

MENGEMBANGKAN APLIKASI DESKTOP DENGAN JAVA



Disusun Oleh

Nama : Wirdatul Jannah

Nim : 201913003

Dosen pengampu

SLAMET TRIYANTO

PROGRAM STUDI TEKNIK INFORMATIKA

POLITEKNIK KAMPAR

2020

KATA PENGANTAR

Assalamualaikum wr.wb

Segala puji bagi kehadiran Allah swt, karena dengan rahmat dan hidayahnya penulis bisa menyelesaikan laporan ini dengan tepat waktu. Sholawat serta salam semoga terlimpah curahkan kepada baginda tercinta kita yaitu Nabi Muhammad SAW yang kita nantikan syafa'atnya di akhir nanti.

Atas limpahan nikmat sehatnya sehingga penulis mampu untuk menyelesaikan pembuatan proposal sebagai nilai pratikum satu dari mata kuliah PBO (Pemrograman Berbasis Objek) dengan judul “Membuat Aplikasi Login Kantin PolkamMart Menggunakan Neatbens”.

Penulis tentu menyadari bahwa laporan ini masih jauh dari kata sempurna dan masih banyak terdapat kesalahan serta kekurangan di dalamnya. Untuk itu, penulis mengharapkan kritik serta saran dari pembaca. Kemudian apabila terdapat banyak kesalahan pada laporan ini penulis mohon maaf yang sebesar-besarnya.

Penulis juga mengucapkan terimakasih kepada semua pihak khususnya kepada Bapak Slamet Triyanto yang telah membimbing pelajaran PBO.

Demikian, semoga proposal ini dapat bermanfaat. Terima kasih.

Bangkinang, 28 Juli 2020

Penulis

BAB I

PENDAHULUAN

1. LATAR BELAKANG

Java adalah bahasa pemrograman yang dapat dijalankan di berbagai komputer termasuk telepon genggam. 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.

Neatbeans adalah pengembangan suatu aplikasi dapat dilakukan dimulai dari suatu setelan perangkat lunak modular bernama modules.

JDK (Java Development Kit) adalah perangkat lunak yang digunakan untuk melakukan proses kompilasi dari kode java ke bytecode yang dapat dimengerti dan dapat di jalankan oleh JRE (Java Runtime Envirotment).

2. TINJAUAN PUTAKA/REFERENSI

- a. Membuat kalkulator sederhana menggunakan Neatbeans bahasa java.
- b. Mempelajari Neatbeans agar memudahkan membuat program.
- c. Agar dapat memahami apa itu neatbeans dan java.

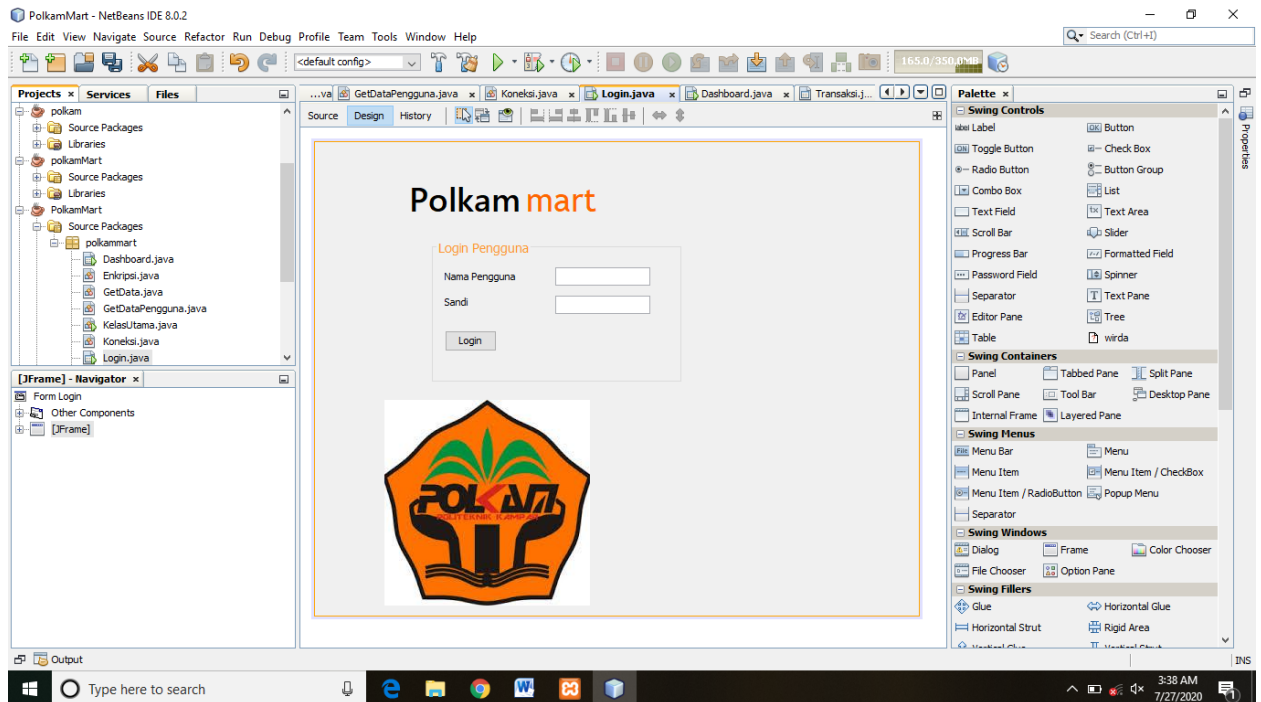
3. ALAT DAN BAHAN

- a. Alat
 - komputer atau laptop.
- b. Bahan
 - Neatbeans

BAB II

PEMBAHASAN

1. ini merupakan codingan dan desain kantin PolkamMart



```
6 package polkammart;
7
8 import com.sun.glass.events.KeyEvent;
9 import java.sql.Connection;
10 import java.sql.ResultSet;
11 import java.sql.Statement;
12 import javax.swing.JOptionPane;
13 import sun.security.provider.MD5;
14
15 /**
16  *
17  * @author oryza
18  */
19 public class Login extends javax.swing.JFrame {
20
```

```

26 Connection koneksi;
27 //Membuat variabel bertipe Statement deng
28 //tipe variabel ini terdapat pada java.sc
29 Statement stat;
30 //Membuat variabel bertipe Resultset deng
31 //tipe variabel ini terdapat pada java.sc
32 ResultSet hasil;
33 //Membuat variabel bertipe String dengan
34 String query;
35 //membuat Instan kelas Enkripsi.java
36 Enkripsi enkrip=new Enkripsi();
37
38 public Login() {
39     initComponents();
40
41     //Instan kelas koneksi
42     Koneksi konek=new Koneksi();

```

```

50         if (konek.konfigurasi().equals("Gagal Koneksi")){
51             JOptionPane.showMessageDialog(this,konek.konfigurasi());
52             System.exit(0);
53         }else{
54             //memanggil variabel statemen yang ada pada kelas Koneksi.java
55             stat=konek.statemen;
56         }

```

```

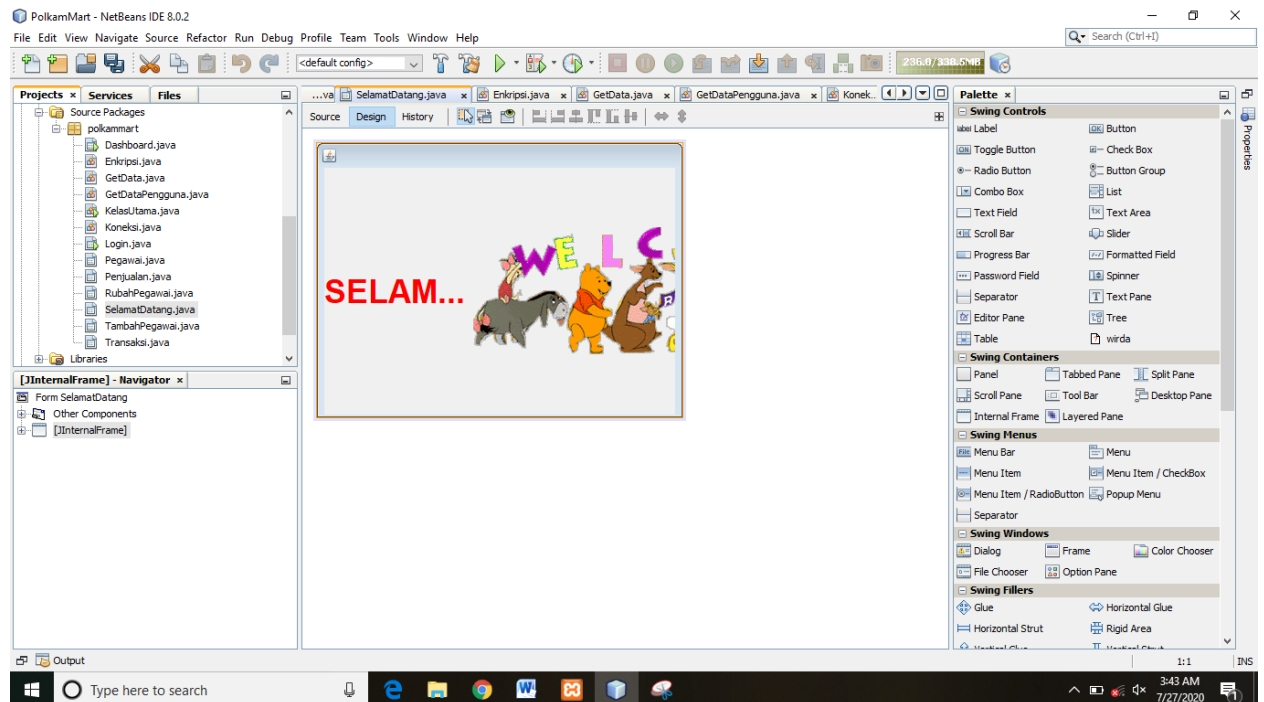
214         try{
215             /*
216              query untuk menampilkan semua data yang terdapat pada
217              tabel pengguna database yang terkoneksi
218              pada kelas Koneksi.java
219             */
220             query="SELECT * FROM pengguna WHERE nama =' "+username+"'AND sandi='"+password+"'";
221             //proses eksekusi query
222             hasil=stat.executeQuery(query);
223             //jika hasilnya true query tersebut dan memberikan return tidak null maka lanjut
224             //jika tidak maka akan leuar messege box dan kembali ke form login
225             if(hasil.next()){
226                 //
227                 GetData data=new GetData();
228                 GetData.setUserLogin(username);
229                 //
230                 if(username.equals(hasil.getString("nama"))&&password.equals(hasil.getString("sandi"))){
231                     Dashboard dashboard= new Dashboard();
232                     dashboard.setVisible(true);
233                     this.setVisible(false);
234                 }
235             }else{
236                 JOptionPane.showMessageDialog(this,
237                 "Maaf Anda Tidak Diizinkan Masuk Sistem"+password,
238                 "Kesalahn Pengguna dan Sandi",
239                 JOptionPane.WARNING_MESSAGE);

```

```

242         }catch(Exception e){
243             System.out.println(" not okee"+e.getMessage());
244         }
245     }
246 }
247
248 private void jTextField1ActionPerformed(java.awt.event.ActionEvent evt) {
249     // TODO add your handling code here:
250
251 }
252
253
254 /**
255  * @param args the command line arguments
256  */
257 public static void main(String args[]) {
258     /* Set the Nimbus look and feel */
259     Look and feel setting code (optional)
260
261     /* Create and display the form */
262     java.awt.EventQueue.invokeLater(new Runnable() {
263         public void run() {
264             new Login().setVisible(true);
265         }
266     });

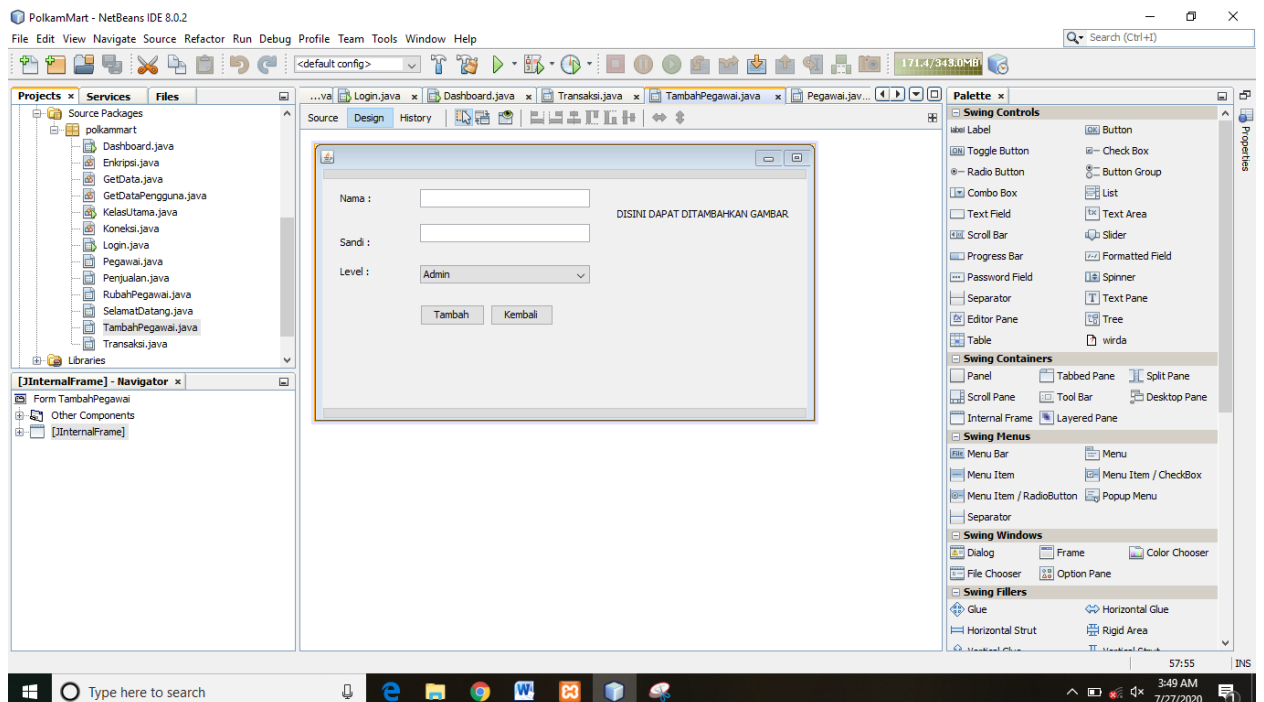
```



```

16 public class SelamatDatang extends javax.swing.JInternalFrame {
17
18     /**
19      * Creates new form SelamatDatang
20      */
21     public SelamatDatang() {
22         initComponents();
23         jLabel1.setText("SELAMAT DATANG "+GetData.getUserLogin());
24     }
25
26     /**
27      * This method is called from within the constructor to initialize the form.
28      * WARNING: Do NOT modify this code. The content of this method is always
29      * regenerated by the Form Editor.
30      */
31     @SuppressWarnings("unchecked")
32     Generated Code
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54 // Variables declaration - do not modify
55 private javax.swing.JLabel jLabel1;
56 private javax.swing.JLabel jLabel2;
57 // End of variables declaration
58 }
59

```



```

19 public class TambahPegawai extends javax.swing.JInternalFrame {
20
21     /**
22      * Creates new form TambahPegawai
23      */
24
25     public TambahPegawai() {
26         initComponents();
27         this.setTitle("Tambah Data Pegawai");
28     }
29
30     /**
31      * This method is called from within the constructor to initialize the form.
32      * WARNING: Do NOT modify this code. The content of this method is always
33      * regenerated by the Form Editor.
34      */
35     @SuppressWarnings("unchecked")
36     // Generated Code

```

```

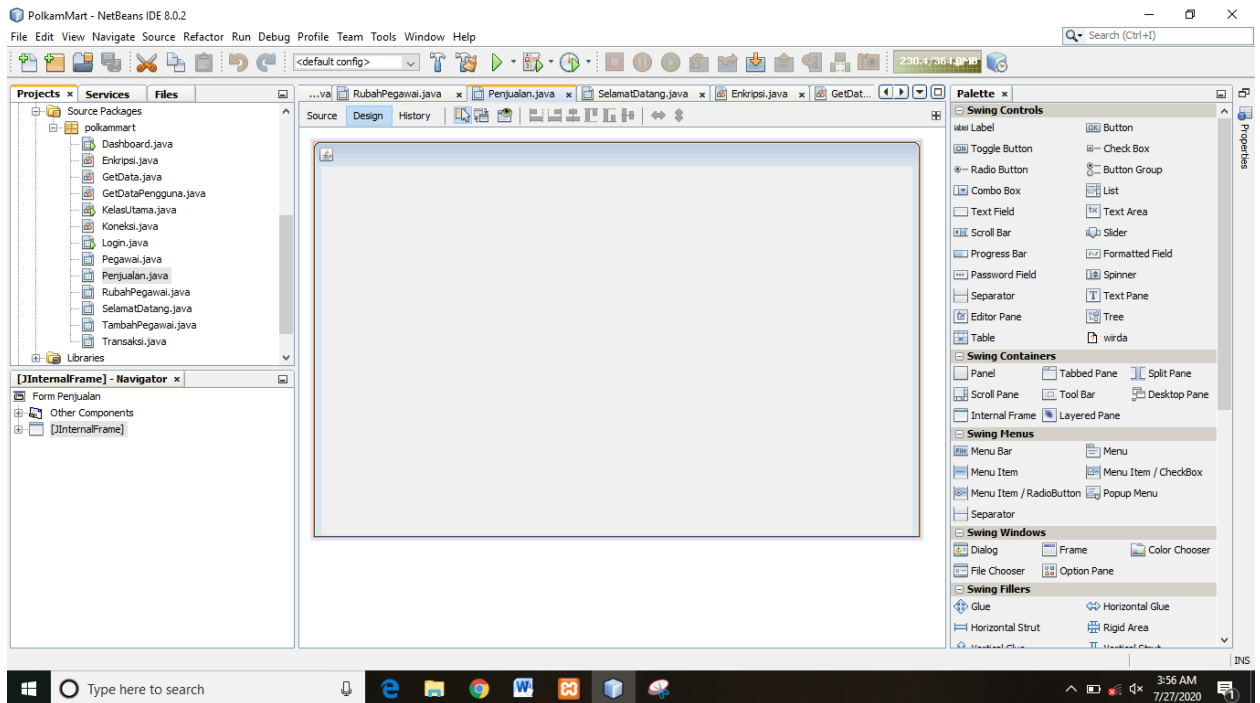
190 private void tambahActionPerformed(java.awt.event.ActionEvent evt) {
191     // TODO add your handling code here:
192     Enkripsi s=new Enkripsi();
193     String z=nama.getText();
194     String y=s.encryptPasswordMD5(sandi.getText());
195     String l=Integer.toString(level.getSelectedIndex()+1);
196     System.out.println("Nama : "+z+" Sandi : "+y+" "+l);
197     Pegawai peg=new Pegawai();
198     try {
199         //membuat statemen pemanggilan data pada table tblGaji dari database
200         Statement stat = (Statement) Koneksi.getKoneksi().createStatement();
201         String query="Select * from pengguna, level where pengguna.level=level.no";
202         ResultSet res = stat.executeQuery(query);
203         String id_pengguna="";
204         while(res.next()){
205             id_pengguna=Integer.toString(Integer.parseInt(res.getString("id"))+1);
206         }
207         String sql = "INSERT INTO pengguna VALUES ('"+id_pengguna+"', '"+z+"', '"+y+"', '"+l+"')";
208         stat.executeUpdate(sql);
209         JOptionPane.showMessageDialog(this,"Penambahan Data Berhasil ");
210
211         peg.AmbilData();
212         this.getParent().add(peg);
213         peg.setVisible(true);
214         peg.setSize(this.getWidth(),this.getHeight());
215         peg.setLocation(0,0);
216         this.setVisible(false);

```

```

224 private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
225     // TODO add your handling code here:
226     Pegawai peg=new Pegawai();
227     peg.AmbilData();
228     this.getParent().add(peg);
229     peg.setVisible(true);
230     peg.setSize(this.getWidth(),this.getHeight());
231     peg.setLocation(0,0);
232     this.setVisible(false);
233 }

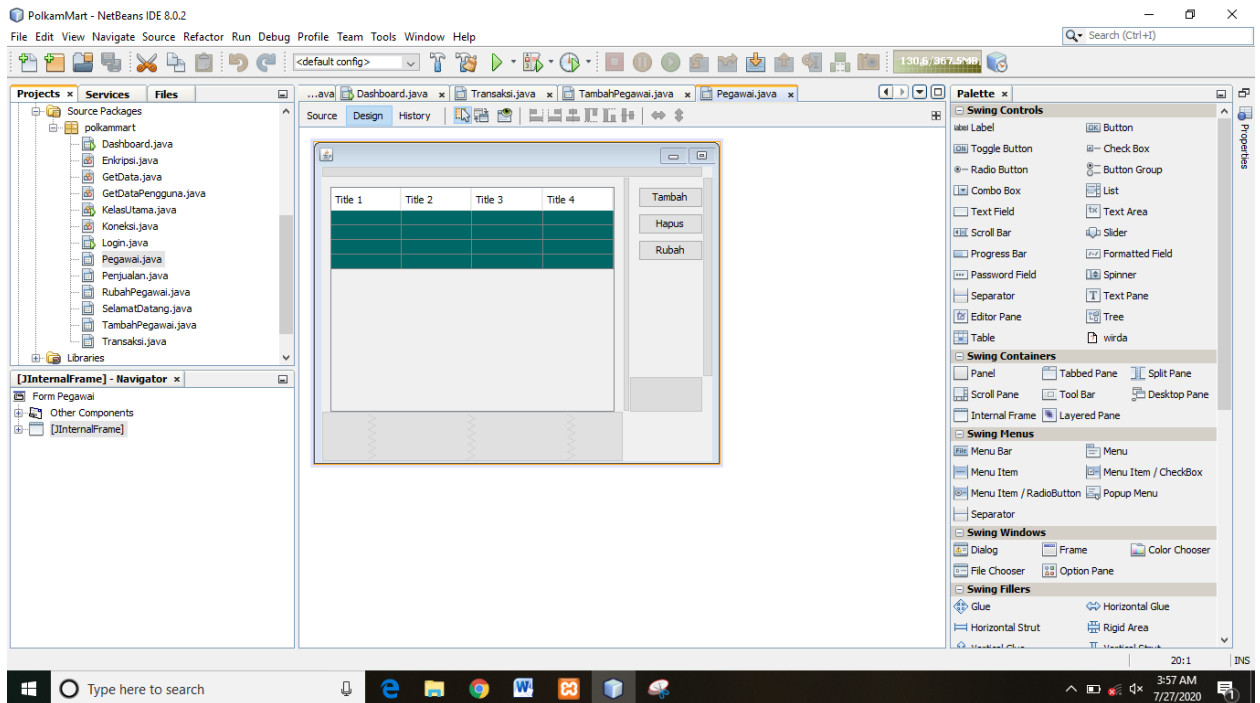
```

```

6      package polkammart;
7
8      /**
9       *
10      * @author oryza
11      */
12     public class Penjualan extends javax.swing.JInternalFrame {
13
14         /**
15          * Creates new form Penjualan
16          */
17         public Penjualan() {
18             initComponents();
19         }

```



```

20 public class Pegawai extends javax.swing.JInternalFrame {
21
22     /**
23      * Creates new form Pegawai
24      */
25     private DefaultTableModel model;
26     Object[ ] obj;
27     public Pegawai() {
28         initComponents();
29         jPanel1.setSize(50 * Pegawai.HEIGHT / 100, 50);
30
31         AmbilData();
32         this.setTitle("Data Karyawan");
33     }
34
35     public void AmbilData() {
36
37         model = new DefaultTableModel();
38         jTable2.setModel(model);
39         model.addColumn("NO");
40         model.addColumn("Nama");
41         model.addColumn("Level Pengguna");
42
43         model.getDataVector().removeAllElements();
44         model.fireTableDataChanged();
45         int i = 1;
46         try{

```

```

50     String sql = "Select * from pengguna, level where pengguna.level=level.no";
51     ResultSet res = stat.executeQuery(sql);
52     //penelusuran baris pada tabel tblGaji dari database
53     while(res.next()){
54         obj= new Object[3];
55
56         obj[0] =i;
57         obj[1] = res.getString("nama");
58         obj[2] = res.getString("nama_lvl");
59
60         model.addRow(obj);
61         i++;
62     }
63 }catch(SQLException err){
64     JOptionPane.showMessageDialog(null, err.getMessage() );

```

```

191 private void tambahActionPerformed(java.awt.event.ActionEvent evt) {
192     // TODO add your handling code here:
193     TambahPegawai tamb=new TambahPegawai();
194     this.getParent().add(tamb);
195     tamb.setVisible(true);
196     tamb.setSize(this.getWidth(),this.getHeight());
197
198     tamb.setLocation(0,0);
199     this.setVisible(false);
200 }

```

```

202 private void hapusActionPerformed(java.awt.event.ActionEvent evt) {
203     // TODO add your handling code here:
204     try{
205         int column = 1;
206         int selectedRowIndex=jTable2.getSelectedRow();
207         String value = jTable2.getModel().getValueAt(selectedRowIndex, column).toString();
208         JOptionPane.showMessageDialog(null, value);
209         String query="Delete from pengguna where nama='"+value+"'";
210         // JOptionPane.showMessageDialog(null, value);
211         //membuat statemen pemanggilan data pada table tblGaji dari database
212         Statement stat = (Statement) Koneksi.getKoneksi().createStatement();
213         stat.executeUpdate(query);
214         AmbilData(1,2,3,4,5);
215     }catch(Exception e){
216         JOptionPane.showMessageDialog(null, e);
217     }
218 }

```

```

220 private void rubahActionPerformed(java.awt.event.ActionEvent evt) {
221     // TODO add your handling code here:
222     int column = 1;
223
224     try{
225         int selectedRowIndex=jTable2.getSelectedRow();
226
227         if(selectedRowIndex<-1){
228             JOptionPane.showMessageDialog(null, "Silahkan Pilih Data Terlebih Dahulu");
229         }else{
230             String value = jTable2.getModel().getValueAt(selectedRowIndex, column).toString();
231             GetDataPengguna.setUserLogin(value);
232             RubahPegawai rbh=new RubahPegawai();
233             this.getParent().add(rbh);
234             rbh.setVisible(true);
235             rbh.setSize(this.getWidth(),this.getHeight());
236             rbh.setLocation(0,0);
237             this.setVisible(false);
238         }
239     }catch (Exception e) {
240         JOptionPane.showMessageDialog(null, e);
241     }
242
243 }

```

```

245 public void refresh(){
246     this.AmbilData();
247 }
248
249 // Variables declaration - do not modify
250 private javax.swing.JButton hapus;
251 private javax.swing.JPanel jPanel1;
252 private javax.swing.JPanel jPanel2;
253 private javax.swing.JScrollPane jScrollPane2;
254 private javax.swing.JTable jTable2;
255 private javax.swing.JButton rubah;
256 private javax.swing.JButton tambah;
257 // End of variables declaration
258
259 private void AmbilData(int i, int i0, int i1, int i2, int i3) {
260     throw new UnsupportedOperationException("Not supported yet.");
261 }
262
263 }

```

- Koneksi

```

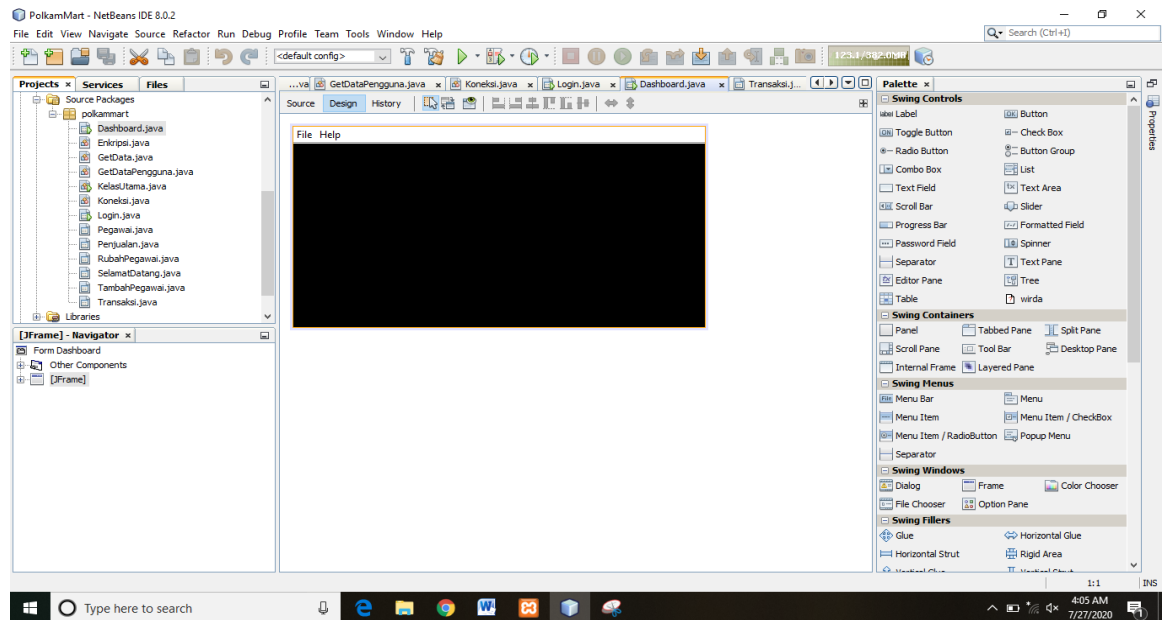
22 public String konfigurasi() {
23     String Message;
24     try{
25         Class.forName("com.mysql.jdbc.Driver");
26         connect=DriverManager.getConnection("jdbc:mysql://localhost/guipraktek","guipraktek","guipraktek");
27         statemen=connect.createStatement();
28
29     }catch(Exception e){
30         System.out.println("gagal koneksi"+ e);
31         return Messege="Gagal Koneksi";
32     }
33     return Messege="Koneksi Basis Data Berhasil";
34 }

```

```

39 public static Connection getKoneksi( ){
40     String host = "jdbc:mysql://localhost/guipraktek",
41     user = "guipraktek",
42     pass = "guipraktek";
43     try{
44         connect = (Connection) DriverManager.getConnection(host, user, pass);
45     }catch (SQLException err){
46         JOptionPane.showMessageDialog(null, err.getMessage( ) );
47     }
48     return connect;
49 }
50 }

```



```

19 public class Dashboard extends javax.swing.JFrame {
20
21     /**
22      * Creates new form Dashboard
23      */
24     SelamatDatang sel;
25     Pegawai peg;
26     TambahPegawai tamPeg;
27     Transaksi trans;
28     GetData data=new GetData();
29     public Dashboard() {
30         initComponents();
31         setExtendedState(getExtendedState() | Dashboard.MAXIMIZED_BOTH);
32
33         sel=new SelamatDatang();
34         peg=new Pegawai();
35         trans=new Transaksi();
36         tamPeg=new TambahPegawai();
37         DesktopPane.add(tamPeg);
38         DesktopPane.add(peg);
39         DesktopPane.add(sel);
40         DesktopPane.add(trans);
41
42         DesktopPane.moveToFront(sel);
43         sel.setVisible(true);
44         sel.setSize(DesktopPane.getWidth(),DesktopPane.getHeight());
45         try {
46             sel.setMaximum(true);
47         } catch (PropertyVetoException ex) {
48             Logger.getLogger(SelamatDatang.class.getName()).log(Level.SEVERE, null, ex);
49         }
50         sel.setLocation(0,0);
51
52     private void jMenuItem1ActionPerformed(java.awt.event.ActionEvent evt) {
53         // TODO add your handling code here:
54
55         sel.setVisible(false);
56         peg.setVisible(true);
57         peg.setSize(DesktopPane.getWidth(),DesktopPane.getHeight());
58         try {
59             peg.setMaximum(true);
60         } catch (PropertyVetoException ex) {
61             Logger.getLogger(Pegawai.class.getName()).log(Level.SEVERE, null, ex);
62         }
63         peg.setLocation(0,0);

```

```

177 private void jMenuItem2ActionPerformed(java.awt.event.ActionEvent evt) {
178     // TODO add your handling code here:
179     sel.setVisible(false);
180     peg.setVisible(false);
181     trans.setVisible(true);
182     trans.setSize(DesktopPane.getWidth(),DesktopPane.getHeight());
183     try {
184         trans.setMaximum(true);
185     } catch (PropertyVetoException ex) {
186         Logger.getLogger(Pegawai.class.getName()).log(Level.SEVERE, null, ex);
187     }
188     trans.setLocation(0,0);
189 }

```

```

191 private void jMenuItem3ActionPerformed(java.awt.event.ActionEvent evt) {
192
193     JOptionPane.showMessageDialog(this, "Anda Akan Keluar Dari System");
194     System.exit(0); // TODO add your handling code here:
195 }

```

```

196
197 /**
198  * @param args the command line arguments
199  */
200 public static void main(String args[]) {
201     /* Set the Nimbus look and feel */
202     Look and feel setting code (optional)
203
204     /* Create and display the form */
205     java.awt.EventQueue.invokeLater(new Runnable() {
206         public void run() {
207             new Dashboard().setVisible(true);
208         }
209     });
210 }

```

```

233 private javax.swing.JDesktopPane DesktopPane;
234 private javax.swing.JMenu jMenu1;
235 private javax.swing.JMenu jMenu2;
236 private javax.swing.JMenu jMenu3;
237 private javax.swing.JMenu jMenu5;
238 private javax.swing.JMenuBar jMenuBar1;
239 private javax.swing.JMenuItem jMenuItem1;
240 private javax.swing.JMenuItem jMenuItem2;
241 private javax.swing.JMenuItem jMenuItem3;
242 // End of variables declaration
243 }

```

```

15 public class Enkripsi {
16     public String encryptPasswordMD5(String pesan) {
17         byte[] bytePesan = pesan.getBytes();
18         MessageDigest md = null;
19         String encPass;
20         try{
21             md=MessageDigest.getInstance("MD5");
22
23         }
24         catch(Exception e){
25             e.printStackTrace();
26         }
27         byte[] hasil = md.digest(bytePesan);
28         return encPass=byteArrToString(hasil);
29     }

31     private static String byteArrToString(byte[] b) {
32
33         StringBuilder sb = new StringBuilder(b.length * 2);
34         for (int i = 0; i < b.length; i++) {
35             int j = b[i] & 0xff;
36             if (j < 16) {
37                 sb.append('0');
38             }
39             sb.append(Integer.toHexString(j));
40         }
41         String res = sb.toString();
42         return res.toUpperCase();
43     }
44 }

12 public class GetData {
13     private static String username;
14
15     public static void setUserLogin(String username) {
16         GetData.username = username;
17     }
18
19
20     public static String getUserLogin(){
21         return username;
22     }
23 }

```



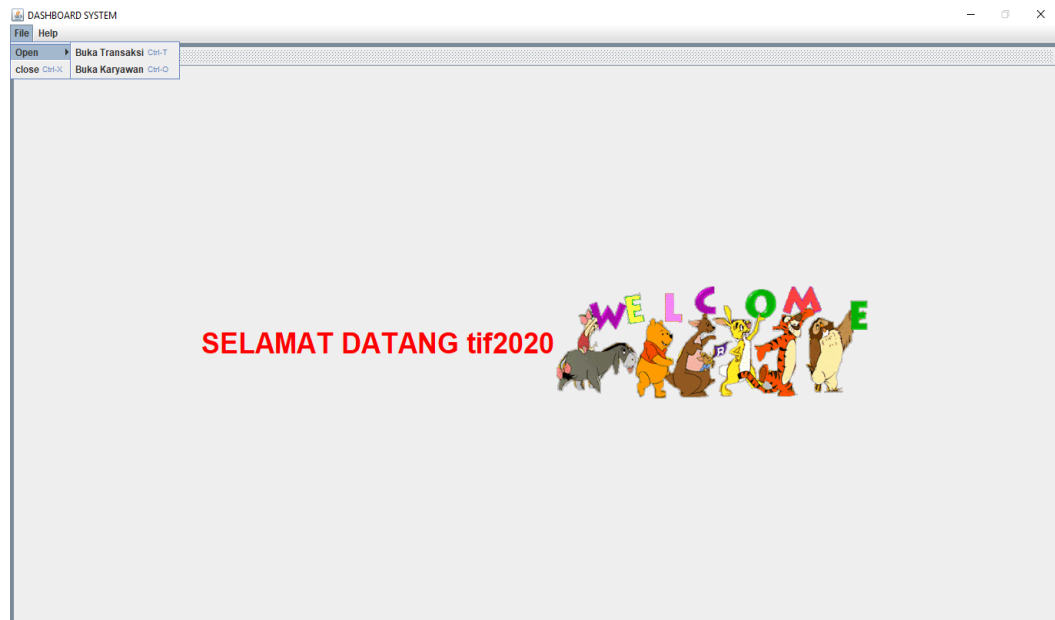
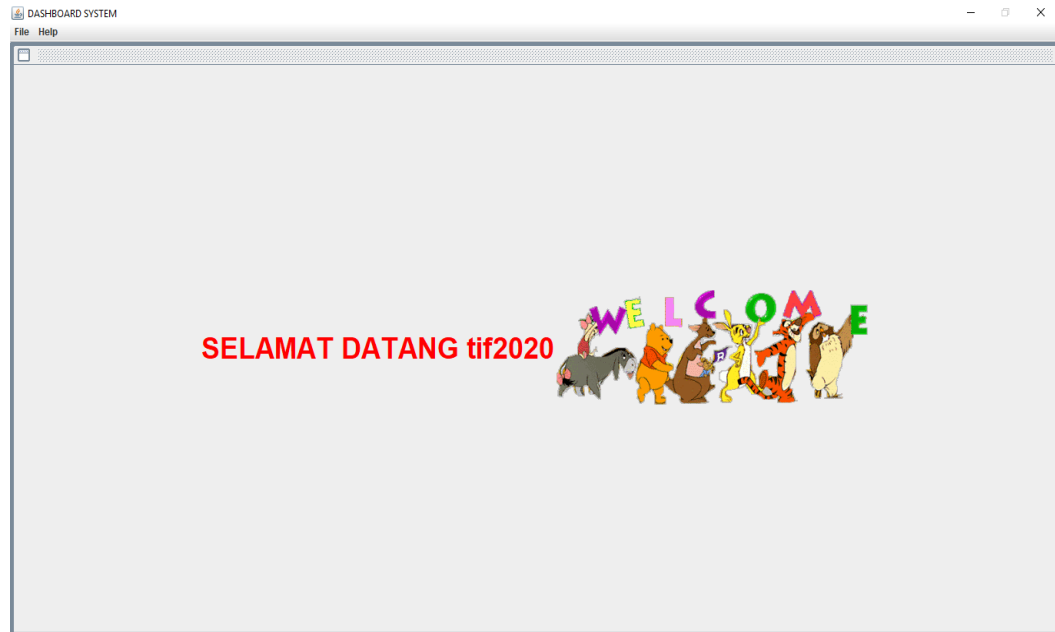
```

12     public class GetDataPengguna {
13         private static String username;
14
15         public static void setUserLogin(String username) {
16             GetDataPengguna.username = username;
17         }
18
19
20         public static String getUserLogin() {
21             return username;
22         }
23     }
24
25     public class KelasUtama {
26
27         /**
28          * @param args the command line arguments
29          */
30         public static void main(String[] args) {
31             // TODO code application logic here
32             Login masuk=new Login();
33             masuk.setVisible(true);
34         }
35     }

```

2. Ini merupakan hasil keseluruhan





DASHBOARD SYSTEM

File Help

Barcode

Nama Barang

Harga Barang

Stok Barang

Ambil Data

Tambah Stok

Hapus Barang

	NO	Barcode	Nama Barang	Harga Satuan	Stok
1		101921	Indomie	5000	40
2		111a	Pop Mie	5000	100
3		12345	photo copy	100	1000

DASHBOARD SYSTEM

File Help

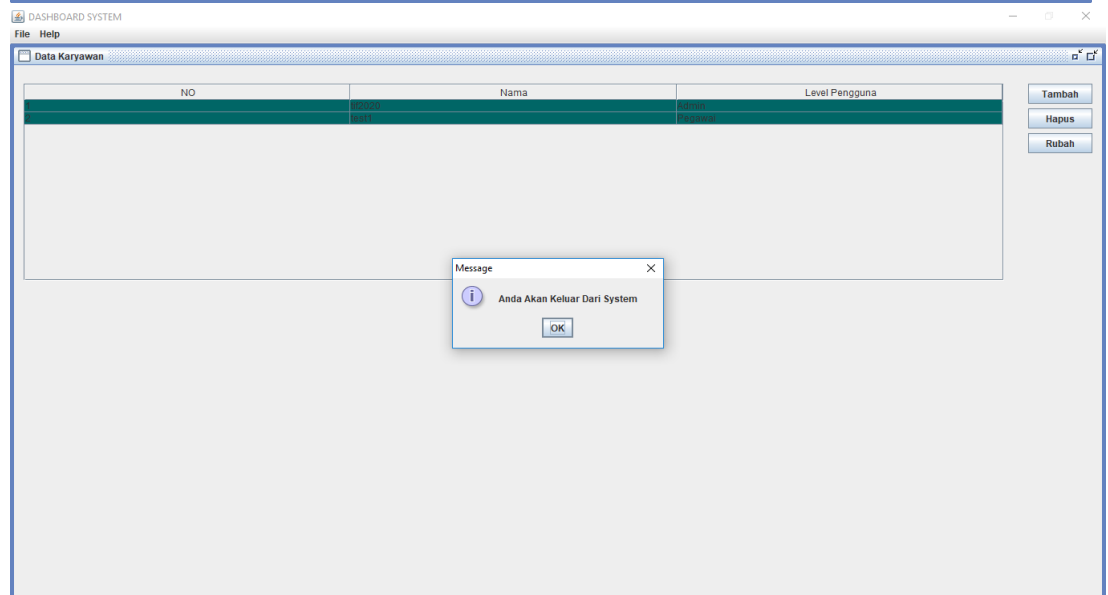
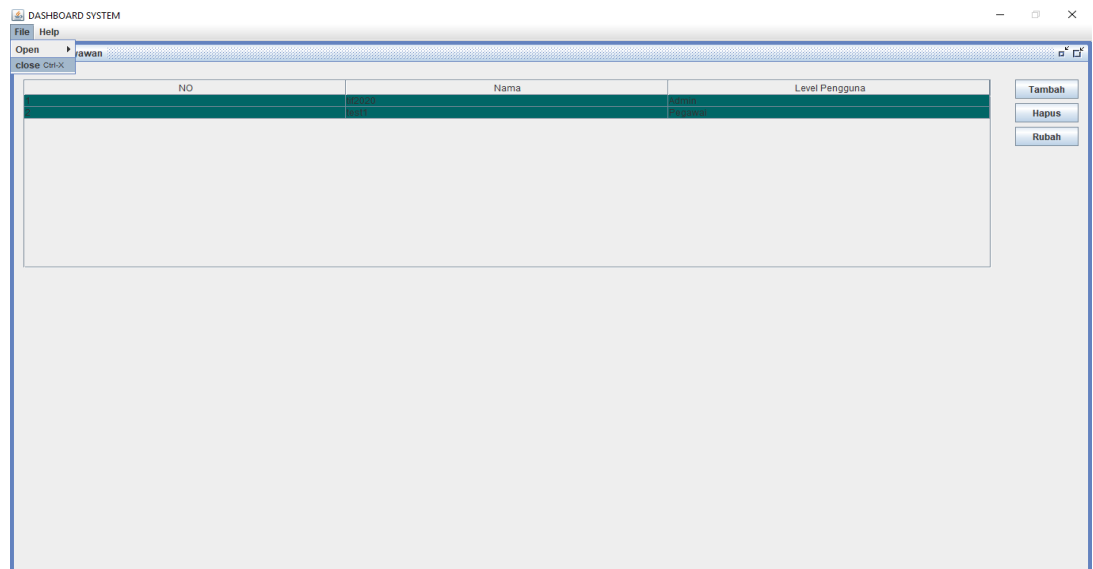
Data Karyawan

NO	Nama	Level Pengguna
1	Admin	Admin
2	Karyawan	Karyawan

Tambah

Hapus

Rubah



BAB III

PENUTUP

1. KESIMPULAN

Dari beberapa bahasan mengenai bahasa java disimpulkan bahwa java merupakan suatu teknologi yang mencakup sebagai bahasa pemrograman tersendiri, juga mencakup sebagai bahasa pemograman tersendiri platform yaitu di mana teknologi ini memiliki virtual machine dan library yang di perlukan untuk menulis dan menjalankan program yang ditulis dangan bahasa pemograman java. Tapi disini bahasa java yang digunakan menggunakan aplikasi neatbeans.

2. SARAN

Dalam pembuatan laporan ini, saya merasa ada banyak kekurangan baik itu dari segi penulisan, tata letak penyajian maupun kosa kata yang mungkin agak kurang di mengerti. Kekurangan atau sesuatu kejanggalan saya mohon maaf,dan disini saya minta untuk diberi saran atau masukan yang membangun dan bermanfaat dari pembaca.