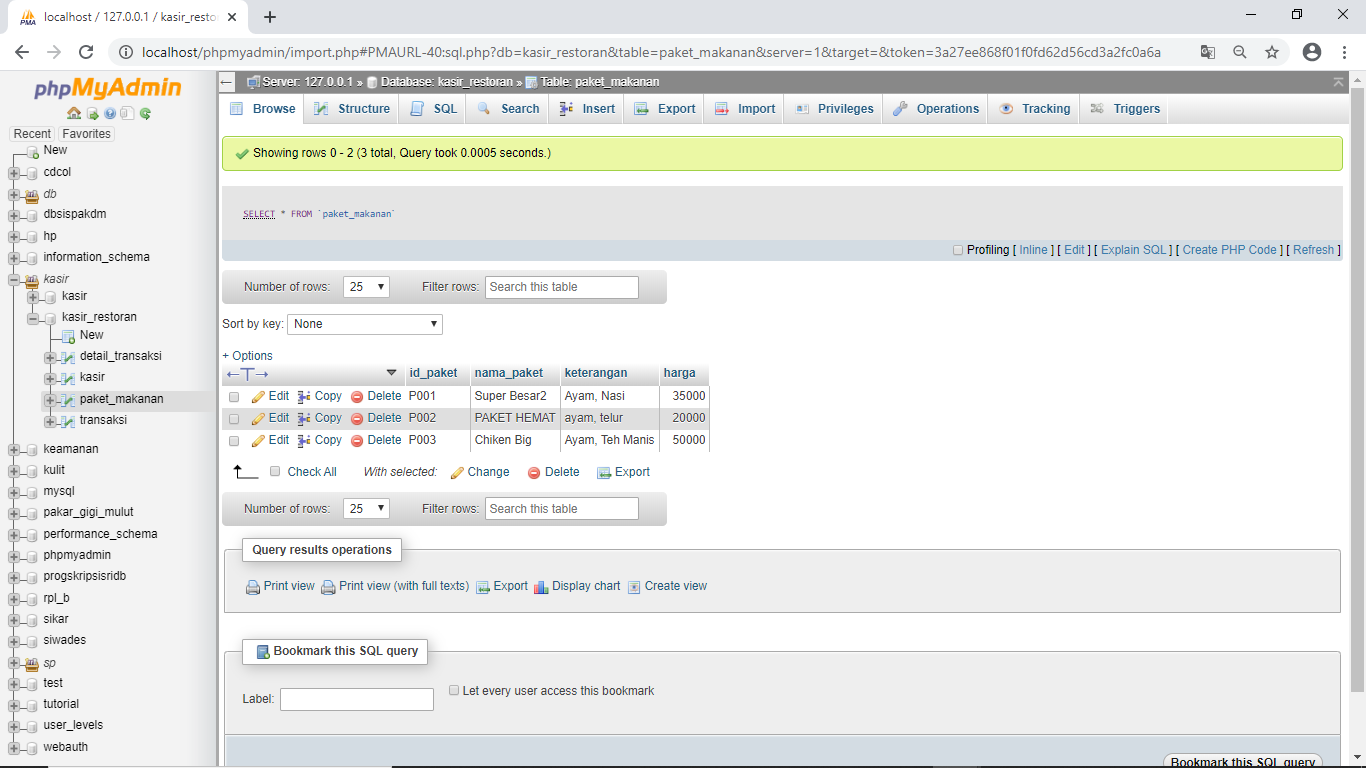
1. Detail atau laporan pemesanan, admin dapat melihat detail dari transaksi pembeli dan data juga akan langsung masuk ke dalam database



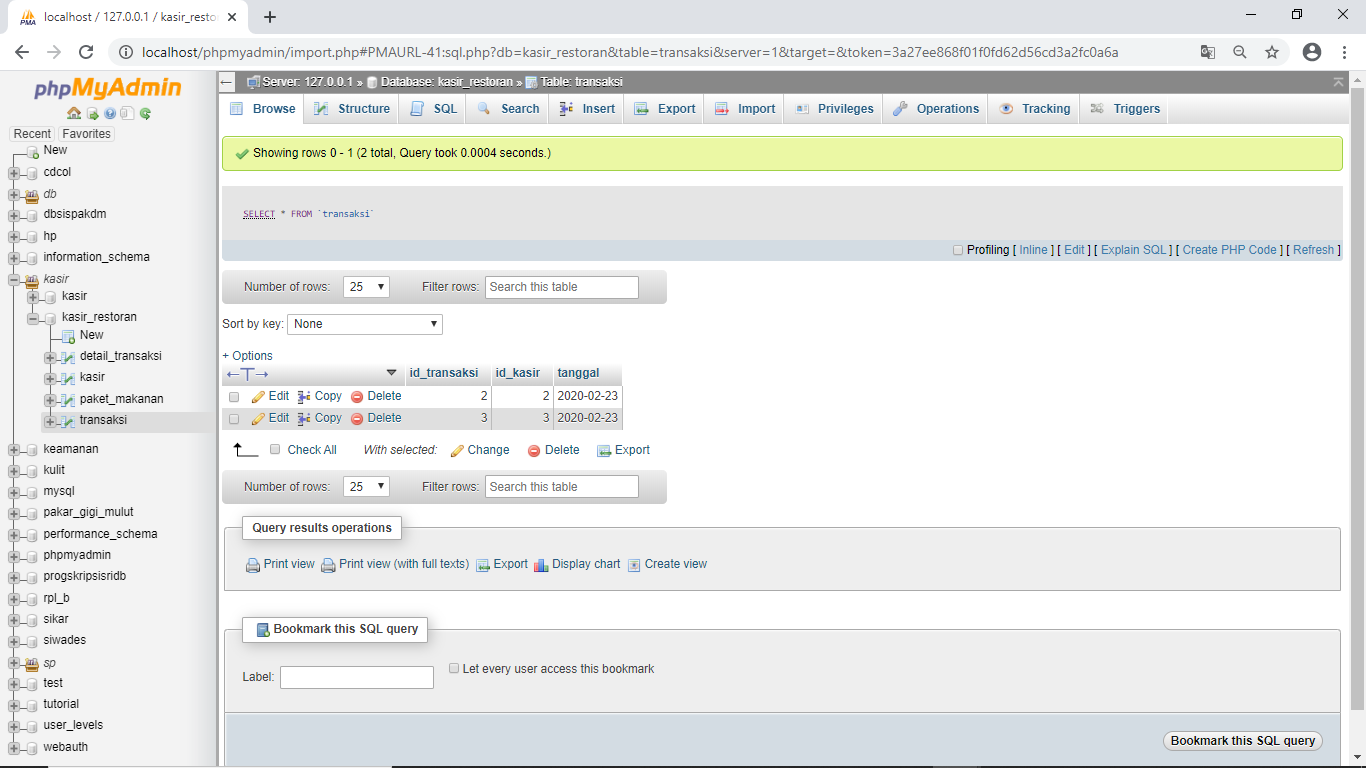
1. Database yang digunakan dalam program kasir
2. Database kasir atau admin



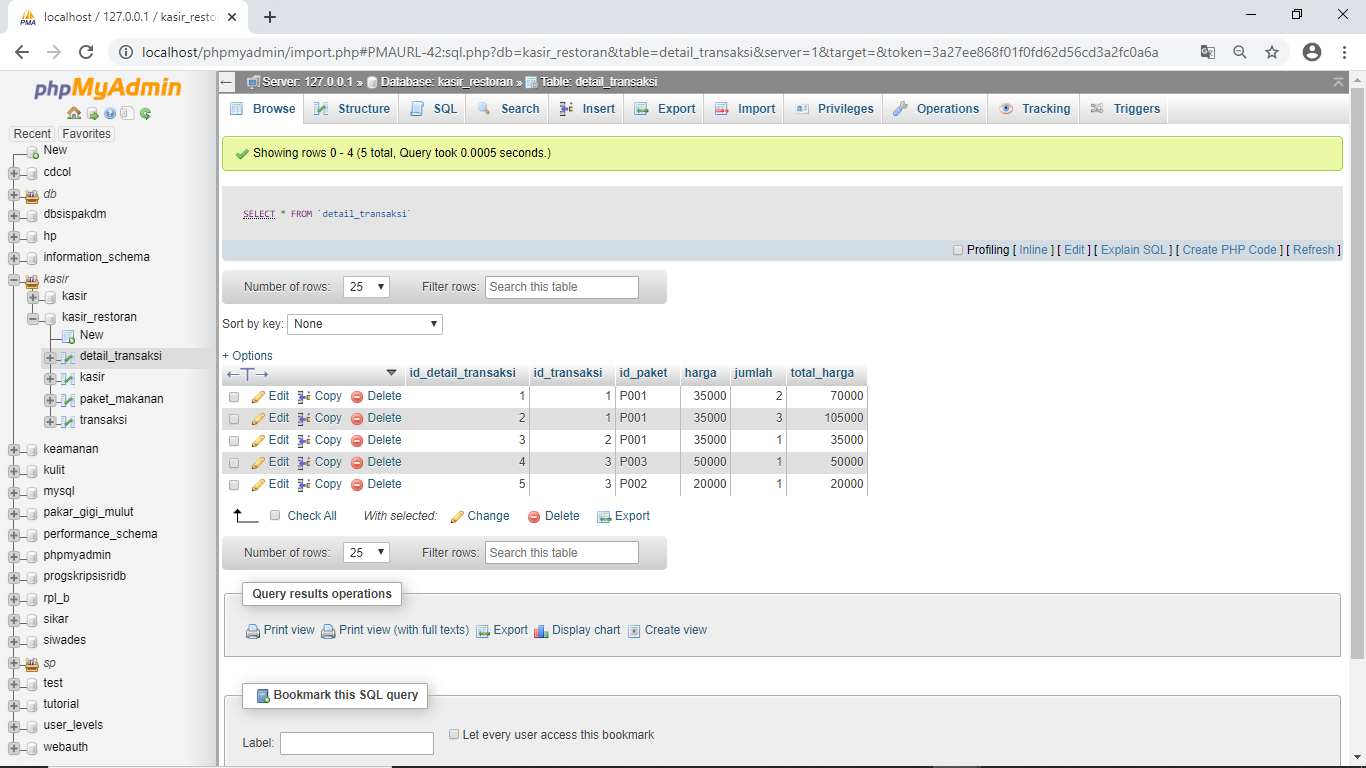
1. Database paket\_makanan



1. Database transaksi



1. Database detail transaksi



1. Sourcode

public void simpan\_no(){

Connection con = koneksi.getConnection();

String sql = "INSERT INTO transaksi VALUES('"+txt\_no\_transaksi.getText()+"','"+txt\_id\_kasir.getText()+"','"+tgl+"')";

try {

stt = con.createStatement();

stt.executeUpdate(sql);

} catch (Exception e) {

}

} private void tabel\_makanan(){

model = new DefaultTableModel();

tbl\_makanan.setModel(model);

model.addColumn("Id Paket Makanan");

model.addColumn("Nama Paket Makanan");

model.addColumn("Keterangan Paket");

model.addColumn("Harga");

}

private void tampil\_tabel(){

Connection con = koneksi.getConnection();

try {

stt = con.createStatement();

String sql = "SELECT \* FROM paket\_makanan";

rss = stt.executeQuery(sql);

while(rss.next()){

Object[] o;

o = new Object[4];

o[0] = rss.getString("id\_paket");

o[1] = rss.getString("nama\_paket");

o[2] = rss.getString("keterangan");

o[3] = rss.getString("harga");

model.addRow(o);

}

} catch (Exception e) {

e.printStackTrace();

}

}

//FORM TRANSAKSI

public void tglskrg(){

Date skrg= new Date();

SimpleDateFormat format= new SimpleDateFormat("dd MMMM yyyy");

SimpleDateFormat format2= new SimpleDateFormat("yyyy-MM-dd");

tgl = format2.format(skrg);

tgl\_transaksi.setText(format.format(skrg));

}

public void no\_transaksi(){

Connection con = koneksi.getConnection();

String sql = "SELECT id\_transaksi FROM transaksi";

try {

stt = con.createStatement();

rss = stt.executeQuery(sql);

if(rss.last()){

txt\_no\_transaksi.setText(String.valueOf(rss.getInt(1)+1));

}

else

txt\_no\_transaksi.setText("1");

} catch (Exception e) {

}

}

public void kode\_makanan(){

Connection con = koneksi.getConnection();

String sql = "SELECT \* FROM paket\_makanan";

try {

stt = con.createStatement();

rss= stt.executeQuery(sql);

while(rss.next()){

cmb\_kode.addItem(rss.getString("id\_paket"));

}

} catch (Exception e) {

}

}

private void tabel\_transaksi(){

model = new DefaultTableModel();

tbl\_transaksi.setModel(model);

model.addColumn("Id Detail Transaksi");

model.addColumn("No Transaksi");

model.addColumn("Id paket");

model.addColumn("Harga");

model.addColumn("Jumlah");

model.addColumn("Total Harga");

}

private void tampil\_tabel\_transaksi(){

Connection con = koneksi.getConnection();

try {

stt = con.createStatement();

String sql = "SELECT \* FROM detail\_transaksi WHERE id\_transaksi = '"+txt\_no\_transaksi.getText()+"'";

rss = stt.executeQuery(sql);

while(rss.next()){

Object[] o;

o = new Object[6];

o[0] = rss.getString(1);

o[1] = rss.getString(2);

o[2] = rss.getString(3);

o[3] = rss.getString(4);

o[4] = rss.getString(5);

o[5] = rss.getString(6);

model.addRow(o);

}

} catch (Exception e) {

e.printStackTrace();

}

}