

BAB IV

IMPLEMENTASI DAN PENGUJIAN

4.1 Spesifikasi

4.1.1 Spesifikasi Perangkat Keras (*Hardware*)

Hardware atau perangkat keras digunakan untuk menunjang atau membantu dalam pengolahan data, yang diharapkan akan mempermudah pengerjaan dan mengolah informasi, sehingga informasi dan data dapat dipenuhi dengan cepat kepada berbagai pihak yang membutuhkan.

Untuk memenuhi standar spesifikasi hardware pembuatan sistem ini maka diperlukan spesifikasi dari hardware sebagai berikut :

1.Asus

- a. Processor : Intel®Atom™ N570 (1.66GHZ, 1MB L2 cache
- b. Hardisk : 320. GB
- c. Memory RAM: 1 GB
- d. Mouse dan Keyboard

4.1.2 Spesifikasi Perangkat Lunak (*Software*)

Untuk mendapatkan kerja sistem yang baik, selain perangkat keras (*hardware*) dibutuhkan juga perangkat lunak (*software*). Perangkat lunak (*software*) yang digunakan dalam pembuatan aplikasi ini adalah sebagai berikut :

- 1. Sistem Operasi Windows 8 (64 bit)
- 2. Microsoft Office 2007
- 3. Notepad++
- 4. Database MySQL

4.2 Implementasi

Tahap Implementasi merupakan tahap lanjutan dari perancangan system yang akan dilakukan jika system disetujui, termaksud program yang telah dibuat agar siap dioperasikan secara optimal sesuai dengan kebutuhan antara lain adalah dengan menerapkan perancangan antarmuka ke dalam bentuk halaman utama, beserta ruang lingkup aplikasi yang akan digunakan dalam penerapannya.

4.2.1 Implementasi Sistem

Sistem yang telah di implementasikan akan dijelaskan pada bagian ini. Tiap rancangan yang telah dibuat sebelumnya di implementasikan kedalam sebuah system.

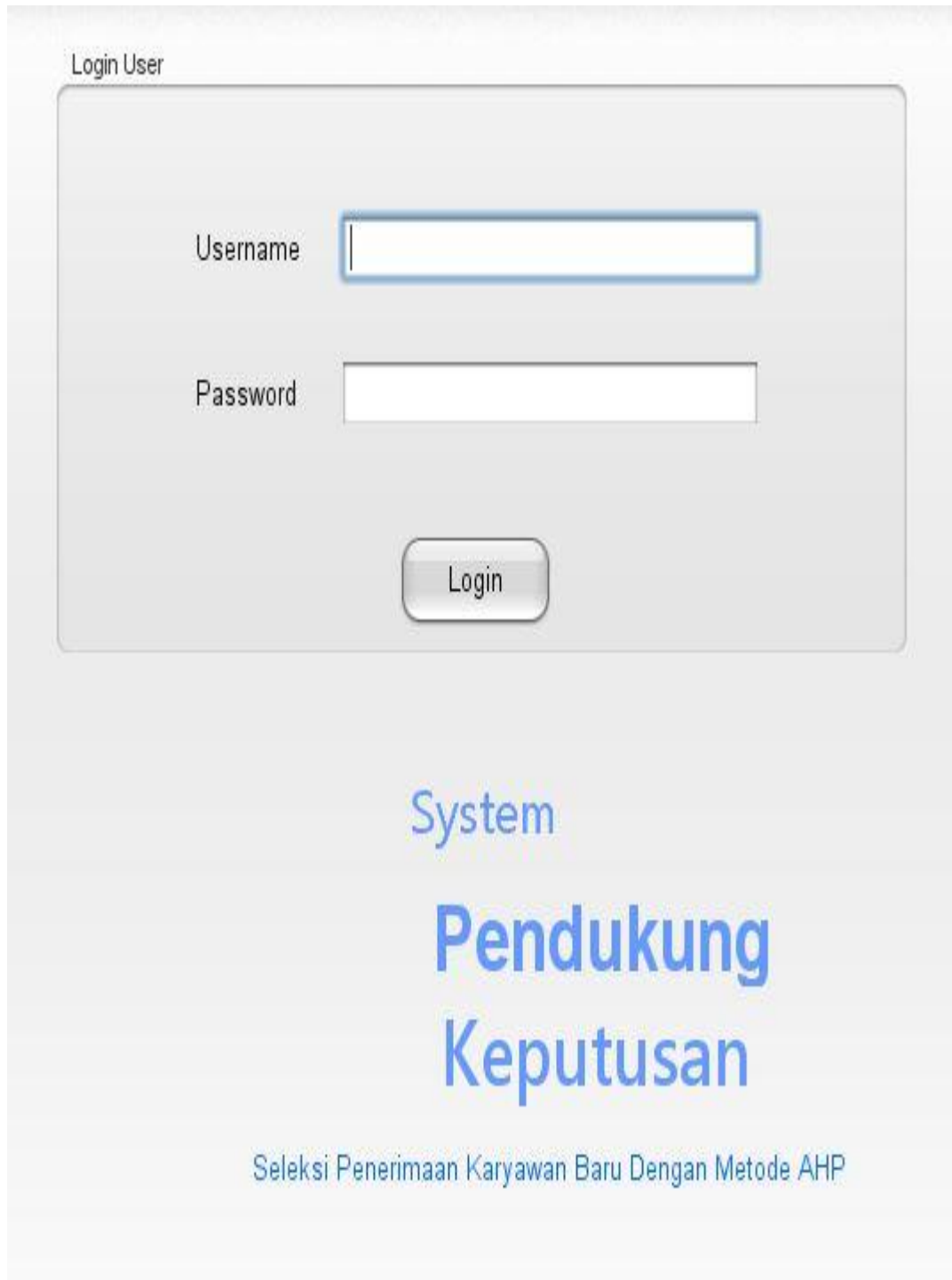
4.2.1.1 Implementasi Antara Muka

Antarmuka dilakukan dengan setiap halaman program yang dibuat dan pengkodeannya dalam bentuk file program. Berikut ini adalah implementasi antarmuka dari sistem penerimaan karyawan terbaik menggunakan metode AHP.

a. Halaman Login

Berikut ini adalah implementasi dari form *login*, dimana form ini digunakan oleh admin untuk masuk ke dalam sistem pendukung keputusan penerimaan karyawan terbaik.

Halaman ini menampilkan form login yang harus diisi oleh pengguna ke dalam sistem. Untuk membuka halaman ini pengguna dapat menjalankan aplikasi penilaian penerimaan karyawan terbaik.



The image shows a web application interface. At the top left, there is a title 'Login User'. Below it is a login form with two input fields: 'Username' and 'Password'. The 'Username' field has a blue border, while the 'Password' field has a grey border. Below the password field is a 'Login' button with a grey gradient and rounded corners. In the center of the page, the word 'System' is written in blue. Below it, the words 'Pendukung' and 'Keputusan' are written in a larger, bold blue font. At the bottom, the text 'Seleksi Penerimaan Karyawan Baru Dengan Metode AHP' is displayed in a smaller blue font.

Login User

Username

Password

Login

System

**Pendukung
Keputusan**

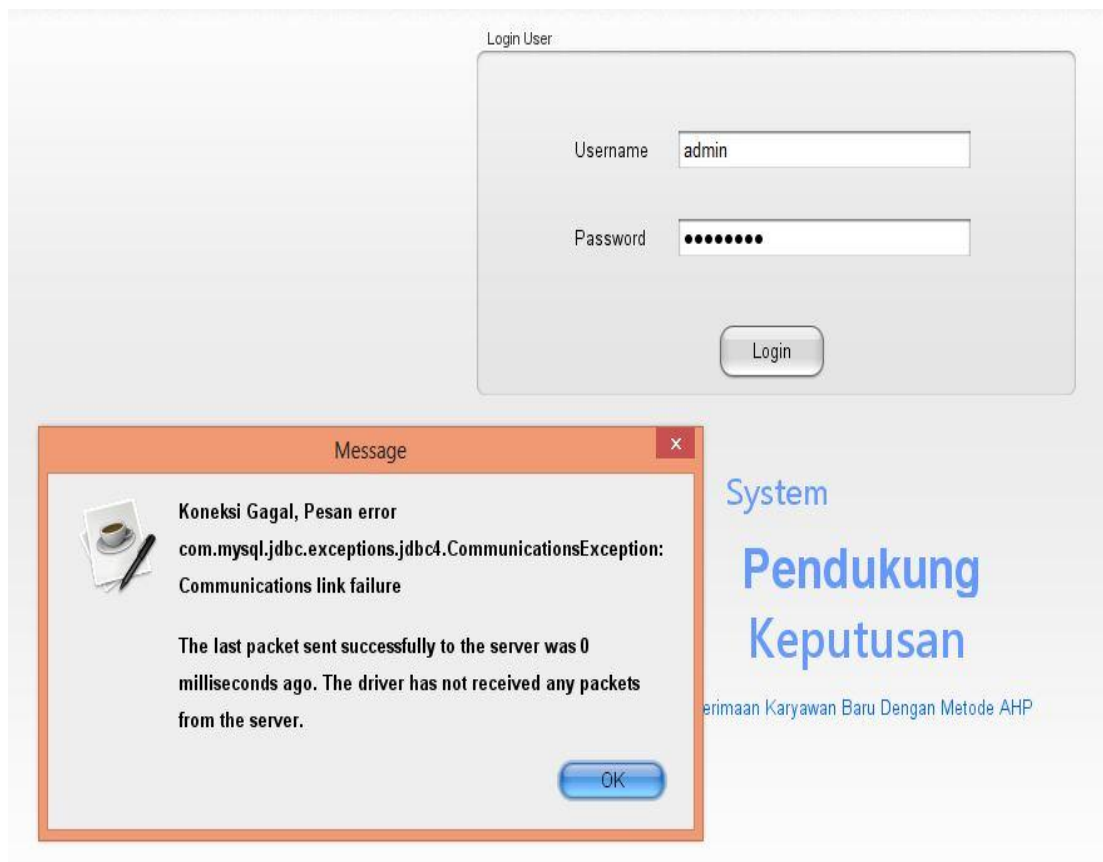
Seleksi Penerimaan Karyawan Baru Dengan Metode AHP

Gambar 4.1 Tampilan Halaman Login

Keterangan Gambar :

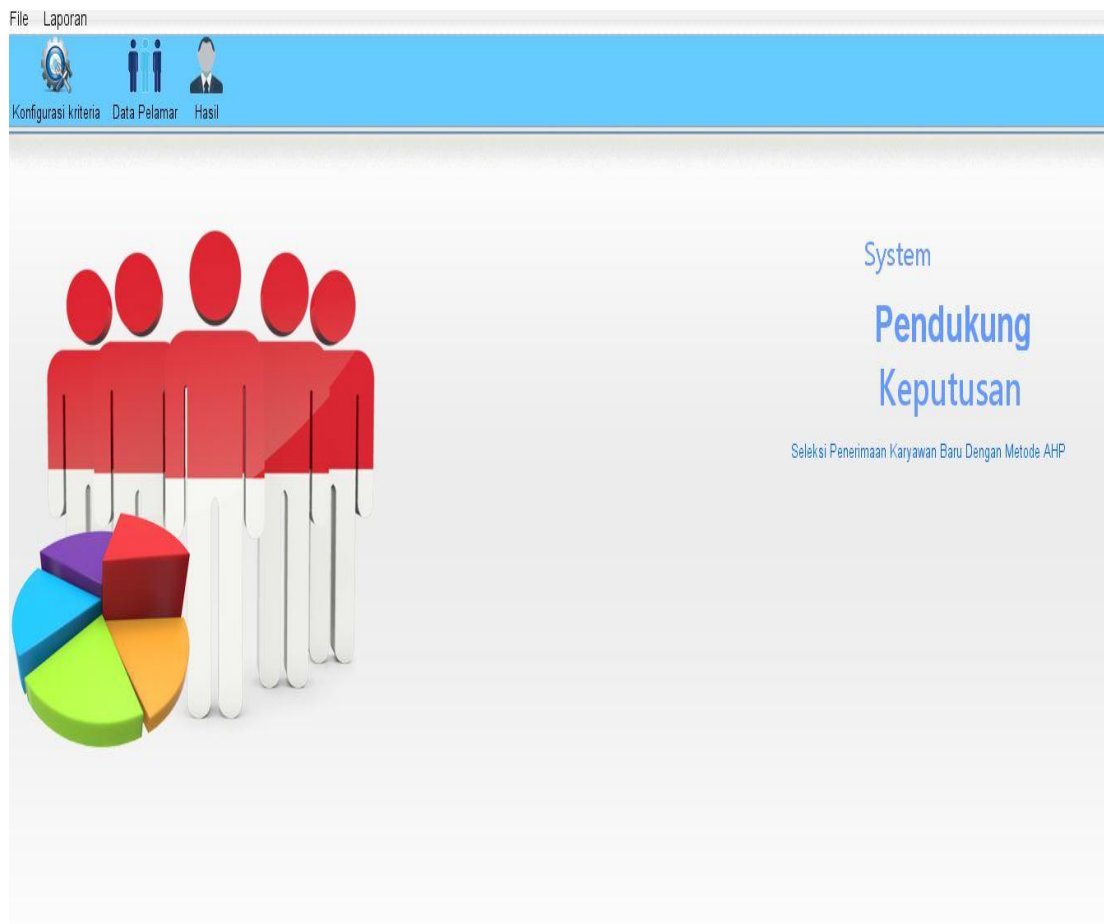
Menu *Login* terdiri dari :

- Kolom User ID, berisikan username pengguna dan harus memasukkan text.
- Kolom Password, berisikan kode verifikasi berupa text atau number yang nanti akan disamakan dengan User ID untuk masuk kedalam sistem.
- Tombol Login, digunakan untuk masuk kedalam sistem jika User ID dan Password benar.



Gambar 4.2 Tampilan Gagal Login

Pada form login (gambar 4.2) masukkan username dan password untuk kemudian akan di klarifikasikan apakah *username* dan *password* tersebut tidak valid. Jika ya, maka pengguna dapat menggunakan sistem ini.



Gambar 4.3 Tampilan Berhasil Login

Pada form login (gambar 4.3) masukkan username dan password untuk kemudian akan di klarifikasikan apakah username dan password tersebut valid . jika ya,maka pengguna mengklarifikasi OK dapat menggunakan sistem ini.

b. Halaman Menu Utama

Form ini adalah menu utama dari sistem yang akan muncul apabila *login* berhasil, dan pada form ini terdapat menu bar Konfigurasi criteria, Data pelamar, dan Hasil. Pada Data Pelamar berisi form data karyawan dan nilai criteria. Konfigurasi criteria terdapat tabel normalisasi perhitungan dari metode AHP. Dan pada Hasil terdapat Hasil dari penerimaan karyawan.



Gambar 4.4 Tampilan Halaman Menu Utama

Keterangan Gambar :

Menu menu bar atas terdiri dari :

1. Menu Konfigurasi Kriteria, menu ini berisi tentang konfigurasi data karyawan seperti konfigurasi nilai bobot, dan olah data sub criteria dari perhitungan matriks dari metode AHP
2. Menu Data Pelamar, menu ini berisi mengenai data calon pelamar
3. Menu Hasil, menu ini berisi hasil dari karyawan yang terpilih sebagai karyawan terbaik.

c. Halaman Input Konfigurasi

Kriteria form yang akan muncul apabila pengguna melakukan pada menu bar yang terdapat pada menu, dengan fungsi untuk menampilkan data karyawan dan memanipulasi pada data, yaitu menambah, mengubah, dan menghapus data karyawan dan data yang ada di dalam database akan ditampilkan secara otomatis ditabel ketika program dijalankan.

Apabila pengguna ingin menambahkan data karyawan, yang pertama dilakukan adalah masuk kehalaman data pelamar, dengan mengisi data-data yang ingin ditambahkan dan apabila ingin disimpan pengguna harus menekan tombol simpan, untuk menghapus data karyawan dapat menekan tombol icon hapus, tombol ubah berfungsi apabila akan dilakukan perubahan pada data karyawan.

Kreteria Bobot

Hitung

Jumlah .

N .

Max .

CI .

CR .

Simpan Bobot

TABEL1	TABEL3
x	x

TABEL2	TABEL4
x	x

Gambar 4.5 Tampilan halaman Konfigurasi Bobot

Nama Kriteria	Nama Sub Kriteria
<div><div>Tambah</div><div>Ubah</div><div>Hapus</div><div>Configurasi Sub_Bobot</div></div>	

Gambar 4.6 Tampilan halaman Olah Data Sub Kriteria

d. Hasil

Form ini adalah form yang akan muncul apabila pengguna melakukan klik menu matriks awal yaitu terdapat tampilan tabel perhitungan hasil pada metode AHP.



The image shows a web application interface. At the top, there is a menu bar with the following items: 'File', 'Konfigurasi', 'Data Pelamar', and 'Hasil'. Below the menu bar, there is a table with the following headers: 'validasi', 'Nama', 'NokTP', and 'Alamat'.

validasi	Nama	NokTP	Alamat
----------	------	-------	--------

Gambar 4.7 Tampilan halaman hasil

4.3 Pengujian

Setelah tahap implementasi dilakukan, maka tahap selanjutnya adalah pengujian sistem. Pengujian sistem ditekankan pada sistem untuk melihat apakah sistem yang dirancang berfungsi sebagaimana mestinya. Pengujian yang telah dilakukan meliputi pengujian menggunakan *White Box* dan *Black Box*.

4.3.1 Pengujian Black Box

Pengujian *Black-Box* berfokus pada persyaratan fungsional perangkat lunak. Dengan demikian pengujian *Black-Box* memungkinkan perekrutan perangkat lunak mendapatkan serangkaian kondisi input yang sepenuhnya menggunakan semua persyaratan fungsional untuk suatu program. Pengujian ini bukan merupakan komplementer dari teknik *White-Box*. Pengujian ini berusaha menemukan kesalahan dalam kategori sebagai berikut :

- a. Fungsi-fungsi yang tidak ada atau hilang
- b. Kesalahan *Interface*
- c. Kesalahan kinerja
- d. Inisialisasi dan kesalahan terminasi

Tidak seperti pengujian *White-Box* yang dilakukan pada saat awal proses pengujian, pengujian *Black-Box* cenderung diaplikasikan selama tahap akhir pengujian.

1. Pengujian Tampilan Menu Awal

NO	Item Uji	Detail Pengujian	Jenis Ujian
1	Login	Verifikasi Login	<i>Black Box</i>
2	Data Pelamar	Tambah data kriteria	<i>Black Box</i>

3	Konfigurasi	Berpindah ke activity konfigurasi bobot	<i>Black Box</i>
4	Konfigurasi	Berpindah ke activity konfigurasi sub bobot	<i>Black Box</i>
5	Hasil	Menampilkan hasil data pelamar	<i>Black Box</i>
6	Menekan Tombol Keluar	Keluar Aplikasi	<i>Black Box</i>

Tabel 4.1 Pengujian Tampilan Menu Awal

2. Hasil Pengujian Pengujian *Black Box*

NO	Data Masukan	Hasil Yang Diharapkan	Kesimpulan
1	Username: admin Password : 123	Proses berhasil data login dimasukan terdaftar kemudian masuk ke sistem	Diterima

Tabel 4.2 Pengujian Tampilan Login (Data Normal)

3. Pengujian Tampilan Menu Login

Test ID	Data Masukan	Hasil Yang Diharapkan	Kesimpulan
1	Username: admin Password : admin	Proses gagal data login dimasukan tidak terdaftar kemudian menampilkan pesan, kesalahan login	Sukses

Tabel 4.3 Pengujian Tampilan Menu Login

4. Pengujian Black Box Input Data Karyawan

No	Data Masukan	Yang Diharapkan	Hasil	Kesimpulan
1	Nama : Dani	Proses berhasil data	Proses	Sukses
2	Nilai SB : 1.00	criteria dimasukkan	berhasil	
3	Nilai B : 0.75	kedalam database		
4	Nilai C : 0.50			

Tabel 4.4 Pengujian Input Data Karyawan

4.4.2 Pengujian White Box

Pengujian white box yang dilakukan adalah melakukan perbandingan antara coding dan gambar pada activity diagram yang ada kemudian lakukan identifikasi persamaannya, pengujian ini diterapkan pada form data barang untuk mengelola data barang.

Source Code	Sequence
<p>To change this template, choose Tools Templates and open the template in the editor.</p> <pre> */ package view; import conn.koneksi; import dialog.SimpanPelamar; import java.sql.ResultSet; import java.sql.SQLException; import java.util.logging.Level; import java.util.logging.Logger; import javax.swing.JOptionPane; import javax.swing.table.DefaultTableModel; public class DataPelamar extends javax.swing.JFrame { static koneksi conn = new koneksi(); static DefaultTableModel tb; /** * Creates new form data_kriteria */ public DataPelamar() { initComponents(); new tbl_model().tabelmodel(); new tbl_model().tampil(); </pre>	<p>Sequence Diagram Data Pelamar</p> <pre> sequenceDiagram actor HRD participant Menu as Menu Data pelamar participant Proses1 as Proses participant Menampilkan as Menampilkan Menu Input Nilai Pelamar participant Proses2 as Proses participant menyimpan as menyimpan Data Pelamar HRD->>Menu: Menu Data Pelamar activate Menu Menu->>Proses1: Proses deactivate Menu activate Proses1 Proses1->>Menampilkan: Tampil() deactivate Proses1 activate Menampilkan Menampilkan->>Proses2: Menginput Data Pelamar deactivate Menampilkan activate Proses2 Proses2->>menyimpan: Simpan deactivate Proses2 activate menyimpan deactivate menyimpan </pre>

<pre> jml_ktr.setText("") jTable1.getRowCount()); } * 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.x */ @SuppressWarnings("unchecked") // <editor-fold defaultstate="collapsed" desc="Generated Code"> private void initComponents() { jScrollPane1 = new javax.swing.JScrollPane(); jTable1 = new javax.swing.JTable(); jToolBar2 = new javax.swing.JToolBar(); jLabel2 = new javax.swing.JLabel(); jml_ktr = new javax.swing.JLabel(); background1 = new cmpnt.background(); jButton1 = new javax.swing.JButton(); jButton2 = new javax.swing.JButton(); jButton3 = new javax.swing.JButton(); </pre>	
--	--


```
setDefaultCloseOperation(javax.swing.WindowConstants.DISPOSE_ON_CLOSE);

jTable1.setModel(new
javax.swing.table.DefaultTableModel(
    new Object [][] {
        { null, null, null, null },
        { null, null, null, null },
        { null, null, null, null },
        { null, null, null, null }
    },
    new String [] {
        "Title 1", "Title 2", "Title 3", "Title
4"
    }
));
jScrollPane1.setViewportViewView(jTable1);

jToolBar2.setFloatable(false);
jToolBar2.setRollover(true);

jLabel2.setText("Jumlah Pelamar :");
jToolBar2.add(jLabel2);

jml_ktr.setText("0");
jToolBar2.add(jml_ktr);
```

```
background1.setBorder(javax.swing.BorderFactory.createBevelBorder(javax.swing.border.BevelBorder.RAISED));

        jButton1.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/img/tambah.png"))); // NOI18N
        jButton1.setText("Tambah");
        jButton1.addActionListener(new
java.awt.event.ActionListener() {
            public void
actionPerformed(java.awt.event.ActionEvent
evt) {
                jButton1ActionPerformed(evt);
            }
        });

        jButton2.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/img/edit.png"))); // NOI18N
        jButton2.setText("Ubah");
        jButton2.addActionListener(new
java.awt.event.ActionListener() {
            public void
actionPerformed(java.awt.event.ActionEvent
evt) {
                jButton2ActionPerformed(evt);
```

```

    }

    });

    jButton3.setIcon(new
javax.swing.ImageIcon(getClass().getResource
ce("/img/hapus.png"))); // NOI18N
    jButton3.setText("Hapus");
    jButton3.addActionListener(new
java.awt.event.ActionListener() {
        public void
actionPerformed(java.awt.event.ActionEvent
evt) {
            jButton3ActionPerformed(evt);
        }
    });

    javax.swing.GroupLayout
background1Layout = new
javax.swing.GroupLayout(background1);

    background1.setLayout(background1Layout);

    background1Layout.setHorizontalGroup(

    background1Layout.createParallelGroup(java
x.swing.GroupLayout.Alignment.LEADING)

    .addGroup(background1Layout.createSequen

```

<pre> tialGroup() .addContainerGap() .addGroup(background1Layout.createParallel Group(javax.swing.GroupLayout.Alignment. LEADING) .addComponent(jButton1, javax.swing.GroupLayout.DEFAULT_SIZE, 197, Short.MAX_VALUE) .addComponent(jButton2, javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE) .addComponent(jButton3, javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)) .addContainerGap()); background1Layout.setVerticalGroup(background1Layout.createParallelGroup(java x.swing.GroupLayout.Alignment.LEADING) .addGroup(background1Layout.createSequen tialGroup() .addGap(85, 85, 85) .addComponent(jButton1) </pre>	
---	--

```
.addPreferredGap(javax.swing.LayoutStyle.C  
omponentPlacement.RELATED)  
    .addComponent(jButton2)  
  
.addPreferredGap(javax.swing.LayoutStyle.C  
omponentPlacement.RELATED)  
    .addComponent(jButton3)  
  
.addContainerGap(javax.swing.GroupLayout.  
DEFAULT_SIZE, Short.MAX_VALUE))  
    );  
  
    javax.swing.GroupLayout layout = new  
    javax.swing.GroupLayout(getContentPane());  
    getContentPane().setLayout(layout);  
    layout.setHorizontalGroup(  
  
    layout.createParallelGroup(javax.swing.Grou  
pLayout.Alignment.LEADING)  
  
    .addGroup(layout.createSequentialGroup()  
        .addComponent(background1,  
        javax.swing.GroupLayout.PREFERRED_SIZ  
E,  
        javax.swing.GroupLayout.DEFAULT_SIZE,  
        javax.swing.GroupLayout.PREFERRED_SIZ  
E)
```

<pre> .addGap(0, 0, 0) .addGroup(layout.createParallelGroup(javax. swing.GroupLayout.Alignment.LEADING) .addComponent(jScrollPane1, javax.swing.GroupLayout.DEFAULT_SIZE, 703, Short.MAX_VALUE) .addComponent(jToolBar2, javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)))); layout.setVerticalGroup(layout.createParallelGroup(javax.swing.Grou pLayout.Alignment.LEADING) .addGroup(layout.createSequentialGroup() .addComponent(jScrollPane1, javax.swing.GroupLayout.DEFAULT_SIZE, 402, Short.MAX_VALUE) .addGap(4, 4, 4) .addComponent(jToolBar2, javax.swing.GroupLayout.PREFERRED_SIZ E, 25, javax.swing.GroupLayout.PREFERRED_SIZ E)) .addComponent(background1, </pre>	
---	--

```

javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE)

    );

    pack();
} // </editor-fold>

private void
jButton1ActionPerformed(java.awt.event.Act
ionEvent evt) {
    // TODO add your handling code here:
    new SimpanPelamar().setVisible(true);
}

private void
jButton2ActionPerformed(java.awt.event.Act
ionEvent evt) {
    // TODO add your handling code here:
    int x = jTable1.getSelectedRow();
    new SimpanPelamar().setVisible(true);
    int f = 0;
    for (int i = 3; i <
jTable1.getColumnCount() - 1; i++) {
        SimpanPelamar.jc.get(i
-
3).setSelectedItem(jTable1.getValueAt(x,
i).toString());
        f++;
    }
}

```

<pre> } SimpanPelamar.jTextField1.setText(jTable1. getValueAt(x, 0).toString()); SimpanPelamar.jTextField2.setText(jTable1. getValueAt(x, 1).toString()); SimpanPelamar.jTextArea1.setText(jTable1. getValueAt(x, 2).toString()); } private void jButton3ActionPerformed(java.awt.event.Act ionEvent evt) { // TODO add your handling code here: int x = jTable1.getSelectedRow(); conn.simpanData("delete from tbl_pelamar where NOKTP=" + jTable1.getValueAt(x, 1).toString()); conn.simpanData("delete from tbl_kriteria_pelamar where NOKTP=" + jTable1.getValueAt(x, 1).toString()); new tbl_model().tabelmodel(); new tbl_model().tampil(); } @param args the command line arguments public static void main(String args[]) { </pre>	
---	--


```

/* Set the Nimbus look and feel */
//<editor-fold    defaultstate="collapsed"
desc=" Look and feel setting code (optional)
">

/* If Nimbus (introduced in Java SE 6) is
not available, stay with the default look and
feel.

    *      For      details      see
http://download.oracle.com/javase/tutorial/uis
wing/lookandfeel/plaf.html
*/

try {
    for
(javax.swing.UIManager.LookAndFeelInfo
info
:
javax.swing.UIManager.getInstalledLookAnd
Feels()) {
        if
("Nimbus".equals(info.getName())) {

javax.swing.UIManager.setLookAndFeel(inf
o.getClassName());
            break;
        }
    }
} catch (ClassNotFoundException ex) {

java.util.logging.Logger.getLogger(DataPela

```

```

mar.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
    } catch (InstantiationException ex) {

java.util.logging.Logger.getLogger(DataPelamar.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
    } catch (IllegalAccessException ex) {

java.util.logging.Logger.getLogger(DataPelamar.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
    }
    catch (javax.swing.UnsupportedLookAndFeelException ex) {

java.util.logging.Logger.getLogger(DataPelamar.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
    }
//</editor-fold>
//</editor-fold>

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

```

```

DataPelamar().setVisible(true);
    }
});
}
// Variables declaration - do not modify
private cmpnt.background background1;
private javax.swing.JButton jButton1;
private javax.swing.JButton jButton2;
private javax.swing.JButton jButton3;
private javax.swing.JLabel jLabel2;
private javax.swing.JScrollPane jScrollPane1;
public static javax.swing.JTable jTable1;
private javax.swing.JToolBar jToolBar2;
private javax.swing.JLabel jLabel1;
// End of variables declaration
public static class tbl_model {

    public void tabelmodel() {
        tb = new DefaultTableModel();
        tb.addColumn("Nama");
        tb.addColumn("NoKTP");
        tb.addColumn("Alamat");
        try {
            ResultSet rs =
conn.ambilData("select * from
tbl_kriteria_pelamar");
            if (rs.next()) {

```

<pre> ResultSet Rename = conn.ambilData("select kriteria from vpelamar group by kriteria order by no asc"); while (Rename.next()) { tb.addColumn(Rename.getString(1).replace(" _ ", " ")); } } jTable1.setModel(tb); } catch (SQLException ex) { JOptionPane.showMessageDialog(null, ex); } } public void tampil() { System.out.println("panggil Rename"); try { ResultSet rs = conn.ambilData("select * from tbl_kriteria_pelamar"); if (rs.next()) { ResultSet Rename = conn.ambilData("call tampil_pelamar()"); System.out.println("Rename: " + </pre>	
--	--

<pre> Rename); for (int i = 0; Rename.next(); i++) { tb.addRow(new Object[]{null}); for (int j = 0; j < 3; j++) { jTable1.setValueAt(Rename.getObject(j + 1), i, j); } for (int j = 3; j < jTable1.getColumnCount(); j++) { jTable1.setValueAt(Rename.getObject(j + 2), i, j); } } } catch (SQLException ex) { Logger.getLogger(DataPelamar.class.getNam e()).log(Level.SEVERE, null, ex); } } } } </pre>	
--	--

Tabel 4.5 White Box