

LAPORAN PRAKTIKUM
OBJEK ORIENTED PROGRAMMING
MENGEMBANGKAN APLIKASI DESKTOP
POLKAM MART DENGAN JAVA



DISUSUN OLEH :
NOVI WAHYUNI (201913014)

DOSEN PENGAMPU :
SLAMET TRIYANTO, S.T

KELAS : 1A TEKNIK INFORMATIKA

PROGRAM STUDI TEKNIK INFORMATIKA
POLITEKNIK KAMPAR
T.A 2019 / 2020

KATA PENGANTAR

Puji syukur kami panjatkan kepada Tuhan Yang Maha Esa, karena berkat rahmat-Nya penulis bisa menyelesaikan laporan praktek yang akan membahas mengenai **“Mengembangkan Aplikasi Desktop Polkam Mart Dengan Java”**.

Banyak sekali pihak yang telah membantu penulis dalam penyusunan laporan praktek ini, baik berupa bantuan materi ataupun berupa motivasi dan dukungan kepada penulis. Semua itu tentu terlalu banyak bagi penulis untuk membalasnya, namun pada kesempatan ini penulis hanya dapat mengucapkan terima kasih kepada :

1. Bapak Slamet Triyanto, S.T, selaku dosen pengampu mata kuliah Praktek Programming Berbasis Objek (PBO).
2. Sahabat dan teman – teman Teknik Informatika yang selalu memberi semangat.
3. Serta seluruh pihak yang tidak dapat disebutkan satu persatu, yang telah banyak membantu.

Laporan praktek ini masih jauh dari sempurna. Oleh karena itu, kritik dan saran yang bersifat membangun sangat penulis harapkan demi sempurnanya laporan ini.

Semoga laporan praktek ini memberikan informasi bagi teman-teman dan bermanfaat untuk pengembangan wawasan dan peningkatan ilmu pengetahuan bagi kita semua.

Bangkinang, 28 Juli 2020

Novi Wahyuni

DAFTAR ISI

| | |
|---|-----|
| KATA PENGANTAR | i |
| DAFTAR ISI..... | ii |
| DAFTAR GAMBAR | iii |
| BAB I TINJAUAN PUSTAKA | 1 |
| A. Landasan Teori..... | 1 |
| 1. Java..... | 1 |
| 2. NetBeans | 1 |
| B. Tujuan Pratikum..... | 2 |
| C. Alat dan Bahan..... | 2 |
| BAB II PEMBAHASAN | 3 |
| A. Import Basis Data/Database pada DBMS | 3 |
| B. Implementasi Pada Project Polkam Mart..... | 8 |
| 1. Design | 8 |
| 2. Sourcode..... | 12 |
| BAB III PENUTUP | 24 |
| A. Kesimpulan | 24 |
| B. Saran..... | 24 |
| DAFTAR PUSTAKA | v |

DAFTAR GAMBAR

| | |
|---|----|
| Gambar 2.1 Download File NetBeans Project | 3 |
| Gambar 2.2 Download File NetBeans Project 2 | 3 |
| Gambar 2.3 Xampp Control | 4 |
| Gambar 2.4 Menu User Akun Pengguna | 4 |
| Gambar 2.5 Add User Account | 5 |
| Gambar 2.6 Add User Account 2 | 6 |
| Gambar 2.7 Menu Impor | 6 |
| Gambar 2.8 Tampilan NetBeans | 7 |
| Gambar 2.9 Menu Open Project | 7 |
| Gambar 2.10 Tampilan Project | 8 |
| Gambar 2.11 Design Form Login | 8 |
| Gambar 2.12 Design Dashboard System | 9 |
| Gambar 2.13 Design Selamat Datang | 9 |
| Gambar 2.14 Design Data Barang | 10 |
| Gambar 2.15 Design Data Karyawan | 11 |
| Gambar 2.16 Design Edit Karyawan | 11 |
| Gambar 2.17 Design Tambah Karyawan | 12 |
| Gambar 2.18 Design Transaksi | 12 |
| Gambar 2.19 Sourcode From Login | 13 |
| Gambar 2.20 Sourcode Dashboard System | 13 |
| Gambar 2.21 Sourcode Dashboard System 2 | 13 |
| Gambar 2.22 Sourcode Dashboard System 3 | 14 |
| Gambar 2.23 Sourcode Dashboard System 4 | 14 |
| Gambar 2.24 Sourcode Selamat Datang | 15 |
| Gambar 2.25 Sourcode Selamat Datang 2 | 15 |
| Gambar 2.26 Sourcode Data Barang | 15 |
| Gambar 2.27 Sourcode Data Barang 2 | 16 |
| Gambar 2.28 Sourcode Data Barang 3 | 16 |
| Gambar 2.29 Sourcode Data Barang 4 | 16 |
| Gambar 3.30 Sourcode Data Barang 5 | 17 |

| | |
|--|----|
| Gambar 2.31 Sourcode Edit Karyawan..... | 17 |
| Gambar 2.32 Sourcode Edit Karyawan 2..... | 18 |
| Gambar 2.33 Sourcode Tambah Karyawan | 18 |
| Gambar 2.34 Sourcode Tambah Karyawan 2 | 18 |
| Gambar 2.35 Sourcode Transaksi | 19 |
| Gambar 2.36 Sourcode Transaksi 2 | 20 |
| Gambar 2.37 Sourcode Transaksi 3 | 20 |
| Gambar 2.38 Sourcode Transaksi 4 | 21 |
| Gambar 2.39 Sourcode Transaksi 5 | 21 |
| Gambar 2.40 Sourcode Transaksi 6 | 21 |
| Gambar 2.41 Sourcode Transaksi 7 | 22 |

BAB I

TINJAUAN PUSTAKA

A. Landasan Teori

1. Java

Menurut wikipedia, Java adalah bahasa pemrograman yang dapat dijalankan di berbagai komputer termasuk telepon genggam. Bahasa ini awalnya dibuat oleh James Gosling saat masih bergabung di Sun Microsystems saat ini merupakan bagian dari Oracle dan dirilis tahun 1995. Bahasa ini banyak mengadopsi sintaksis yang terdapat pada C dan C++ namun dengan sintaksis model objek yang lebih sederhana serta dukungan rutin-rutin aras bawah yang minimal. Aplikasi-aplikasi berbasis java umumnya dikompilasi ke dalam p-code (*bytecode*) dan dapat dijalankan pada berbagai Mesin Virtual Java (JVM). Java merupakan bahasa pemrograman yang bersifat umum/non-spesifik (*general purpose*), dan secara khusus didisain untuk memanfaatkan dependensi implementasi seminimal mungkin. Karena fungsionalitasnya yang memungkinkan aplikasi java mampu berjalan di beberapa platform sistem operasi yang berbeda, java dikenal pula dengan slogannya, "*Tulis sekali, jalankan di mana pun*". Saat ini java merupakan bahasa pemrograman yang paling populer digunakan, dan secara luas dimanfaatkan dalam pengembangan berbagai jenis perangkat lunak aplikasi ataupun aplikasi.

2. NetBeans

Menurut wikipedia, NetBeans adalah suatu serambi pengembangan perangkat lunak yang ditulis dalam bahasa pemrograman Java. Serambi Pada NetBeans, pengembangan suatu aplikasi dapat dilakukan dimulai dari setelan perangkat lunak modular bernama modules. Semula, aplikasi NetBeans IDE ini diperuntukkan bagi pengembangan dalam Java. Namun, aplikasi ini juga mendukung program-program pembuatan bahasa lain secara khusus seperti PHP, C/C++ dan HTML5. NetBeans adalah alat lintas serambi serta penerapannya dijalankan pada Microsoft Windows, Mac OS

X, Linux, Solaris dan serambi-serambi lainnya yang mendukung JVM yang sepadan.

Awal kemunculan aplikasi NetBeans telah ada sejak tahun 1997 yaitu sebagai sebuah proyek kuliah. Pada tahun tersebut, suatu perusahaan dibangun oleh Roman Staněk di sekitar proyek kuliah tersebut lalu perusahaan tersebut memulai memproduksi versi NetBeans IDE komersial hingga akhirnya dibeli oleh Sun Microsystems pada tahun 1999 lalu menjadikan NetBeans IDE sebagai serambi bersifat sumber terbuka pada bulan Juni 1999.

B. Tujuan Praktek

1. Mengetahui aplikasi yang digunakan untuk mengembangkan aplikasi desktop polkam mart.
2. Mengetahui fungsi – fungsi dasar aplikasi Netbeans yang digunakan untuk mengembangkan aplikasi desktop polkam mart.

C. Alat dan Bahan

1. Alat
 - Komputer / Laptop.
 - Mouse.
2. Bahan
 - Aplikasi JDK.
 - Aplikasi NetBeans.
 - NetBeans project yang telah didownload.

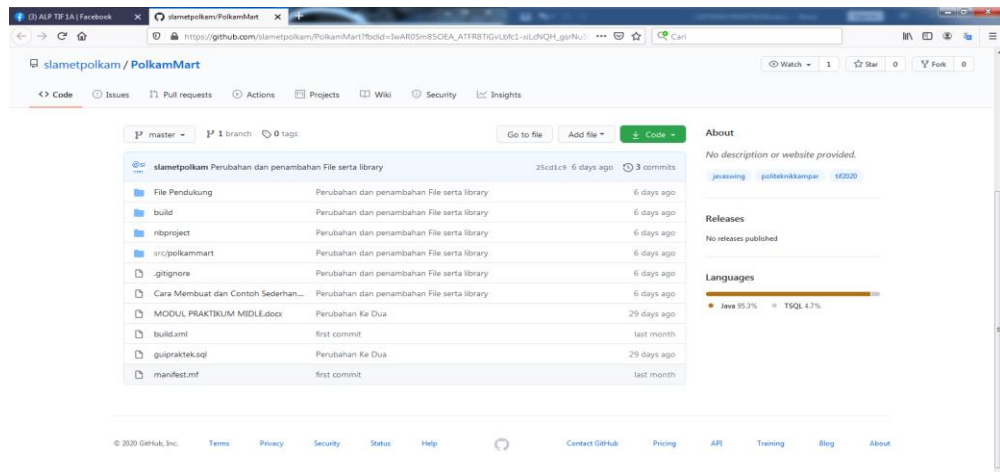
BAB II

PEMBAHASAN

A. Import Basis Data/Database pada (DBMS)

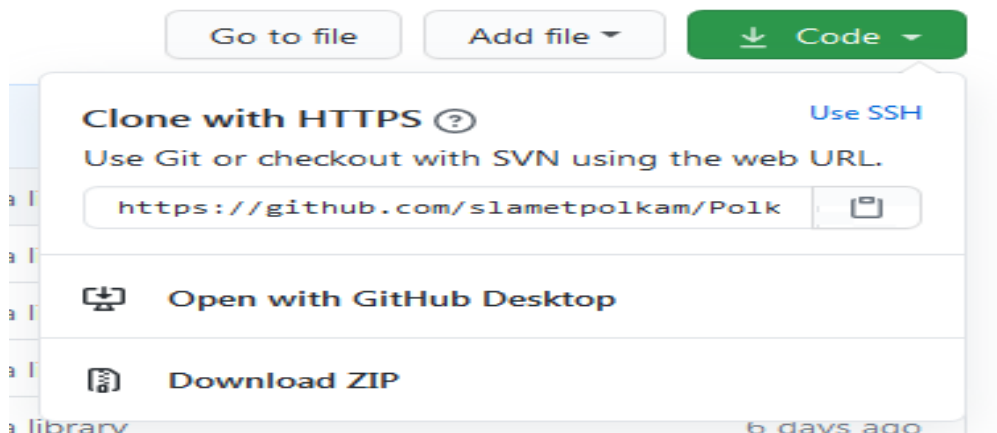
Untuk mengimport basis data/database pada DBMS, pastikan terlebih dahulu anda telah instal xampp pada komputer anda dan mempunyai database yang akan diimportkan. Berikut cara mengimportkan database pada DBMS.

- Pastikan anda telah memiliki netbeans project yang telah anda buat atau bisa juga Download file netbeans project nya di <https://github.com/slametpolkam/PolkamMart.git>



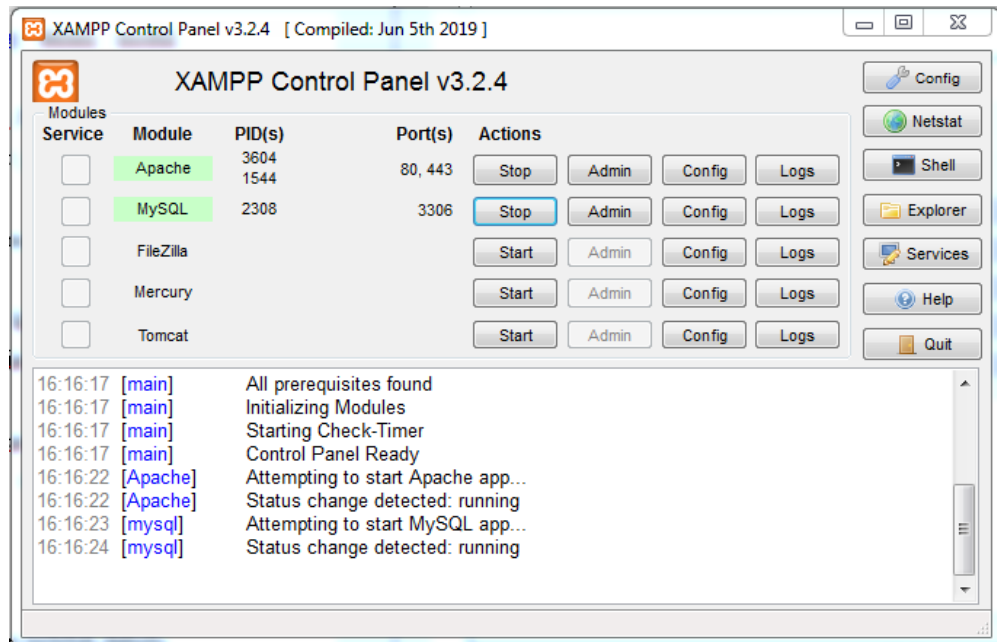
Gambar 2.1 Download File NetBeans Project

- Download bentuk .zip



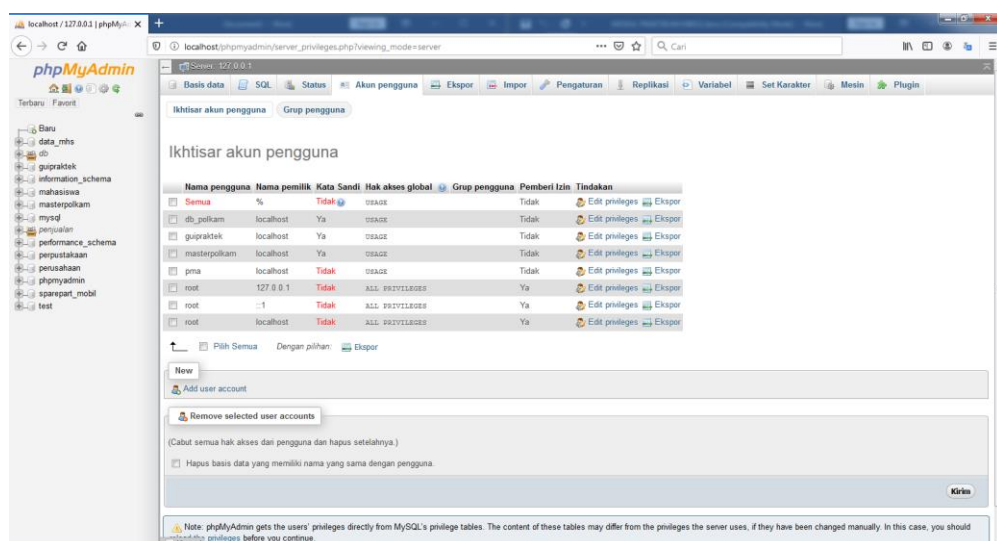
Gambar 2.2 Download File NetBeans Project 2

- Setelah di download dalam bentuk .zip, lalu ekstrak dalam 1 folder.
- Selanjutnya, jalankan xampp control.
- Setelah xampp control dijalankan, klik “Start” pada Apache dan MySQL.



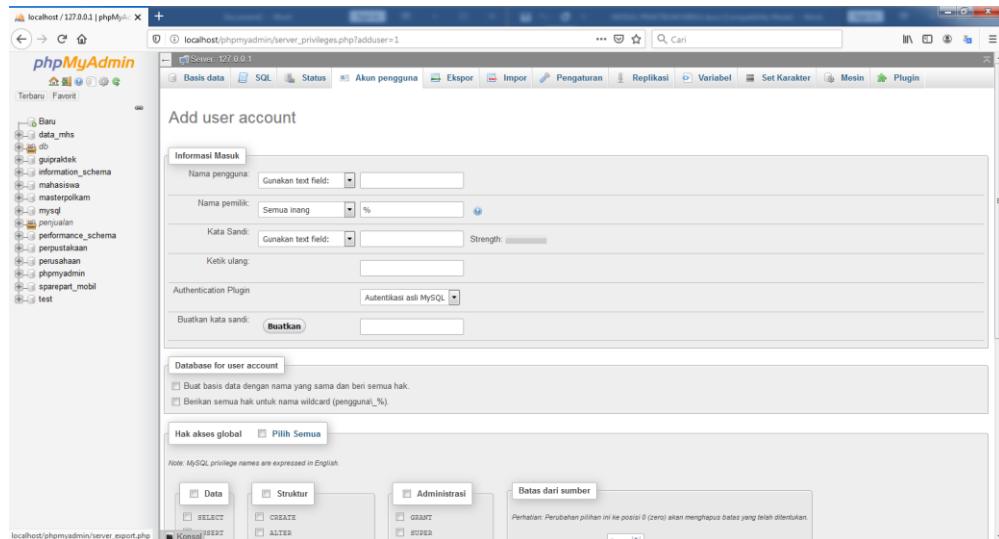
Gambar 2.3 Xampp Control

- Selanjutnya dengan browser internet anda, silahkan akses halaman “localhost/phpmyadmin”.
- Selanjutnya adalah membuat basis data. Pilih menu **Akun Pengguna > Add user account**.



Gambar 2.4 Menu User Akun Pengguna

- Selanjutnya akan diarahkan ke halaman Add user account, dengan tampilan sebagai berikut :

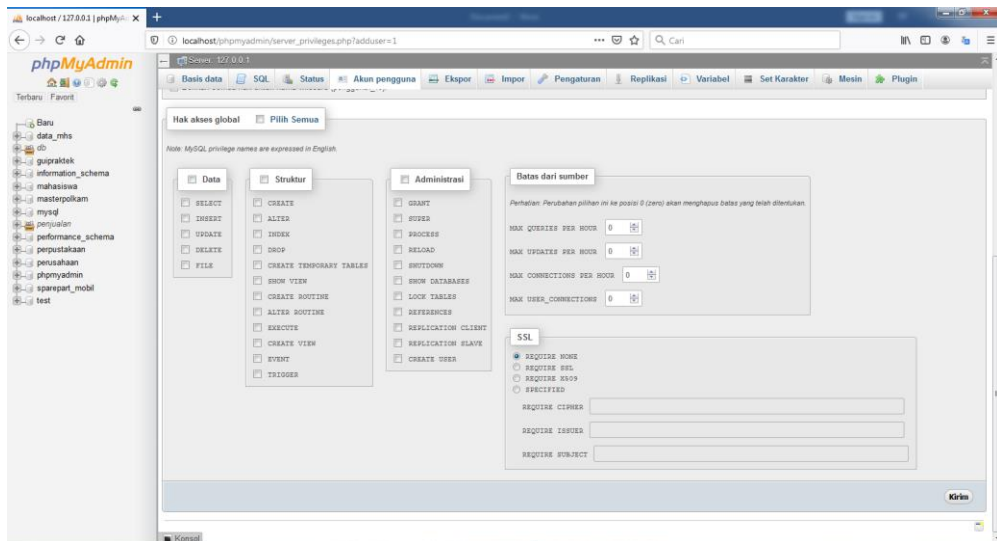


Gambar 2.5 Add User Account

Note :

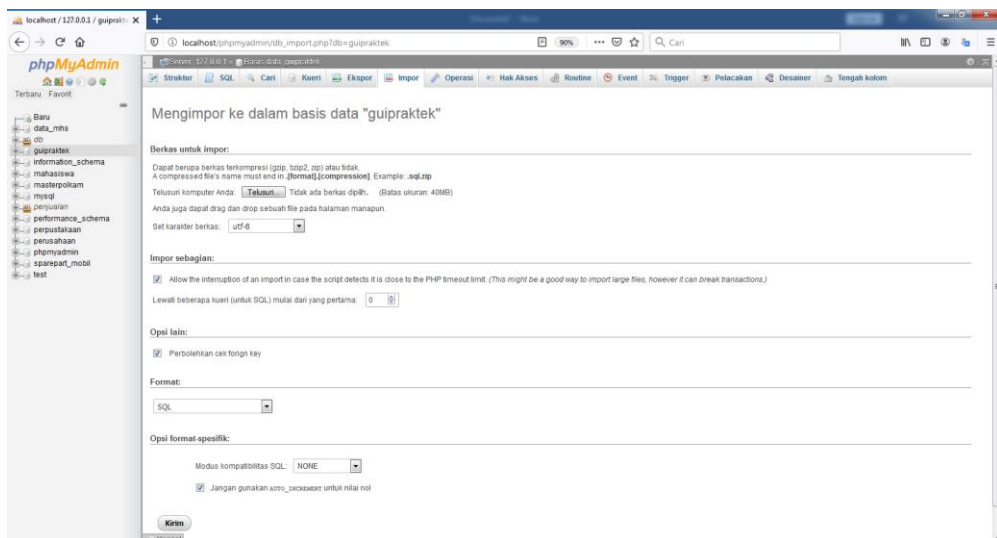
Isi setiap kolom dengan hal sebagai berikut :

1. Kolom Nama Pengguna di isi dengan **guipraktek**.
 2. Kolom Nama Pemilik pilih Hostname **Local**.
 3. Kolom Kata Sandi buat sandi yang sama dengan nama username yakni **guipraktek**.
 4. Kolom Kata Ulang tulis password ulang
 5. Kolom Database for user account ceklis **Buat basis data dengan nama yang sama dan beri semua hak**.
- Jika telah selesai, scrool kebawah hingga menampilkan tulisan **Go/Kirim**.



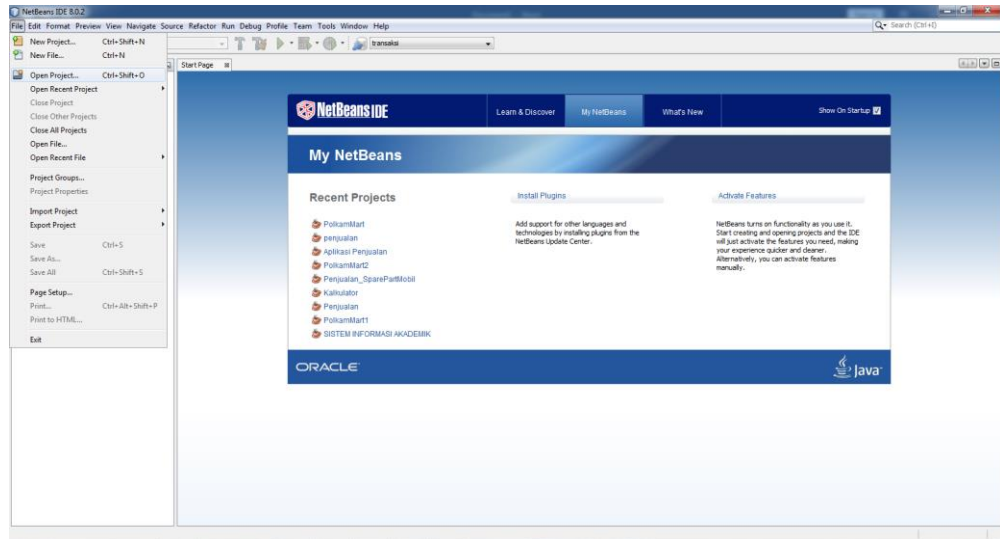
Gambar 2.6 Add User Account 2

- Lalu pada database **guipraktek** pilih menu **Import** > **Telusuri** (lihat file yang telah di download dan di ekstrak dalam bentuk folder, cari tulisan **guipraktek.sql**), lalu klik **Kirim**.



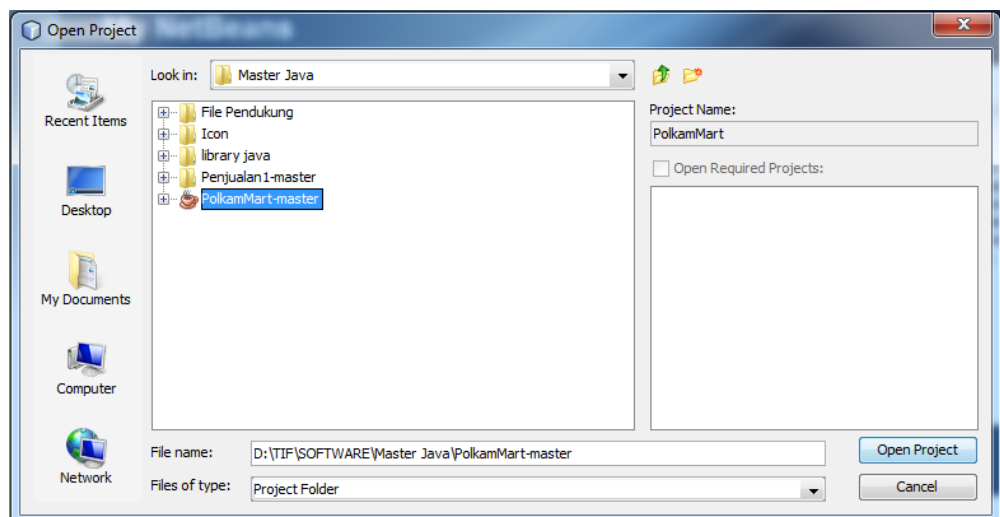
Gambar 2.7 Menu Import

- Lalu buka netbeans, pilih **File** > **Open Project**.



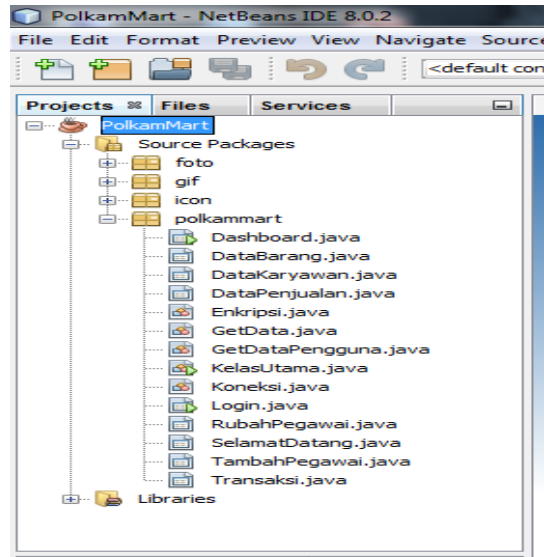
Gambar 2.8 Tampilan NetBeans

- Masukkan project yang telah diunduh, lalu klik **Open Project**.



Gambar 2.9 Menu Open Project

- Lalu tampilannya seperti ini.



Gambar 2.10 Tampilan Project

B. Implementasi Pada Project Polkam Mart

Pada project polkam mart ini yaitu mengubah/menambah/memperbaharui design dan sourcode yang ada. Berikut hasil dari implementainya :

1. Design

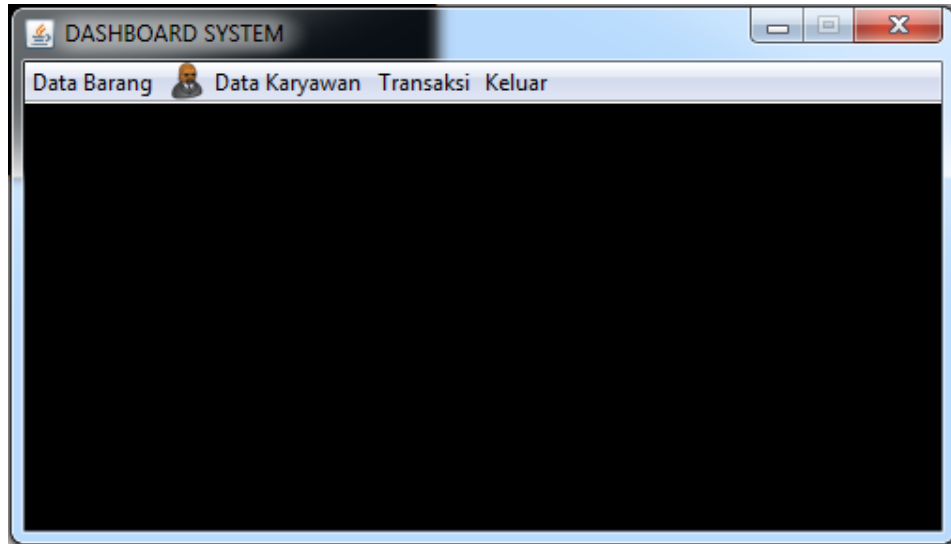
- From Login.

Pada design from login ini menggunakan palette antara lain : JFrame, Panel, label, text field, button, password field, dan check box. Juga memasukkan foto dan icon.

Gambar 2.11 Design Form Login

- Dashboard System

Pada design dashboar system ini menggunakan palette antara lain : JFrame, Menu Bar, Menu, dan Desktop Pane. Juga memasukkan icon.



Gambar 2.12 Design Dashboard System

- Selamat Datang

Pada design selamat datang system ini menggunakan palette antara lain : JInternalFrame dan label. Juga memasukkan foto, icon, dan gif.



Gambar 2.13 Design Selamat Datang

- Data Barang

Pada design data barang system ini menggunakan palette antara lain : JInternalFrame, panel, label, button, text field, dan scroll pane. Juga memasukkan foto dan icon.

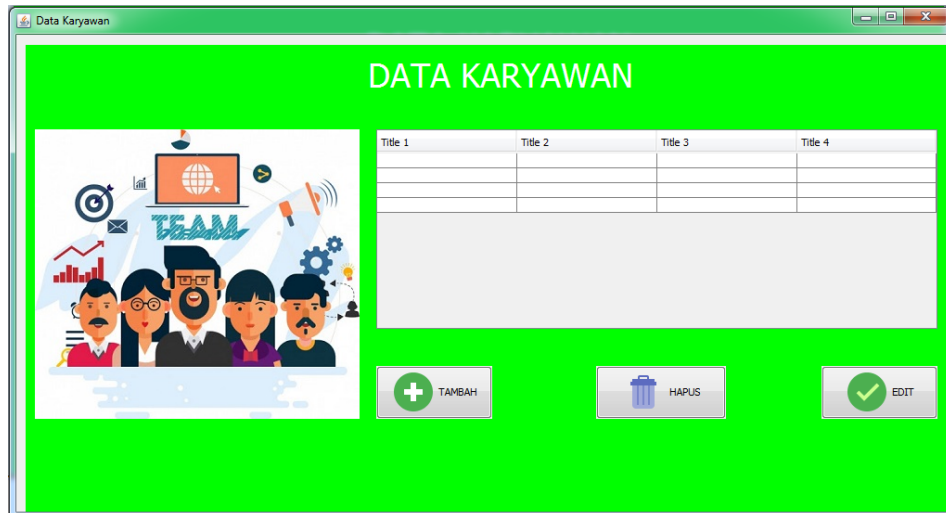
The screenshot shows a Java Swing window titled "Data Barang" with a red background. The window contains the following elements:

- Labels and Input Fields:** On the left side, there are four labels: "Barcode", "Nama Barang", "Harga Barang", and "Stok Barang", each followed by a white text input field.
- Central Image:** In the center, there is a white square containing a colorful illustration of a shopping basket filled with various items like a water bottle, fruits, and vegetables.
- Action Buttons:** On the right side, there are three buttons: "HAPUS" (with a trash can icon), "SIMPAN" (with a green plus icon), and "EDIT" (with a green checkmark icon).
- Table:** At the bottom of the window, there is a table with 4 columns and 5 rows. The columns are labeled "Title 1", "Title 2", "Title 3", and "Title 4".

Gambar 2.14 Design Data Barang

- Data Karyawan

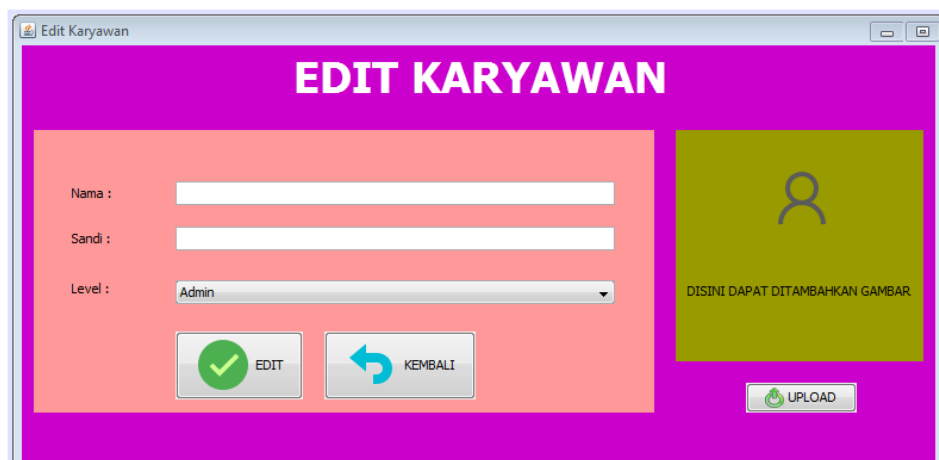
Pada design data karyawan system ini menggunakan palette antara lain : JInternalFrame, panel, label, scroll pane, dan button. Juga memasukkan foto dan icon.



Gambar 2.15 Design Data Karyawan

- Edit Karyawan

Pada design edit karyawan system ini menggunakan palette antara lain : JInternalFrame, panel, text field, label, combo box, dan button. Juga memasukkan foto dan icon.



Gambar 2.16 Design Edit Karyawan

- Tambah Karyawan

Pada design tambah karyawan system ini menggunakan palette antara lain : JInternalFrame, panel, text field, label, combo box, dan button. Juga memasukkan foto dan icon.

Gambar 2.17 Design Tambah Karyawan

- Transaksi

Pada design transaksi system ini menggunakan palette antara lain : JFrame, panel, button, label, text field, button, combo box, dan scroll pane. Juga memasukkan foto dan icon.

Gambar 2.18 Design Transaksi

2. Sourcode

- Form Login

Pada sourcode from login yang ditambahkan yaitu untuk melihat/mencek password yang telah dimasukkan dan tombol batal.

```

private void cekPasswordActionPerformed(java.awt.event.ActionEvent evt) {
    if (cekPassword.isSelected()) {
        jPasswordField1.setEchoChar((char) 0);
    } else {
        jPasswordField1.setEchoChar('*');
    }
}

private void jButton2MouseClicked(java.awt.event.MouseEvent evt) {
    // TODO add your handling code here:
    Login login = new Login();
    login.setVisible(true);
    this.setVisible(false);
}

```

Gambar 2.19 Sourcode From Login

- Dashboard System

Pada sourcode dashboard system yang ditambahkan yaitu fungsi menu transaksi dan keluar.

```

public class Dashboard extends javax.swing.JFrame {

    /**
     * Creates new form Dashboard
     */
    SelamatDatang sel;
    DataKaryawan peg;
    TambahPegawai tamPeg;
    DataBarang bar;
    Transaksi trans;
    GetData data=new GetData();
}

```

Gambar 2.20 Sourcode Dashboard System

```

public Dashboard() {
    initComponents();
    setExtendedState(getExtendedState() | Dashboard.MAXIMIZED_BOTH);

    /**
     * Membuat Instan dari kelas SelamatDatang.java
     * kemudian menambahkan DesktopPane dengan frame SelamatDatang
     * dan membuat full screen dari DesktopPane
     */

    sel=new SelamatDatang();
    peg=new DataKaryawan();
    bar=new DataBarang();
    tamPeg=new TambahPegawai();
    trans=new Transaksi();
    DesktopPane.add(tamPeg);
    DesktopPane.add(peg);
    DesktopPane.add(sel);
    DesktopPane.add(bar);
    DesktopPane.add(trans);

    DesktopPane.moveToFront(sel);
    sel.setVisible(true);
    sel.setSize(DesktopPane.getWidth(),DesktopPane.getHeight());
    try {
        sel.setMaximum(true);
    } catch (PropertyVetoException ex) {
        Logger.getLogger(SelamatDatang.class.getName()).log(Level.SEVERE, null, ex);
    }
    sel.setLocation(0,0);
}

```

Gambar 2.21 Sourcode Dashboard System 2

```

private void DataBarangMouseClicked(java.awt.event.MouseEvent evt) {
    // TODO add your handling code here:
    sel.setVisible(false);
    peg.setVisible(false);
    bar.setVisible(true);
    bar.setSize(DesktopPane.getWidth(),DesktopPane.getHeight());
    try {
        bar.setMaximum(true);
    } catch (PropertyVetoException ex) {
        Logger.getLogger(DataKaryawan.class.getName()).log(Level.SEVERE, null, ex);
    }
    bar.setLocation(0,0);
}

private void DataKaryawanMouseClicked(java.awt.event.MouseEvent evt) {
    // TODO add your handling code here:
    sel.setVisible(false);
    peg.setVisible(true);
    peg.setSize(DesktopPane.getWidth(),DesktopPane.getHeight());
    try {
        peg.setMaximum(true);
    } catch (PropertyVetoException ex) {
        Logger.getLogger(DataKaryawan.class.getName()).log(Level.SEVERE, null, ex);
    }
    peg.setLocation(0,0);
}

```

Gambar 2.22 Sourcode Dashboard System 3

```

private void TransaksiMouseClicked(java.awt.event.MouseEvent evt) {
    // TODO add your handling code here:
    sel.setVisible(false);
    peg.setVisible(false);
    bar.setVisible(false);
    trans.setVisible(true);
    trans.setSize(DesktopPane.getWidth(),DesktopPane.getHeight());
    try {
        trans.setMaximum(true);
    } catch (PropertyVetoException ex) {
        Logger.getLogger(DataKaryawan.class.getName()).log(Level.SEVERE, null, ex);
    }
    trans.setLocation(0,0);
}

private void KeluarMouseClicked(java.awt.event.MouseEvent evt) {
    // TODO add your handling code here:
    Login log= new Login();
    log.setVisible(true);
    this.setVisible(false);
}

```

Gambar 2.23 Sourcode Dashboard System 4

- Selamat Datang

Pada sourcode selamat datang yang ditambahkan yaitu menampilkan jam dan tanggal.

```
package polkammart;

import java.beans.PropertyVetoException;
import java.util.Calendar;
import java.util.GregorianCalendar;
import java.util.logging.Level;
import java.util.logging.Logger;
```

Gambar 2.24 Sourcode Selamat Datang

```
public SelamatDatang() {
    initComponents();
    jLabel2.setText("POLKAM MART");

    new Thread(){
        public void run(){
            while(true){
                Calendar kal = new GregorianCalendar();
                String namabulan[] = {"Januari", "Februari", "Maret", "April", "Mei", "Juni", "Juli", "Agustus", "September", "Oktober", "November", "Desember"};
                int tahun = kal.get(Calendar.YEAR);
                int bulan = kal.get(Calendar.MONTH);
                int hari = kal.get(Calendar.DAY_OF_MONTH);
                int jam = kal.get(Calendar.HOUR_OF_DAY);
                int menit = kal.get(Calendar.MINUTE);
                int detik = kal.get(Calendar.SECOND);
                String tanggal = hari+" "+namabulan[bulan]+" "+tahun;
                String waktu = jam+"."+menit+"."+detik;

                labelTanggal.setText(tanggal);
                labelJam.setText(waktu);
            }
        }
    }.start();
}
```

Gambar 2.25 Sourcode Selamat Datang 2

- Data Barang

Pada sourcode data barang yang ditambahkan yaitu mengisi barcode secara otomatis dan pencarian barang yang diinginkan.

```
package polkammart;

import java.sql.Connection;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import java.util.Random;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.swing.JOptionPane;
import javax.swing.table.DefaultTableModel;
```

Gambar 2.26 Sourcode Data Barang

```

public class DataBarang extends javax.swing.JInternalFrame {
    public Statement st;
    public ResultSet res;
    /**
     * Creates new form DataBarang
     */
    private DefaultTableModel model;
    Connection cn = polkamart.Koneksi.getKoneksi();
    Object[ ] obj;
    public DataBarang() {
        initComponents();
        AmbilData();
        Barcode();
        Ulang();
    }
}

```

Gambar 2.27 Sourcode Data Barang 2

```

private void Barcode() {
    try{
        Random angka = new Random();
        String angkaa = String.valueOf(angka.nextInt(999999999));
        String barcode = "B0"+angkaa;
        Barcode.setText(barcode);
        Barcode.setEnabled(false);
    }catch(Exception e){
        JOptionPane.showMessageDialog(null, e.getMessage());
    }
}

public void Ulang() {
    NamaBarang.setText("");
    Harga.setText("");
    Stok.setText("");
    simpan.setEnabled(true);
    ubah.setEnabled(true);
    hapus.setEnabled(true);
}

```

Gambar 2.28 Sourcode Data Barang 3

```

private void tcariKeyReleased(java.awt.event.KeyEvent evt) {
    // TODO add your handling code here:
    model=new DefaultTableModel();
    tabelBarang.setModel(model);
    model.addColumn("No");
    model.addColumn("Barcode");
    model.addColumn("Nama Barang");
    model.addColumn("Harga Satuan");
    model.addColumn("Stok");

    model.getDataVector().removeAllElements();
    model.fireTableDataChanged();
    int i=1;
    try{
        //membuat statemen pemanggilan data pada tabel tblGaji dari database
        Statement stat = (Statement) Koneksi.getKoneksi().createStatement();

        String sql = " select * from barang where barcode like '"+tcari.getText()+"%' or nama_barang like '"+tcari.getText()+"%' or harga like '"+tcari.getText()+"%' or stok like '"+tcari.getText()+"%' ";
        ResultSet res = stat.executeQuery(sql);
        //penelusuran baris pada tabel tblGaji dari database
        while(res.next()){
            obj= new Object[5];

            obj[0] =i;
            obj[1] = res.getString("barcode");
            obj[2] = res.getString("nama_barang");
            obj[3] = res.getString("harga");
            obj[4] = res.getString("stok");

            model.addRow(obj);
            i++;
        }
    }catch(SQLException err){
        JOptionPane.showMessageDialog(null, err.getMessage());
    }
}

```

Gambar 2.29 Sourcode Data Barang 4

```

private void simpanActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:

    String z=Barcode.getText();
    String y>NamaBarang.getText();
    String l=Harga.getText();
    String j=Stok.getText();
    try {
        Statement stat = (Statement) Koneksi.getKoneksi().createStatement();
        String sql = "INSERT INTO barang VALUES ('"+z+"', '"+y+"', '"+l+"', '"+j+"', '')";
        stat.executeUpdate(sql);
        System.out.println(sql);
        JOptionPane.showMessageDialog(this, "Penambahan Data Berhasil ");
        AmbilData();
        Barcode();
        Ulang();
    } catch (SQLException ex) {
        Logger.getLogger(TambahPegawai.class.getName()).log(Level.SEVERE, null, ex);
    }
}

```

Gambar 3.30 Sourcode Data Barang 5

- Edit Karyawan

Pada sourcode edit karyawan yang ditambahkan yaitu mengupload / memasukkan foto.

```

package polkammart;

import java.awt.Image;
import java.io.ByteArrayOutputStream;
import java.io.File;
import java.io.FileInputStream;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.swing.ImageIcon;
import javax.swing.JFileChooser;
import javax.swing.JOptionPane;

/**
 *
 * @author oryza
 */
public class RubahPegawai extends javax.swing.JInternalFrame {
    String namafile = null;
    byte[] fotokaryawan = null;

```

Gambar 2.31 Sourcode Edit Karyawan

```

private void UploadActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    JFileChooser file = new JFileChooser();
    file.showOpenDialog(null);
    File f = file.getSelectedFile();
    namafile = f.getAbsolutePath();
    ImageIcon imageicon = new ImageIcon(new ImageIcon(namafile).getImage().getScaledInstance(Image.getWidth(), Image.getHeight(), Image.SCALE_SMOOTH));
    Image.setIcon(imageicon);

    try{
        File image = new File(namafile);
        FileInputStream tis = new FileInputStream(image);
        ByteArrayOutputStream bos = new ByteArrayOutputStream();
        byte[] buf = new byte[1024];
        for(int readnum; (readnum = tis.read(buf)) != -1;){
            bos.write(buf, 0, readnum);
        }
    }
    catch (Exception e) {
        JOptionPane.showMessageDialog(null, e);
    }
}

```

Gambar 2.32 Sourcode Edit Karyawan 2

- Tambah Karyawan

Pada sourcode tambah karyawan yang ditambahkan yaitu mengupload / memasukkan foto.

```

package polkammart;

import java.awt.Image;
import java.io.ByteArrayOutputStream;
import java.io.File;
import java.io.FileInputStream;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.swing.ImageIcon;
import javax.swing.JFileChooser;
import javax.swing.JOptionPane;

/**
 *
 * @author oryza
 */
public class TambahPegawai extends javax.swing.JInternalFrame {
    String namafile = null;
    byte[] fotokaryawan = null;
}

```

Gambar 2.33 Sourcode Tambah Karyawan

```

private void UploadActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    JFileChooser file = new JFileChooser();
    file.showOpenDialog(null);
    File f = file.getSelectedFile();
    namafile = f.getAbsolutePath();
    ImageIcon imageicon = new ImageIcon(new ImageIcon(namafile).getImage().getScaledInstance(Image.getWidth(), Image.getHeight(), Image.SCALE_SMOOTH));
    Image.setIcon(imageicon);

    try{
        File image = new File(namafile);
        FileInputStream tis = new FileInputStream(image);
        ByteArrayOutputStream bos = new ByteArrayOutputStream();
        byte[] buf = new byte[1024];
        for(int readnum; (readnum = tis.read(buf)) != -1;){
            bos.write(buf, 0, readnum);
        }
    }
    catch (Exception e) {
        JOptionPane.showMessageDialog(null, e);
    }
}

```

Gambar 2.34 Sourcode Tambah Karyawan 2

- Transaksi

Pada soucode transaksi terdiri dari : pemilihan nama barang yang direlasikan ke database tabel barang dan akan menampilkan hasilnya secara otomatis, menghitung jumlah pembelian, menambahkan jumlah pembelian apabila pembelian lebih dari satu dan akan disimpan kedalam tabel, menghapus jumlah pembelian yang ada tabel, menjumlahkan total pembelian keseluruhan nya, dan melakukan pembayaran.

```
package polkammart;

import java.sql.Connection;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import java.time.LocalDateTime;
import java.time.format.DateTimeFormatter;
import java.util.Random;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.swing.JOptionPane;
import javax.swing.table.DefaultTableModel;

public class Transaksi extends javax.swing.JInternalFrame {
    private DefaultTableModel model;
    public Statement st;
    public ResultSet rs;
    Connection cn = polkammart.Koneksi.getKoneksi();
    //public long bayarr;
    //public long totall;

    public Transaksi() {
        initComponents();
        TampilData();
        Judul();
        Reset();
        kd_transaksi();
        Barang();
        //kd_barcode();

        KodeTransaksi.setEnabled(false);
        Barcode.setEnabled(false);
        Harga.setEnabled(false);
        Stok.setEnabled(false);
        Tambah.setEnabled(false);
        Kembalian.setEnabled(false);
        Selesai.setEnabled(false);
    }
}
```

Gambar 2.35 Sourcode Transaksi


```

private void TampilData(){
    try {
        st = cn.createStatement();
        model.getDataVector().removeAllElements();
        model.fireTableDataChanged();
        rs = st.executeQuery("select * from keranjang");
        while (rs.next()){
            Object[] data = {
                rs.getString("nama_barang"),
                rs.getString("harga_barang"),
                rs.getString("jumlah_barang"),
                rs.getString("harga")
            };
            model.addRow(data);
        }
    } catch (Exception e) {
        e.printStackTrace();
    }
}

public void Judul(){
    Object[] judul = {"Nama Barang","Harga Barang","Jumlah Barang","Harga"};
    model = new DefaultTableModel(null, judul);
    tabelbarang.setModel(model);
}

private void Reset(){
    Barcode.setText("");
    NamaBarang.setSelectedItem(null);
    HargaBarang.setText("");
    Stok.setText("");
    JumlahBarang.setText("");
    Harga.setText("");
    KodeTransaksi.setText("");
}

```

Gambar 2.36 Sourcode Transaksi 2

```

private void kd_transaksi(){
    try{
        java.util.Date tgl = new java.util.Date();
        java.text.SimpleDateFormat kal = new java.text.SimpleDateFormat("yyMMdd");
        java.text.SimpleDateFormat tanggal = new java.text.SimpleDateFormat("yyyyMMdd");

        String sql = "Select max(no_transaksi) from transaksi where tanggal =" + tanggal.format(tgl);
        st = cn.createStatement();
        rs = st.executeQuery(sql);
        while(rs.next()){
            Long a = rs.getLong(1);
            if (a == 0) {
                this.KodeTransaksi.setText(kal.format(tgl)+"0000"+(a+1));
            } else {
                this.KodeTransaksi.setText(""+(a+1));
            }
        }
    } catch (Exception e) {
        JOptionPane.showMessageDialog(null, e.getMessage());
    }
}

public void Barang(){
   >NamaBarang.removeAllItems();
   >NamaBarang.addItem("Pilih Barang");
    try {
        st = cn.createStatement();
        rs = st.executeQuery("select * from barang");
        while (rs.next()) {
           >NamaBarang.addItem(rs.getString("nama_barang"));
        }
    } catch (Exception e) {
        e.printStackTrace();
    }
}

```

Gambar 2.37 Sourcode Transaksi 3

```

private void TambahActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    if("").equals(JumlahBarang.getText()){
        JOptionPane.showMessageDialog(null, "Lengkapi Data!");
    }else{
        try {
            st=cn.createStatement();
            st.executeUpdate("insert into keranjang set " + " nama_barang ='" + NamaBarang.getSelectedItemAt()
                + "', harga_barang = '" + HargaBarang.getText() + "', jumlah_barang = '"
                + JumlahBarang.getText() + "', harga = '" + Harga.getText() + "';");
            TampilData();
            Reset();

            int total = 0;
            for (int i = 0; i < tabelbarang.getRowCount(); i++) {
                int amount = Integer.parseInt((String)tabelbarang.getValueAt(i, 3));
                total += amount;
            }
            this.Total.setText("" + total);

            JOptionPane.showMessageDialog(null, "Berhasil");
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}

```

Gambar 2.38 Sourcode Transaksi 4

```

private void HitungActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    if("").equals(HargaBarang.getText())||"".equals(Stok.getText())||"".equals(JumlahBarang.getText()){
        JOptionPane.showMessageDialog(null, "Lengkapi Data Anda");
    }else{
        Integer jumlah = Integer.parseInt(JumlahBarang.getText());
        Integer stok = Integer.parseInt(this.Stok.getText());
        if (jumlah > stok){
            JOptionPane.showMessageDialog(null, "Stok Hanya Ada "+stok);
        }else{
            int a = Integer.parseInt(HargaBarang.getText());
            int b = Integer.parseInt(JumlahBarang.getText());
            int hitung = a * b;
            Harga.setText(""+hitung);
            Tambah.setEnabled(true);
            Hitung.setEnabled(true);
        }
    }
}

```

Gambar 2.39 Sourcode Transaksi 5

```

private void HargaBarangActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    try {
        HargaBarang.setText("");
        st = cn.createStatement();
        rs = st.executeQuery("select * from barang where harga_barang = '" + HargaBarang.getText() + "'");
    } catch (Exception e) {
        JOptionPane.showMessageDialog(null, e.getMessage());
    }
}

private void NamaBarangActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    try {
        HargaBarang.setText("");
        Stok.setText("");
        st = cn.createStatement();
        rs = st.executeQuery("select * from barang where nama_barang = '" + NamaBarang.getSelectedItemAt() + "'");
        while (rs.next()) {
            if ("Pilih Barang".equals(NamaBarang.getSelectedItemAt())) {
                HargaBarang.setText("");
                Stok.setText("");
            } else {
                Barcode.setText(rs.getString("barcode"));
                HargaBarang.setText(rs.getString("harga"));
                Stok.setText(rs.getString("stok"));
            }
        }
    } catch (Exception e) {
        JOptionPane.showMessageDialog(null, e.getMessage());
    }
}

```

Gambar 2.40 Sourcode Transaksi 6

```

private void JumlahBarangKeyTyped(java.awt.event.KeyEvent evt) {
    // TODO add your handling code here:
    if(!Character.isDigit(evt.getKeyChar())){
        evt.consume();
    }
}

private void tabelbarangMouseClicked(java.awt.event.MouseEvent evt) {
    // TODO add your handling code here:
    NamaBarang.setSelectedItem(model.getValueAt(tabelbarang.getSelectedRow(), 0) + "");
    HargaBarang.setText(model.getValueAt(tabelbarang.getSelectedRow(), 1) + "");
    JumlahBarang.setText(model.getValueAt(tabelbarang.getSelectedRow(), 2) + "");
    Harga.setText(model.getValueAt(tabelbarang.getSelectedRow(), 3) + "");
}

private void HapusActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    try {
        int jawab;
        if ((jawab=JOptionPane.showConfirmDialog(null, "Hapus ?", "konfirmasi", JOptionPane.YES_NO_OPTION))==0) {
            st.executeUpdate("delete from barang where "+"barcode='"+model.getValueAt(tabelbarang.getSelectedRow(), 0)+"'");
            TampilData();
            Reset();
            JOptionPane.showMessageDialog(null, "Berhasil dihapus");
        }
    } catch (Exception e) {
        JOptionPane.showMessageDialog(null, "Gagal diHapus! "+e.getMessage());
    }
}

private void PembayaranKeyReleased(java.awt.event.KeyEvent evt) {
    // TODO add your handling code here:
    int kmbl = 0;
    kmbl = Integer.parseInt(Pembayaran.getText())-Integer.parseInt(Total.getText());
    Kembalian.setText(String.valueOf(kmbl));
}

```

Gambar 2.41 Sourcode Transaksi 7

BAB III

PENUTUP

A. Kesimpulan

1. Mengembangkan aplikasi desktop polkam mart dengan menggunakan NetBeans telah berhasil dibuat.
2. Mengembangkan aplikasi desktop polkam mart ini menggunakan palette, icon, foto, dan gif.

B. Saran

Adapun beberapa saran yang dapat disampaikan pada Laporan Praktek ini sebagai berikut :

1. Perlu adanya pengembangan lebih lanjut supaya aplikasi desktop polkam mart ini bisa digunakan dengan maksimal.
2. Untuk lebih menyempurnakan pengembangan aplikasi desktop polkam mart yang telah dibuat ini, diharapkan kedepannya menambahkan source code atau design supaya aplikasi desktop polkam mart tersebut bisa digunakan dengan maksimal.

DAFTAR PUSTAKA

<https://id.wikipedia.org/wiki/Java>

<https://id.wikipedia.org/wiki/NetBeans>