

LAPORAN

**“PEMBUATAN PROGRAM POLKAM MART MENGGUNAKAN
APLIKASI NETBEANS”**



Disusun Oleh :

KHAIRUNNISA’

NIM 201913001

DOSEN PENGAMPU : SLAMET TRIYANTO, S.ST

PEMOGRAMAN BERORIENTASI OBJEK (PBO)

POLITEKNIK KAMPAR

2020

KATA PENGANTAR

Puji syukur saya ucapkan kepada Allah SWT atas segala rahmat dan karunia- Nya, sehingga penulis telah dapat menyelesaikan laporan dengan judul **“PEMBUATAN PROGRAM POLKAM MART MENGGUNAKAN APLIKASI NETBEANS”** ini dengan baik dan tepat waktu yang dibuat berdasarkan hasil kegiatan harian selama belajar di Kampus Politeknik Kampar.

Dalam proses pengerjaan tugas ini, penulis melakukan beberapa penelitian yang tak lupa mendapatkan bimbingan, arahan dan pengetahuan hingga penulis mampu menyelesaikan tugas ini dengan baik. Maka dari itu, kami mengucapkan terima kasih kepada pihak-pihak yang telah membantu dalam penyelesaian Laporan ini, dan terutama rasa terima kasih kepada Bapak Andri Nofiar.Am, S.Kom, M.Kom. selaku dosen pengampu mata kuliah Multimedia.

Dan penulisan laporan ini penulis menyadari sepenuhnya bahwa masih banyak terdapat kekurangan dalam penulisan Laporan ini. Untuk itu kritik dan saran yang sifatnya membangun sangat penulis harapkan demi kesempurnaan laporan ini.

Bangkinang, 4 Agustus 2020

Penulis

DAFTAR ISI

KATA PENGANTAR.....	i
DAFTAR ISI	ii
DAFTAR GAMBAR	iii
BAB I	5
TINJAUAN PUSTAKA.....	5
1.1 Pengertian.....	5
1.2 Tujuan Praktikum	6
1.1 Alat dan Bahan	6
BAB II.....	7
PEMBAHASAN	7
2.1 Langkah-langkah	7
2.2 Hasil Pembahasan.....	9
2.2.1 Design	9
2.2.2 Source Code.....	14
2.2.3 Database	28
BAB III.....	29
PENUTUP	29
3.1 Kesimpulan.....	29
3.2 Saran	29
DAFTAR PUSTAKA	iv

DAFTAR GAMBAR

Gambar 2.1 Unduh Program.....	7
Gambar 2.2 Guipraktek.....	7
Gambar 2.3 Import Database.....	8
Gambar 2.4 Buka NetBeans.....	8
Gambar 2.5 Desain Login.....	9
Gambar 2.6 Desain Selamat Datang.....	10
Gambar 2.7 Desain Dashboard.....	10
Gambar 2.8 Desain Penjualan.....	11
Gambar 2.9 Desain Daftar Barang.....	11
Gambar 2.10 Desain Pegawai.....	12
Gambar 2.11 Desain Rubah Pegawai.....	12
Gambar 2.12 Desain Tambah Pegawai.....	13
Gambar 2.13 Source Code Login 1.....	14
Gambar 2.14 Source Code Login 2.....	14
Gambar 2.15 Source Code Login 3.....	15
Gambar 2.16 Source Code Login 4.....	15
Gambar 2.17 Source Code Selamat Datang	16
Gambar 2.18 Source Code Dashboard 1.....	16
Gambar 2.19 Source Code Dashboard 2.....	17
Gambar 2.20 Source Code Penjualan 1.....	17
Gambar 2.21 Source Code Penjualan 2.....	18
Gambar 2.22 Dashboard 1.....	19
Gambar 2.23 Dashboard 2.....	20
Gambar 2.24 Dashboard 3.....	20
Gambar 2.25 Dashboard 4.....	21
Gambar 2.26 Dashboard 5.....	21
Gambar 2.27 Dashboard 6.....	21
Gambar 2.28 Pegawai 1.....	22
Gambar 2.29 Pegawai 2.....	23
Gambar 2.30 Pegawai 3.....	23
Gambar 2.31 Pegawai 4.....	24
Gambar 2.32 Rubah Pegawai 1.....	25

Gambar 2.33 Rubah Pegawai 2.....	26
Gambar 2.34 Rubah Pegawai 3.....	26
Gambar 2.35 Rubah Pegawai 4.....	26
Gambar 2.36 Tambah Pegawai 1.....	27
Gambar 2.37 Tambah Pegawai 2.....	27
Gambar 2.38 Tambah Pegawai 3.....	28
Gambar 2.39 Database.....	28

BAB I

TINJAUAN PUSTAKA

1.1 Pengertian

a. Java

Menurut Bambang Haryanto, *Java* merupakan bahasa berorientasi objek untuk pengembangan aplikasi mandiri, aplikasi berbasis *internet*, aplikasi untuk perangkat cerdas yang dapat berkomunikasi lewat *internet*/jaringan komunikasi. (Bambang Haryanto, 2011:2, Esensi- esensi Bahasa Pemograman *Java*. Yogyakarta:Andi)

b. Neatbeans

Menurut materibelajar.co.id *Neatbeans* merupakan sebuah aplikasi *Intergrated Development Environment (IDE)* berbasis Java dari *Sun Microsystem* yang berjalan di atas *swing*. *Swing* disini adalah sebuah teknologi Java untuk pengembangan aplikasi *desktop* yang dapat berjalan di *Windows*, *Linux*, *Mac OS X* dan juga *Solaris*.

c. Jframe

Menurut fangirlh wang, *Jframe* merupakan turunan dan pengembangan dari *class Frame*. Secara teknis *class Jframe* berbeda dengan meskipun ada garis Hirearki pada keduanya. Jika pada *Frame* kita bisa memasukkan sejumlah komponen secara langsung, maka pada *Jframe* kita harus memasukkan komponen-komponen tersebut melalui *container* dari *Jframe*. (fangirlhwang.wordpress.com).

d. Source Code

Menurut technopark, Surakarta.go.id, *Source code* merupakan komponen dasar sebuah program *computer* yang diciptakan oleh seorang *programmer*. Komponen ini bisa dibaca dan dipahami dengan mudah oleh manusia.

e. GUI

GUI merupakan antarmuka pada sistem operasi komputer yang menggunakan menu grafis. Menu grafis ini maksudnya terdapat tampilan yang lebih ditekankan untuk membuat sistem operasi yang *user-friendly* agar para pengguna lebih nyaman menggunakan komputer. Menu grafis itu ya seperti ada grafis-grafis atau gambar-gambar dan tampilan yang tujuannya untuk memudahkan para pengguna menggunakan sistem operasi.

1.2 Tujuan Praktikum

- a. Mengetahui aplikasi untuk membuat Program Polkam Mart.
- b. Mengetahui fungsi dasar dalam membangun fungsi- fungsi dalam pembuatan Program Polkam Mart.
- c. Mengetahui fungsi dari *Tool* yang terdapat pada Netbeans.
- d. Mengetahui *source code* untuk pembuatan Program Polkam Mart.
- e. Agar dapat membuat Program Polkam Mart di Aplikasi NetBeans.

1.1 Alat dan Bahan

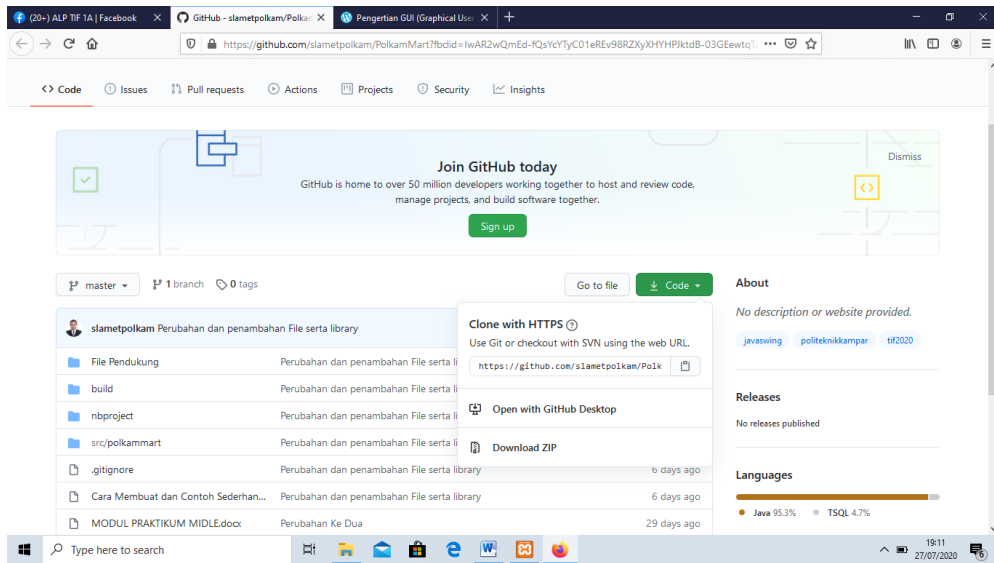
- a. PC/Laptop
- b. Aplikasi Neatbeans 8.0.2
- c. Jdk
- d. Xampp

BAB II

PEMBAHASAN

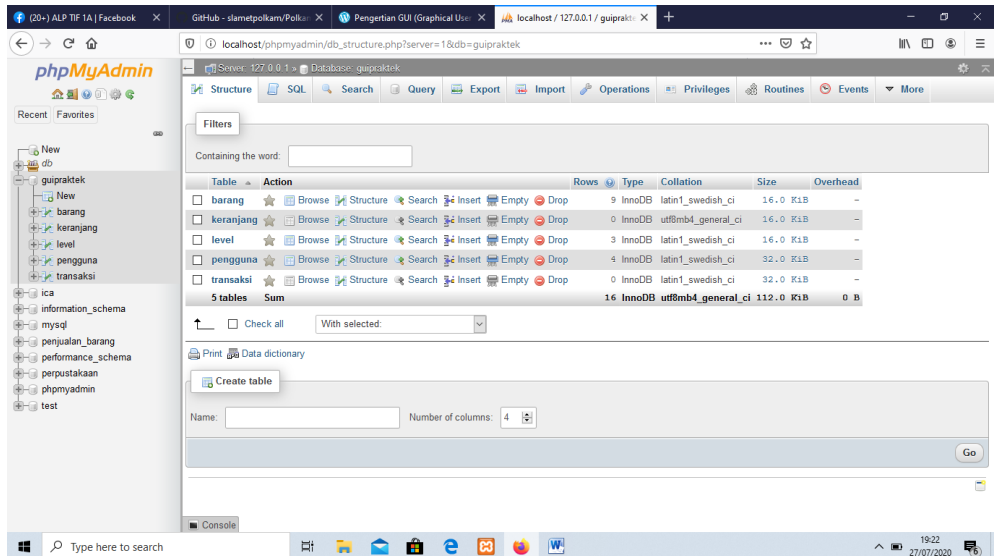
2.1 Langkah-langkah

1. Unduh program yang telah dikirim di GitHub



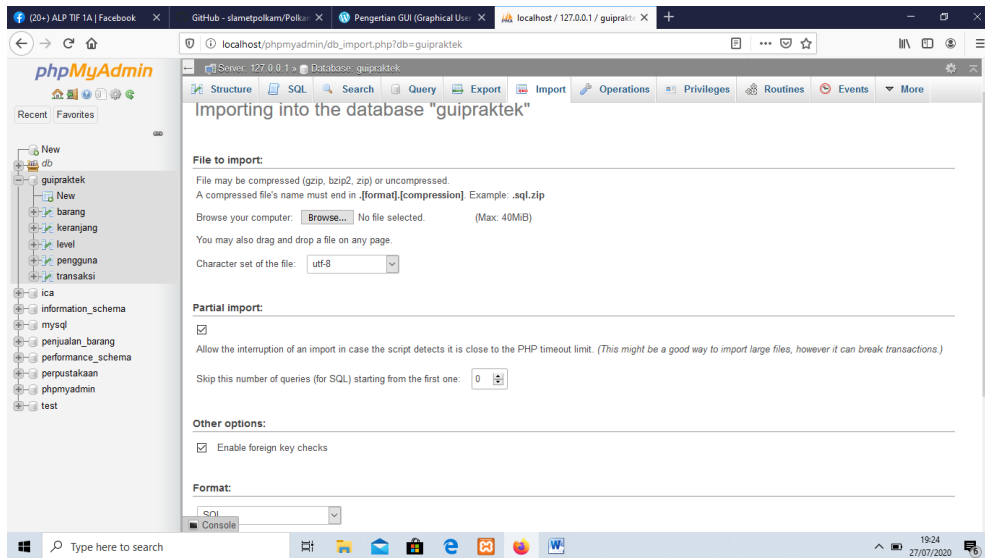
Gambar 2.1 Unduh Program

2. Importkan database polkam mart ke mysql dengan nama database guipraktek



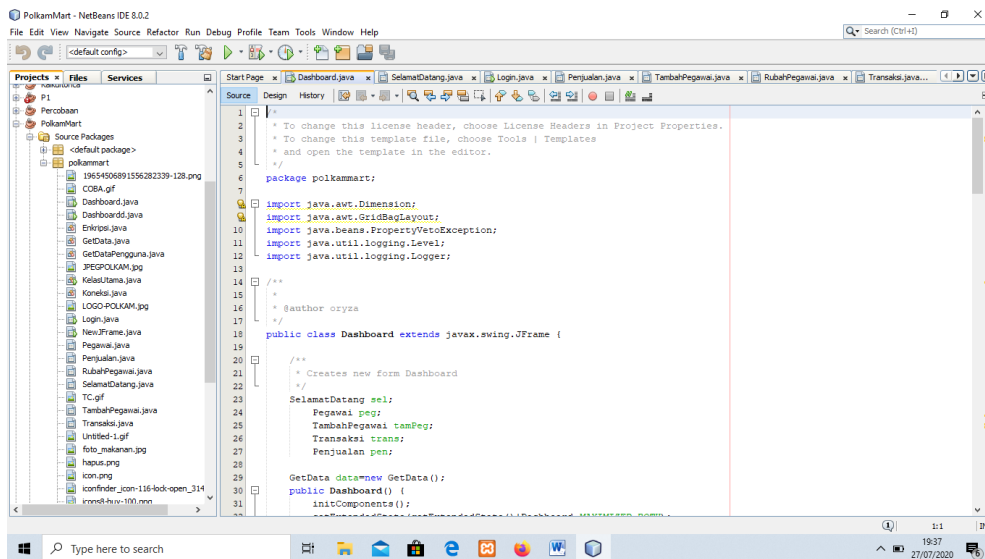
Gambar 2.2 Guipraktek

3. Lalu klik import untuk membuka program di NetBeans



Gambar 2.3 Import database

4. Setelah itu buka Netbeans, pilih program polkam mart



Gambar 2.4 Buka NetBeans

5. Lalu kembangkan program sesuai yang kamu inginkan, dalam hal Design maupun Source Code nya.

2.2 Hasil Pembahasan

2.2.1 Design

A. Login



The image shows a login form design for 'POLKAM MART'. The form has a blue header with the text 'POLKAM MART' in large, bold, black and red letters. Below the header, the text 'LOGIN PENGGUNA' is displayed. The form contains two input fields: 'Nama Pengguna' (User Name) and 'Sandi' (Password). The 'Nama Pengguna' field is preceded by a user icon, and the 'Sandi' field is preceded by a lock icon. Below the input fields is a 'LOGIN' button with a right-pointing arrow icon.

Gambar 2.5 Design Login

Penjelasan : Pada design Login saya merubah warna background, menambahkan Tulisan Polkam Mart, dan menambahkan Icon.

B. Selamat Datang



Gambar 2.6 Design Selamat Datang

Penjelasan : Pada design Selamat datang saya merubah warna background, menambahkan tulisan, dan juga gif.

C. Dashboard



Gambar 2.7 Design DashBoard

Penjelasan : Pada design dashboard saya merubah warna background dan menambahkan icon.

D. Penjualan



Gambar 2.8 Design Penjualan

Penjelasan : Pada design Penjualan, saya membuat program penjualan dan juga merubah warna background, menambahkan tulisan Penjualan, dan icon.

E. Daftar Barang



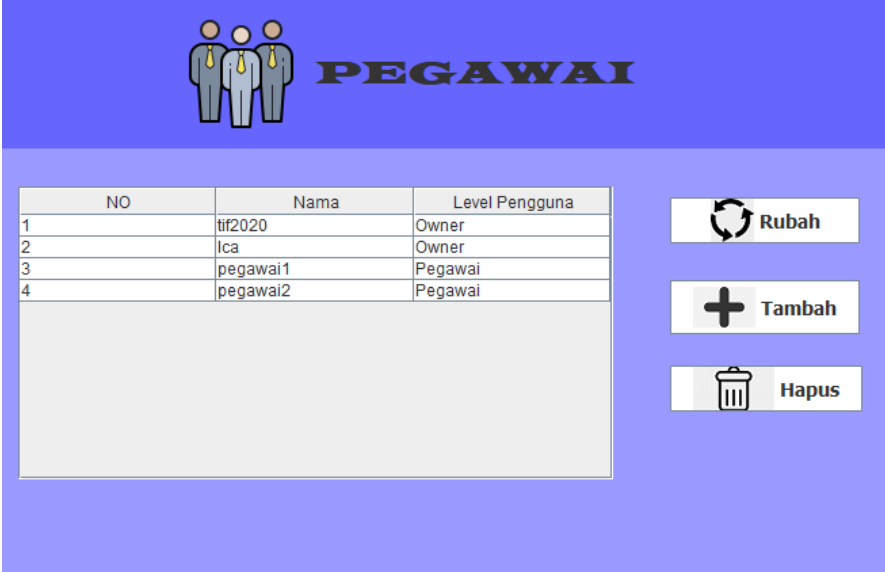
NO	Barcode	Nama Barang	Harga Satuan	Stok
1	000001	Ajnomote	1000	12
2	000002	Aqua gelas	500	24
3	000003	Air-air	1000	24
4	000004	Biakut	2000	12
5	000005	Bonbon Langkat	500	10
6	000006	Bun Cake	5000	12
7	000007	Coconut oil	10000	12
8	000008	Dancee	5000	12
9	000009	Deiry Milk	10000	12

Gambar 2.9 Design Daftar Barang

Penjelasan : Pada design Daftar barang saya merubah warna

background, menambahkan tulisan daftar barang, dan icon.

F. Pegawai



NO	Nama	Level Pengguna
1	tif2020	Owner
2	lca	Owner
3	pegawai1	Pegawai
4	pegawai2	Pegawai

Gambar 2.10 Design Pegawai

Penjelasan : Pada design Pegawai saya merubah warna backroud, menambahkan tulisan Pegawai dan Icon.

G. Rubah Pegawai



Gambar 2.11 Design Rubah Pegawai

Penjelasan : Pada desing Rubah pegawai saya merubah warna background, menambahkan tulisan Rubah pegawai dan icon.

H. Tambah Pegawai



The screenshot shows a web application window with a blue header and a light blue body. The header contains an icon of three people and the title 'TAMBAH PEGAWAI'. The body contains a form with the following fields:

- Nama :
- Sandi :
- Level :

Below the form are three buttons: 'Tambah', 'Kembali', and 'Upload'. To the right of the form, there is a text label 'DISINI DAPAT DITAMBAHKAN GAMBAR'.

Gambar 2.12 Design Tambah Pegawai

Penjelasan : Pada design tambah pegawai saya merubah warna background , menambahkan tulisan Tambah pegawai dan icon.

2.2.2 Source Code

A. Login

```
package polkamart;

import com.sun.glass.events.KeyEvent;
import java.sql.Connection;
import java.sql.ResultSet;
import java.sql.Statement;
import javax.swing.JOptionPane;
import sun.security.provider.MD5;

/**
 *
 * @author oryza
 */
public class Login extends javax.swing.JFrame {

    /**
     * Creates new form Login
     */
    //Membuat variabel bertipe Connection dengan nama koneksi
    //tipe variabel ini terdapat pada java.sql lihat pada bagian import java.sql
    Connection koneksi;
    //Membuat variabel bertipe Statement dengan nama stat
    //tipe variabel ini terdapat pada java.sql lihat pada bagian import java.sql
    Statement stat;
    //Membuat variabel bertipe Resultset dengan nama hasil
    //tipe variabel ini terdapat pada java.sql lihat pada bagian import java.sql
    ResultSet hasil;
    //Membuat variabel bertipe String dengan nama query
    String query;
    //membuat Instan kelas Enkripsi.java
    Enkripsi enkrip=new Enkripsi();
}
```

Gambar 2.13 Source Code Login 1

```
public Login() {
    initComponents();

    //Instan kelas koneksi
    Koneksi konek=new Koneksi();
    /*
     * memanggil fungsi konfigurasi untuk mengetes koneksi dengan database
     * dan menampilkan messege box jika koneksi berhasil maka akan tampil
     * messege box koneksi berhasil dan lanjut menampilkan form login
     * jika gagal maka muncul messege box dan keluar dari system
     */

    if (konek.konfigurasi().equals("Gagal Koneksi")){
        JOptionPane.showMessageDialog(this,konek.konfigurasi());
        System.exit(0);
    }else{
        //memanggil variabel statemen yang ada pada kelas Koneksi.java dengan instan yang telah dibuat
        stat=konek.statemen;
    }
}

/**
 * This method is called from within the constructor to initialize the form.
 * WARNING: Do NOT modify this code. The content of this method is always
 * regenerated by the Form Editor.
 */
@SuppressWarnings("unchecked")
```

Gambar 2.14 Source Code Login 2

```

private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    //variabel username dan password akan mengambil String yang di masukan melalui form
    String username=jTextField1.getText();
    //enkripsi passord yang diperoleh dari jLabel2 ke MD5 lewat kelas Enkripsi
    String password=enkrip.encryptPasswordMD5(jPasswordField1.getText());
    //mencoba melakukan query kedalam database yang telah terhubung pada konstruktor
    try{
        /*
         query untuk menampilkan semua data yang terdapat pada
         tabel pengguna database yang terkoneksi
         pada kelas Koneksi.java
        */
        query="SELECT * FROM pengguna WHERE nama ='"+username+"'AND sandi='"+password+"'";
        //proses eksekusi query
        hasil=stat.executeQuery(query);
        //jika hasilnya true query tersebut dan memberikan return tidak null maka lanjut
        //jika tidak maka akan leuar messege box dan kembali ke form login
        if(hasil.next()){
            GetData data=new GetData();
            GetData.setUserLogin(username);
            if(username.equals(hasil.getString("nama"))&&password.equals(hasil.getString("sandi"))){
                Dashboard dashboard= new Dashboard();
                dashboard.setVisible(true);
                this.setVisible(false);
            }
        }else{
            JOptionPane.showMessageDialog(this,
            "Maaf Anda Tidak Diizinkan Masuk Sistem"+password,
            "Kesalahn Pengguna dan Sandi",
            JOptionPane.WARNING_MESSAGE);
        }
    }
}

```

Gambar 2.15 Source Code Login 3

```

    }catch(Exception e){
        System.out.println(" not okee"+e.getMessage());
    }
}

private void jTextField1ActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
}

/**
 * @param args the command line arguments
 */
public static void main(String args[]) {
    /* Set the Nimbus look and feel */
    Look and feel setting code (optional)

    /* Create and display the form */
    java.awt.EventQueue.invokeLater(new Runnable() {
        public void run() {
            new Login().setVisible(true);
        }
    });
}

```

Gambar 2.16 Source Code Login 4

B. Selamat Datang

```
package polkammart;

import java.beans.PropertyVetoException;
import java.util.logging.Level;
import java.util.logging.Logger;

/**
 *
 * @author oryza
 */
public class SelamatDatang extends javax.swing.JInternalFrame {

    /**
     * Creates new form SelamatDatang
     */
    public SelamatDatang() {
        initComponents();
        jLabel1.setText("SELAMAT DATANG : "+GetData.getUserLogin());
    }
}
```

Gambar 2.17 Source Code Selamat Datang

C. Dashboard

```
package polkammart;

import java.awt.Dimension;
import java.awt.GridBagLayout;
import java.beans.PropertyVetoException;
import java.util.logging.Level;
import java.util.logging.Logger;

/**
 *
 * @author oryza
 */
public class Dashboard extends javax.swing.JFrame {

    /**
     * Creates new form Dashboard
     */
    SelamatDatang sel;
    Pegawai peg;
    TambahPegawai tamPeg;
    Transaksi trans;
    Penjualan pen;

    GetData data=new GetData();
    public Dashboard() {
        initComponents();
        setExtendedState(getExtendedState() | Dashboard.MAXIMIZED_BOTH);
    }
}
```

Gambar 2.18 Source Code Dashboard 1

```

        sel=new SelamatDatang();
        peg=new Pegawai();
        trans=new Transaksi();
        tamPeg=new TambahPegawai();
        pen=new Penjualan();
        DesktopPane.add(tamPeg);
        DesktopPane.add(peg);
        DesktopPane.add(sel);
        DesktopPane.add(trans);
        DesktopPane.add(pen);

        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 2.19 Source Code Dashboard 2

Penjelasan : Pada dashboard saya menambahkan Penjualan

D. Penjualan

```

package polkammart;

import java.sql.Connection;
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.JOptionPane;
import javax.swing.table.DefaultTableModel;

/**
 *
 * @author oryza
 */
public class Penjualan extends javax.swing.JInternalFrame {
    // private DefaultTableModel model;
    // public Statement st;
    // public ResultSet rs;
    // Connection cn = polkammart.Koneksi.getKoneksi();
    //

    public String hrgbrng = "";

    public double n1, n2, n3, jumlah;
    double m1;

    public Penjualan() {
        initComponents();
    }

```

Gambar 2.20 Source Code Penjualan 1

```
private void totalActionPerformed(java.awt.event.ActionEvent evt) {  
    // TODO add your handling code here:  
  
    n1 = Double.parseDouble(text1.getText());  
    n2 = Double.parseDouble(text2.getText());  
    n3 = Double.parseDouble(text4.getText());  
    m1 = n1 * n2 ;  
    jumlah = n3 -m1 ;  
  
    text3.setText(Double.toString(m1));  
    text5.setText(Double.toString(jumlah));  
  
}  
  
private void keluarActionPerformed(java.awt.event.ActionEvent evt) {  
    System.exit(0);  
}  
  
private void batalActionPerformed(java.awt.event.ActionEvent evt) {  
    text5.setText("");  
}  
  
private void text4ActionPerformed(java.awt.event.ActionEvent evt) {  
    // TODO add your handling code here:  
}  
  
private void text3ActionPerformed(java.awt.event.ActionEvent evt) {  
    text3.setText(Double.toString(m1));  
}
```

Gambar 2.21 Source Code Penjualan 2

Penjelasan : Dalam Penjualan saya membuat fungsi untuk pembelian, dan menambahkan hitung yang mana untuk jumlah bayar dan kembaliannya menjadi otomatis

E. Daftar Barang

```
package polkamart;

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.JOptionPane;
import javax.swing.table.DefaultTableModel;

/**
 *
 * @author oryza
 */
public class Transaksi extends javax.swing.JInternalFrame {

    /**
     * Creates new form Transaksi
     */
    private DefaultTableModel model;
    Object[] obj;
    public Transaksi() {
        initComponents();

        AmbilData();
    }
}
```

Gambar 2.22 Source Code Daftar barang1

```

public void AmbilData() {

    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 table tblGaji dari database
        Statement stat = (Statement) Koneksi.getKoneksi().createStatement();

        String sql = "Select * from barang";
        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++;
        }
    }
}

```

Gambar 2.23 Source Code Daftar barang 2

```

private void jButton3ActionPerformed(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();
    } catch (SQLException ex) {
        Logger.getLogger(TambahPegawai.class.getName()).log(Level.SEVERE, null, ex);
    }
}

```

Gambar 2.24 Source Code Daftar barang 3

```

private void ubahActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    int column1 = 1;
    if(ubah.getText()=="Ambil Data"){
        try{
            int selectedRowIndex=tabelBarang.getSelectedRow();

            if(selectedRowIndex<-1){
                JOptionPane.showMessageDialog(null, "Silahkan Pilih Data Terlebih Dahulu");
            }else{
                String[]value = new String[5];
                for(int i=1;i<value.length;i++){
                    String values= tabelBarang.getModel().getValueAt(selectedRowIndex, i).toString();
                    value[i]=values;
                }
                barcode.setText(value[1]);
                namabarang.setText(value[2]);
                harga.setText(value[3]);
                stok.setText(value[4]);
                ubah.setText("Ubah Data");
            }
        }catch (Exception e) {
            JOptionPane.showMessageDialog(null, e);
        }
    }else if(ubah.getText()=="Ubah Data"){
        String z=barcode.getText();
        String y=namabarang.getText();
        String l=harga.getText();
        String j=stok.getText();
        try {
            Statement stat = (Statement) Koneksi.getKoneksi().createStatement();

```

Gambar 2.25 Source Code Daftar barang 4

```

try {
    Statement stat = (Statement) Koneksi.getKoneksi().createStatement();
    String sql = "UPDATE barang SET barcode='"+z+"',nama_barang='"+y+"',harga='"+l+"',stok='"+j+"',keterangan='' WHERE barcode='"+z+"'";
    stat.executeUpdate(sql);
    System.out.println(sql);
    JOptionPane.showMessageDialog(this,"Perubahan Data Berhasil ");
    barcode.setText("");
    namabarang.setText("");
    harga.setText("");
    stok.setText("");
    AmbilData();
} catch (SQLException ex) {
    Logger.getLogger(TambahPegawai.class.getName()).log(Level.SEVERE, null, ex);
}

```

Gambar 2.26 Source Code Daftar barang 5

```

private void jButton4ActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    try{
        int selectedRowIndex=tabelBarang.getSelectedRow();

        if(selectedRowIndex<-1){
            JOptionPane.showMessageDialog(null, "Silahkan Pilih Data Terlebih Dahulu");
        }else{
            String[]value = new String[5];
            for(int i=1;i<value.length;i++){
                String values= tabelBarang.getModel().getValueAt(selectedRowIndex, i).toString();
                value[i]=values;
            }

            String query="Delete from barang where barcode='"+value[1]+"';

            //membuat statemen pemanggilan data pada table tblGaji dari database
            Statement stat = (Statement) Koneksi.getKoneksi().createStatement();
            stat.executeUpdate(query);
            AmbilData();
        }
    }catch (Exception e) {
        JOptionPane.showMessageDialog(null, e);
    }
}

```

Gambar 2.27 Source Code Daftar barang 6

F. Pegawai

```
package polkammart;

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.JOptionPane;
import javax.swing.table.DefaultTableModel;

/**
 *
 * @author oryza
 */
public class Pegawai extends javax.swing.JInternalFrame {

    /**
     * Creates new form Pegawai
     */
    private DefaultTableModel model;
    Object[ ] obj;
    public Pegawai() {
        initComponents();
        jPanell.setSize(50*Pegawai.HEIGHT/100,50);

        AmbilData();
        this.setTitle("Data Karyawan");
    }
}
```

Gambar 2.28 Source Code Pegawai 1

```

public void AmbilData() {
    model=new DefaultTableModel();
    jTable2.setModel(model);
    model.addColumn("NO");
    model.addColumn("Nama");
    model.addColumn("Level Pengguna");

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

        String sql = "Select * from pengguna, level where pengguna.level=level.no";
        ResultSet res = stat.executeQuery(sql);
        //penelusuran baris pada tabel tblGaji dari database
        while(res.next()){
            obj= new Object[3];

            obj[0] =i;
            obj[1] = res.getString("nama");
            obj[2] = res.getString("nama_lv1");

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

```

Gambar 2.29 Source Code Pegawai 2

```

private void tambahActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    TambahPegawai tamb=new TambahPegawai();
    this.getParent().add(tamb);
    tamb.setVisible(true);
    tamb.setSize(this.getWidth(),this.getHeight());

    tamb.setLocation(0,0);
    this.setVisible(false);
}

private void hapusActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    try{
        int column = 1;
        int selectedRowIndex=jTable2.getSelectedRow();
        String value = jTable2.getModel().getValueAt(selectedRowIndex, column).toString();
        JOptionPane.showMessageDialog(null, value);
        String query="Delete from pengguna where nama='"+value+"'";
        JOptionPane.showMessageDialog(null, value);
        //membuat statemen pemanggilan data pada table tblGaji dari database
        Statement stat = (Statement) Koneksi.getKoneksi().createStatement();
        stat.executeUpdate(query);
        AmbilData();
    }catch(Exception e){
        JOptionPane.showMessageDialog(null, e);
    }
}

```

Gambar 2.30 Source Code Pegawai 3


```

private void rubahActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    int column = 1;

    try{
        int selectedRowIndex=jTable2.getSelectedRow();

        if(selectedRowIndex<-1){
            JOptionPane.showMessageDialog(null, "Silahkan Pilih Data Terlebih Dahulu");
        }else{
            String value = jTable2.getModel().getValueAt(selectedRowIndex, column).toString();
            GetDataPengguna.setUserLogin(value);
            RubahPegawai rbh=new RubahPegawai();
            this.getParent().add(rbh);
            rbh.setVisible(true);
            rbh.setSize(this.getWidth(),this.getHeight());
            rbh.setLocation(0,0);
            this.setVisible(false);
        }
    }catch (Exception e) {
        JOptionPane.showMessageDialog(null, e);
    }
}

public void refresh(){
    this.AmbilData();
}

```

Gambar 2.31 Source Code Pegawai 4

Penjelasan : Dalam tabel pegawai saya menambahkan pilihan level, yaitu kasir dan owner.

G. Rubah Pegawai

```

package polkammart;

import java.awt.Image;
import java.io.ByteArrayOutputStream;
import java.io.File;
import java.lang.invoke.MethodHandle;
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;
import java.io.FileInputStream;

/**
 *
 * @author oryza
 */
public class RubahPegawai extends javax.swing.JInternalFrame {

    /**
     * Creates new form TambahPegawai
     */
    String id_pengguna="";
    String namafile = null;

    public RubahPegawai() {
        initComponents();
        this.setTitle("Rubah Data Pegawai");
        nama.disable();
        try{
            //membuat statemen pemanggilan data pada table tblGaji dari database
            Statement stat = (Statement) Koneksi.getKoneksi().createStatement();

            String sql = "Select * from pengguna where nama='"+GetDataPengguna.getUserLogin()+"'";
            ResultSet res = stat.executeQuery(sql);
            //penelusuran baris pada tabel tblGaji dari database
            while(res.next()){

```

Gambar 2.32 Source Code Rubah pegawai 1

```

private void imageActionPerformed(java.awt.event.ActionEvent evt) {
    JFileChooser chooser = new JFileChooser();
    chooser.showOpenDialog(null);
    File f = chooser.getSelectedFile();
    namafile = f.getAbsolutePath();
    ImageIcon imageIcon = new ImageIcon(new ImageIcon(namafile).getImage().getScaledInstance(lbl_img.getWidth(), lbl_img.getHeight(), Image.SCALE_SMOOTH));
    lbl_img.setIcon(imageIcon);
    try{
        File image = new File(namafile);
        FileInputStream fis = new FileInputStream(image);
        ByteArrayOutputStream bos = new ByteArrayOutputStream();
        byte[] buf = new byte[1024];
        for (int readNum; (readNum=fis.read(buf))!=-1;){
            bos.write(buf,0,readNum);
        }
    } catch (Exception e){
        JOptionPane.showMessageDialog(null, e);
    }
}

```

```

        while(res.next()){
            nama.setText(res.getString("nama"));
            int lvl=Integer.parseInt(res.getString("level"));
            tingkat.setSelectedIndex(lvl-1);
            id_pengguna=Integer.toString(Integer.parseInt(res.getString("id"))+1);
        }
    }catch(SQLException err){
        JOptionPane.showMessageDialog(null, err.getMessage());
    }
}

```

Gambar 2.33 Source Code Rubah pegawai 2

```

private void tambahActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    Enkripsi s=new Enkripsi();
    String z=nama.getText();
    String y=s.encryptPasswordMD5(sandi.getText());
    String l=Integer.toString(tingkat.getSelectedIndex()+1);
    System.out.println("Nama : "+z+" Sandi : "+y+" "+l);
    Pegawai peg=new Pegawai();
    try {
        //membuat statemen pemanggilan data pada table tblGaji dari database
        Statement stat = (Statement) Koneksi.getKoneksi().createStatement();
        String query="Select * from pengguna, level where pengguna.level=level.no";
        ResultSet res = stat.executeQuery(query);

    }

    String sql = "UPDATE pengguna SET sandi='"+y+"',level='"+l+"' WHERE nama='"+z+"'";
    stat.executeUpdate(sql);
    JOptionPane.showMessageDialog(this,"Perubahan Data Berhasil ");

    peg.AmbilData();
    this.getParent().removeAll();
    this.getParent().add(peg);
    peg.setVisible(true);
    peg.setSize(this.getWidth(),this.getHeight());
    peg.setLocation(0,0);
    this.dispose();

} catch (SQLException ex) {
    Logger.getLogger(RubahPegawai.class.getName()).log(Level.SEVERE, null, ex);
}
}

```

Gambar 2.34 Source Code Rubah pegawai 3

```

private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    Pegawai peg=new Pegawai();
    peg.AmbilData();
    this.getParent().add(peg);
    peg.setVisible(true);
    peg.setSize(this.getWidth(),this.getHeight());
    peg.setLocation(0,0);
    this.setVisible(false);
}

```

Gambar 2.35 Source Code Rubah pegawai 4

H. Tambah Pegawai

```
package polkammart;

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.JOptionPane;

/**
 *
 * @author oryza
 */
public class TambahPegawai extends javax.swing.JInternalFrame {

    /**
     * Creates new form TambahPegawai
     */

    public TambahPegawai() {
        initComponents();
        this.setTitle("Tambah Data Pegawai");
    }
}
```

Gambar 2.36 Source Code Tambah pegawai 1

```
private void tambahActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    Enkripsi s=new Enkripsi();
    String z=nama.getText();
    String y=s.encryptPasswordMD5(sandi.getText());
    String l=Integer.toString(level.getSelectedIndex()+1);
    System.out.println("Nama : "+z+" Sandi : "+y+" "+l);
    Pegawai peg=new Pegawai();
    try {
        //membuat statemen pemanggilan data pada table tblGaji dari database
        Statement stat = (Statement) Koneksi.getKoneksi().createStatement();
        String query="Select * from pengguna, level where pengguna.level=level.no";
        ResultSet res = stat.executeQuery(query);
        String id_pengguna="";
        while(res.next()){
            id_pengguna=Integer.toString(Integer.parseInt(res.getString("id"))+1);
        }
        String sql = "INSERT INTO pengguna VALUES ('"+id_pengguna+"', '"+z+"', '"+y+"', '"+l+"')";
        stat.executeUpdate(sql);
        JOptionPane.showMessageDialog(this,"Penambahan Data Berhasil ");

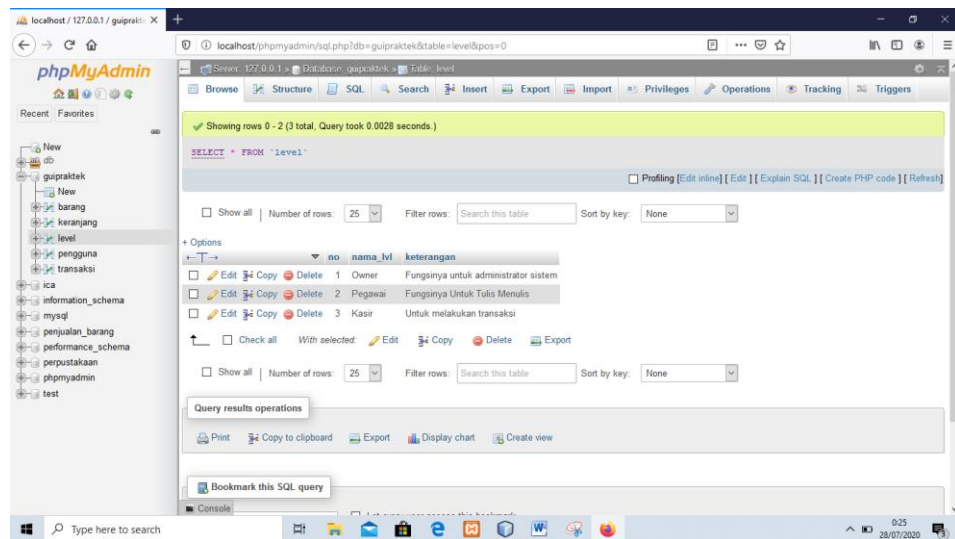
        peg.AmbilData();
        this.getParent().add(peg);
        peg.setVisible(true);
        peg.setSize(this.getWidth(),this.getHeight());
        peg.setLocation(0,0);
        this.setVisible(false);
    } catch (SQLException ex) {
        Logger.getLogger(TambahPegawai.class.getName()).log(Level.SEVERE, null, ex);
    }
}
```

Gambar 2.37 Source Code Tambah pegawai 2

```
private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    Pegawai peg=new Pegawai();
    peg.AmbilData();
    this.getParent().add(peg);
    peg.setVisible(true);
    peg.setSize(this.getWidth(),this.getHeight());
    peg.setLocation(0,0);
    this.setVisible(false);
}
}
```

Gambar 2.38 Source Code Tambah pegawai 3

2.1 Database



Gambar 2.39 Database

Penjelasan : Pada tabel level saya menambahkan level owner dan kasir.

BAB III

PENUTUP

3.1 Kesimpulan

Kesimpulan mengenai laporan praktikum ini ialah dimana bahasa Java sebagai bahasa pemrograman yang berkembang pasti pada saat ini memiliki peranan penting, bukan hanya untuk *user* yang merasakan manfaat lebih dengan menggunakan aplikasi bahasa pemrograman Java, tetapi pemrograman juga dapat berkarya dan meningkatkan nilai estetika yang baik dan lebih baik, yang bagus, dan lebih bagus, yang efisien dan lebih efisien dengan mengandalkan fitur-fitur yang berada didalamnya untuk mengoperasikan pemrograman bahasa Java.

Dan dengan pembuatan Program Polkam Mart ini diharapkan bisa menambah kreatifitas agar aplikasi yang selanjutnya akan dibuat dapat meningkatkan daya saing dibidang komputer maupun pemasaran.

Program Polkam Mart ini dibuat untuk memudahkan proses pembelian dan penyimpanan catatan pembelian, agar lebih memudahkan kasir.

Pada *design* program ini saya menggunakan, Label, JButton, JTextField, ComboBox, Menu item, dan juga table.

3.2 Saran

Saran penulis mengenai pembuatan Program Polkam Mart menggunakan Java NetBeans ini agar dapat dikembangkan lagi, dalam hal *design* dan juga dalam programnya, agar lebih baik lagi.

DAFTAR PUSTAKA

<http://fangirlhwang.wordpress.com>

<http://materibelajar.co.id>

<http://m-risa.blogspot.com>

<http://id.scribd.com>

<http://www.academia.edu>

<http://www.technopark,Surakarta.go.id>