LISTING PROGRAM

A. Data Karyawan

```
1.  /*
2.   * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
3.   * Click nbfs://nbhost/SystemFileSystem/Templates/GUIForms/JInternalFrame.java to edit this template
4.   */
5.  package Form;
6.
7.  import Koneksi.KoneksiDb;
8.  import java.awt.event.KeyEvent;
9.  import java.sql.Connection;
10. import java.sql.ResultSet;
11. import java.sql.Statement;
12. import java.text.DateFormat;
13. import java.text.SimpleDateFormat;
14. import java.sql.Date;
15. import java.sql.SQLException;
16. import javax.swing.JOptionPane;
17. import javax.swing.plaf.basic.BasicInternalFrameUI;
18. import javax.swing.table.DefaultTableModel;
19.
20. /**
21.  *
22.  * @author resar
23.  */
24.
25. public class Karyawan extends javax.swing.JInternalFrame {
26.
27.     /**
28.      * Creates new form Profile
29.      */
30.     KoneksiDb k = new KoneksiDb();
31.     Connection con;
32.     String nipGuru = "";
33.     DateFormat dateFormat = new SimpleDateFormat("dd/MM/yyyy");
34.
35.     public Karyawan() throws SQLException {
36.         this.con = k.getConnect();
37.         initComponents();
38.         this.setBorder(javax.swing.BorderFactory.createEmptyBorder(0,0,0,0));
39.         BasicInternalFrameUI ui = (BasicInternalFrameUI)this.getUI();
40.         ui.setNorthPane(null);
41.         showGuru();
42.         tblGuru.setAutoCreateRowSorter(true);
43.     }
44.
45.     public void showGuru(){
46.         DefaultTableModel model = (DefaultTableModel) tblGuru.getModel();
47.
48.         Statement st;
49.         ResultSet rs;
50.
51.         String query = "SELECT * FROM tbl_alternatif";
52.         String status = null;
53.
54. //        String s2[] = { "male", "female", "others" };
55. //        cbJk = new JComboBox(s2);
56. //
57.         try {
58.             st = con.createStatement();
59.             rs = st.executeQuery(query);
```

```
60.
61.             Object[] row = new Object[4];
62.             int no = 0;
63.             while(rs.next()){
64.                 //row[0] = ++no;
65.                 row[0] = rs.getString("id_alternatif");
66.                 row[1] = rs.getString("nama");
67.                 row[2] = rs.getString("alamat");
68.                 row[3] = rs.getString("telp");
69.                 model.addRow(row);
70.             }
71.         } catch (Exception e) {
72.             e.printStackTrace();
73.         }
74.     }
75.
76.
77.
78.     public void RefreshPage(){
79.         DefaultTableModel model = (DefaultTableModel)tblGuru.getModel();
80.         model.setRowCount(0);
81.         showGuru();
82.         bersih();
83.     }
84.
85.     /**
86.      * This method is called from within the constructor to initialize the form.
87.      * WARNING: Do NOT modify this code. The content of this method is always
88.      * regenerated by the Form Editor.
89.      */
90.     @SuppressWarnings("unchecked")
91.     // <editor-fold defaultstate="collapsed" desc="Generated Code">//GEN-BEGIN:initComponents
92.     private void initComponents() {
93.
94.         jScrollPane1 = new javax.swing.JScrollPane();
95.         tblGuru = new javax.swing.JTable();
96.         btnAdd = new javax.swing.JButton();
97.         btnEdit = new javax.swing.JButton();
98.         btnDelete = new javax.swing.JButton();
99.         lblNIP = new javax.swing.JLabel();
100.        tfNIP = new javax.swing.JTextField();
101.        lblNama = new javax.swing.JLabel();
102.        tfNama = new javax.swing.JTextField();
103.        lblAgama = new javax.swing.JLabel();
104.        tfNoTlp = new javax.swing.JTextField();
105.        lblNoTelp = new javax.swing.JLabel();
106.        btnReset = new javax.swing.JButton();
107.        jScrollPane2 = new javax.swing.JScrollPane();
108.        tfAgama = new javax.swing.JTextArea();
109.
110.        setFocusCycleRoot(false);
111.        setFocusable(false);
112.        setPreferredSize(new java.awt.Dimension(1200, 580));
113.        setRequestFocusEnabled(false);
114.
115.        tblGuru.setAutoCreateRowSorter(true);
116.        tblGuru.setBorder(javax.swing.BorderFactory.createCompoundBorder());
117.        tblGuru.setModel(new javax.swing.table.DefaultTableModel(
118.            new Object []{} {
119.
120.            },
121.            new String [] {
122.                "ID", "NAMA", "ALAMAT", "NO TELP"
123.            }
```

```
124.        ));
125.        tblGuru.setCursor(new java.awt.Cursor(java.awt.Cursor.HAND_CURSOR));
126.        tblGuru.addMouseListener(new java.awt.event.MouseAdapter() {
127.            public void mouseClicked(java.awt.event.MouseEvent evt) {
128.                tblGuruMouseClicked(evt);
129.            }
130.        });
131.        jScrollPane1.setViewportView(tblGuru);
132.
133.        btnAdd.setFont(new java.awt.Font("Segoe UI", 1, 12)); // NOI18N
134.        btnAdd.setText("ADD");
135.        btnAdd.setCursor(new java.awt.Cursor(java.awt.Cursor.HAND_CURSOR));
136.        btnAdd.addActionListener(new java.awt.event.ActionListener() {
137.            public void actionPerformed(java.awt.event.ActionEvent evt) {
138.                btnAddActionPerformed(evt);
139.            }
140.        });
141.
142.        btnEdit.setFont(new java.awt.Font("Segoe UI", 1, 12)); // NOI18N
143.        btnEdit.setText("EDIT");
144.        btnEdit.setCursor(new java.awt.Cursor(java.awt.Cursor.HAND_CURSOR));
145.        btnEdit.addActionListener(new java.awt.event.ActionListener() {
146.            public void actionPerformed(java.awt.event.ActionEvent evt) {
147.                btnEditActionPerformed(evt);
148.            }
149.        });
150.
151.        btnDelete.setFont(new java.awt.Font("Segoe UI", 1, 12)); // NOI18N
152.        btnDelete.setText("DELETE");
153.        btnDelete.setCursor(new java.awt.Cursor(java.awt.Cursor.HAND_CURSOR));
154.        btnDelete.addActionListener(new java.awt.event.ActionListener() {
155.            public void actionPerformed(java.awt.event.ActionEvent evt) {
156.                btnDeleteActionPerformed(evt);
157.            }
158.        });
159.
160.        lblNIP.setFont(new java.awt.Font("Segoe UI", 0, 14)); // NOI18N
161.        lblNIP.setText("ID :");
162.
163.        lblNama.setFont(new java.awt.Font("Segoe UI", 0, 14)); // NOI18N
164.        lblNama.setText("NAMA:");
165.
166.        lblAgama.setFont(new java.awt.Font("Segoe UI", 0, 14)); // NOI18N
167.        lblAgama.setText("ALAMAT:");
168.
169.        tfNoTlp.addKeyListener(new java.awt.event.KeyAdapter() {
170.            public void keyTyped(java.awt.event.KeyEvent evt) {
171.                tfNoTlpKeyTyped(evt);
172.            }
173.        });
174.
175.        lblNoTelp.setFont(new java.awt.Font("Segoe UI", 0, 14)); // NOI18N
176.        lblNoTelp.setText("NO TELP:");
177.
178.        btnReset.setFont(new java.awt.Font("Segoe UI", 1, 12)); // NOI18N
179.        btnReset.setText("RESET");
180.        btnReset.setCursor(new java.awt.Cursor(java.awt.Cursor.HAND_CURSOR));
181.        btnReset.addActionListener(new java.awt.event.ActionListener() {
182.            public void actionPerformed(java.awt.event.ActionEvent evt) {
183.                btnResetActionPerformed(evt);
184.            }
185.        });
186.
187.        tfAgama.setColumns(20);
```

```
188.        tfAgama.setRows(5);
189.        jScrollPane2.setViewportView(tfAgama);
190.
191.        javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
192.        getContentPane().setLayout(layout);
193.        layout.setHorizontalGroup(
194.            layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
195.            .addGroup(layout.createSequentialGroup()
196.                .addGap(30, 30, 30)
197.                .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
198.                    .addComponent(jScrollPane1, javax.swing.GroupLayout.PREFERRED_SIZE, 662,
    javax.swing.GroupLayout.PREFERRED_SIZE)
199.                    .addGroup(layout.createSequentialGroup()
200.                        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
201.                            .addGroup(layout.createSequentialGroup()
202.                                .addComponent(lblNama)
203.                                .addGap(32, 32, 32)
204.                                .addComponent(tfNama, javax.swing.GroupLayout.PREFERRED_SIZE, 119,
    javax.swing.GroupLayout.PREFERRED_SIZE))
205.                            .addGroup(layout.createSequentialGroup()
206.                                .addComponent(lblNIP)
207.                                .addGap(52, 52, 52)
208.                                .addComponent(tfNIP, javax.swing.GroupLayout.PREFERRED_SIZE, 119,
    javax.swing.GroupLayout.PREFERRED_SIZE)
209.                                .addGap(31, 31, 31)
210.                                .addComponent(lblAgama)))
211.                        .addGap(10, 10, 10)
212.                        .addComponent(jScrollPane2, javax.swing.GroupLayout.PREFERRED_SIZE,
    javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
213.                        .addGap(18, 18, 18)
214.                        .addComponent(lblNoTelp)
215.                        .addGap(18, 18, 18)
216.                        .addComponent(tfNoTlp, javax.swing.GroupLayout.PREFERRED_SIZE, 119,
    javax.swing.GroupLayout.PREFERRED_SIZE))
217.                    .addGroup(layout.createSequentialGroup()
218.                        .addComponent(btnAdd, javax.swing.GroupLayout.PREFERRED_SIZE, 125,
    javax.swing.GroupLayout.PREFERRED_SIZE)
219.                        .addGap(18, 18, 18)
220.                        .addComponent(btnEdit, javax.swing.GroupLayout.PREFERRED_SIZE, 125,
    javax.swing.GroupLayout.PREFERRED_SIZE)
221.                        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
222.                        .addComponent(btnDelete, javax.swing.GroupLayout.PREFERRED_SIZE, 125,
    javax.swing.GroupLayout.PREFERRED_SIZE)
223.                        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
224.                        .addComponent(btnReset, javax.swing.GroupLayout.PREFERRED_SIZE, 125,
    javax.swing.GroupLayout.PREFERRED_SIZE)))
225.                .addGap(0, 486, Short.MAX_VALUE))
226.        );
227.        layout.setVerticalGroup(
228.            layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
229.            .addGroup(layout.createSequentialGroup()
230.                .addGap(20, 20, 20)
231.                .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
232.                    .addGroup(layout.createSequentialGroup()
233.                        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
234.                            .addComponent(lblNIP)
235.                            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
236.                                .addComponent(tfNIP, javax.swing.GroupLayout.PREFERRED_SIZE,
    javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
237.                                .addComponent(lblAgama))
238.                            .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
239.                                .addComponent(lblNoTelp)
```

```
                        .addComponent(tfNoTlp, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)))
                    .addGap(13, 13, 13)
                    .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                        .addComponent(lblNama)
                        .addComponent(tfNama, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)))
                .addComponent(jScrollPane2, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
            .addGap(28, 28, 28)
            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                .addComponent(btnAdd)
                .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                    .addComponent(btnEdit)
                    .addComponent(btnDelete)
                    .addComponent(btnReset)))
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
            .addComponent(jScrollPane1, javax.swing.GroupLayout.PREFERRED_SIZE, 190,
javax.swing.GroupLayout.PREFERRED_SIZE)
            .addContainerGap(64, Short.MAX_VALUE))
    );

    pack();
}// </editor-fold>//GEN-END:initComponents

private void tblGuruMouseClicked(java.awt.event.MouseEvent evt) {//GEN-
FIRST:event_tblGuruMouseClicked
    DefaultTableModel model = (DefaultTableModel)tblGuru.getModel();
    int i = tblGuru.getSelectedRow();
    String a = model.getValueAt(i, 0).toString();
    String b = model.getValueAt(i, 1).toString();
    String c = model.getValueAt(i, 2).toString();
    String d = model.getValueAt(i, 3).toString();

    tfNIP.setText(a);
    tfNama.setText(b);
    tfAgama.setText(c);
    tfNoTlp.setText(d);
//      tfNPM.setText(model.getValueAt(i, 0).toString());
//      tfNama.setText(model.getValueAt(i, 1).toString());
//      cbAngkatan.getModel().setSelectedItem(model.getValueAt(i, 2).toString());

}//GEN-LAST:event_tblGuruMouseClicked

private void btnAddActionPerformed(java.awt.event.ActionEvent evt) {//GEN-
FIRST:event_btnAddActionPerformed
    // TODO add your handling code here:
    if(checkField() == 1){
        try {
            Statement st = con.createStatement();
            String sql = "INSERT into tbl_alternatif (id_alternatif, nama, alamat,telp) values
('"+tfNIP.getText()+"', '"+Utility.capitalizeWord(tfNama.getText())+"',
'"+Utility.capitalizeWord(tfAgama.getText())+"', '"+tfNoTlp.getText()+"')";
            st.executeUpdate(sql);
            RefreshPage();
        } catch (SQLException e) {
            JOptionPane.showMessageDialog(null, e);
        }

    } else{
        JOptionPane.showMessageDialog(this, "Isi Semua Field!");
    }
}//GEN-LAST:event_btnAddActionPerformed

```

```
296.    private void btnEditActionPerformed(java.awt.event.ActionEvent evt) {//GEN-
    FIRST:event_btnEditActionPerformed
297.        // TODO add your handling code here:
298.        if(tfNIP.equals("")){
299.            JOptionPane.showMessageDialog(this, "Pilih Data Yang Ingin Di Edit!");
300.        } else{
301.            int conf;
302.                conf = JOptionPane.showConfirmDialog(null, "Apakah Anda Yakin Ingin Mengedit Data
    Ini?"+tfNIP.getText(),
303.                    "Warning",JOptionPane.WARNING_MESSAGE);
304.                try {
305.                Statement st = con.createStatement();
306.                String sql = "UPDATE `tbl_alternatif` SET `nama` = '"
    +Utility.capitalizeWord(tfNama.getText())+"', "+
307.                        "`alamat` = '"+Utility.capitalizeWord(tfAgama.getText())+"', "+
308.                        "`telp` = '"+tfNoTlp.getText()+"' "+
309.                        "WHERE `id_alternatif` ='"+tfNIP.getText()+"'";
310.                st.executeUpdate(sql);
311.            RefreshPage();
312.                } catch (SQLException e) {
313.                    JOptionPane.showMessageDialog(null, e);
314.                }
315.        }
316.    }//GEN-LAST:event_btnEditActionPerformed
317.
318.    private void btnDeleteActionPerformed(java.awt.event.ActionEvent evt) {//GEN-
    FIRST:event_btnDeleteActionPerformed
319.        // TODO add your handling code here:
320.        if(tfNIP.equals(""))
321.         {
322.             JOptionPane.showMessageDialog(this, "Pilih Data Yang Ingin Di Hapus!");
323.         }
324.         else{
325.            int conf;
326.            conf = JOptionPane.showConfirmDialog(null, "Apakah Anda Yakin Ingin Menghapus Data Ini?",
327.                    "Warning",JOptionPane.WARNING_MESSAGE);
328.            if(conf == 0){
329.                try {
330.                Statement st = con.createStatement();
331.                String sql = "DELETE from tbl_alternatif where id_alternatif ='"+tfNIP.getText()+"'";
332.                st.executeUpdate(sql);
333.            RefreshPage();
334.                } catch (SQLException e) {
335.                    JOptionPane.showMessageDialog(null, e);
336.                }
337.            }
338.         }
339.    }//GEN-LAST:event_btnDeleteActionPerformed
340.
341.    private void btnResetActionPerformed(java.awt.event.ActionEvent evt) {//GEN-
    FIRST:event_btnResetActionPerformed
342.        // TODO add your handling code here:
343.        tfNIP.setText("");
344.        tfNama.setText("");
345.        tfAgama.setText("");
346.        tfNoTlp.setText("");
347.    }//GEN-LAST:event_btnResetActionPerformed
348.
349.    private void tfNoTlpKeyTyped(java.awt.event.KeyEvent evt) {//GEN-FIRST:event_tfNoTlpKeyTyped
350.        char c = evt.getKeyChar();
351.        if ( ((c < '0') || (c > '9')) && (c != KeyEvent.VK_BACK_SPACE)) {
352.                evt.consume();
353.        }
354.    }//GEN-LAST:event_tfNoTlpKeyTyped
```

```
355.
356.     public void bersih(){
357.     tfNIP.setText("");
358.         tfNama.setText("");
359.         tfAgama.setText("");
360.         tfNoTlp.setText("");
361.     }
362.
363.     public int checkField(){
364.         if(
365.                 tfNIP.getText().trim().isEmpty() ||
366.                 tfNama.getText().trim().isEmpty() ||
367.                 tfAgama.getText().trim().isEmpty() ||
368.                 tfNoTlp.getText().trim().isEmpty()
369.                 ) {
370.             return 0;
371.         } else{
372.             return 1;
373.         }
374.     }
375.     // Variables declaration - do not modify//GEN-BEGIN:variables
376.     private javax.swing.JButton btnAdd;
377.     private javax.swing.JButton btnDelete;
378.     private javax.swing.JButton btnEdit;
379.     private javax.swing.JButton btnReset;
380.     private javax.swing.JScrollPane jScrollPane1;
381.     private javax.swing.JScrollPane jScrollPane2;
382.     private javax.swing.JLabel lblAgama;
383.     private javax.swing.JLabel lblNIP;
384.     private javax.swing.JLabel lblNama;
385.     private javax.swing.JLabel lblNoTelp;
386.     private javax.swing.JTable tblGuru;
387.     private javax.swing.JTextArea tfAgama;
388.     private javax.swing.JTextField tfNIP;
389.     private javax.swing.JTextField tfNama;
390.     private javax.swing.JTextField tfNoTlp;
391.     // End of variables declaration//GEN-END:variables
392. }
```

B. Data Kriteria

```
1.  /*
2.   * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
3.   * Click nbfs://nbhost/SystemFileSystem/Templates/GUIForms/JInternalFrame.java to edit this template
4.   */
5.  package Form;
6.
7.  import Koneksi.KoneksiDb;
8.  import java.awt.event.KeyEvent;
9.  import java.sql.Connection;
10. import java.sql.ResultSet;
11. import java.sql.Statement;
12. import java.text.DateFormat;
13. import java.text.SimpleDateFormat;
14. import java.sql.Date;
15. import java.sql.SQLException;
16. import javax.swing.JOptionPane;
17. import javax.swing.plaf.basic.BasicInternalFrameUI;
18. import javax.swing.table.DefaultTableModel;
19.
20. /**
21.  *
22.  * @author resar
23.  */
```

```
24.
25.  public class Kriteria extends javax.swing.JInternalFrame {
26.
27.      /**
28.       * Creates new form Profile
29.       */
30.      KoneksiDb k = new KoneksiDb();
31.      Connection con;
32.      String nipGuru = "";
33.      DateFormat dateFormat = new SimpleDateFormat("dd/MM/yyyy");
34.
35.      public Kriteria() throws SQLException {
36.          this.con = k.getConnect();
37.          initComponents();
38.          this.setBorder(javax.swing.BorderFactory.createEmptyBorder(0,0,0,0));
39.          BasicInternalFrameUI ui = (BasicInternalFrameUI)this.getUI();
40.          ui.setNorthPane(null);
41.          showGuru();
42.          tblGuru.setAutoCreateRowSorter(true);
43.      }
44.
45.      public void showGuru(){
46.          DefaultTableModel model = (DefaultTableModel) tblGuru.getModel();
47.
48.          Statement st;
49.          ResultSet rs;
50.
51.          String query = "SELECT * FROM tbl_kriteria";
52.          String status = null;
53.
54.  //        String s2[] = { "male", "female", "others" };
55.  //        cbJk = new JComboBox(s2);
56.  //
57.          try {
58.              st = con.createStatement();
59.              rs = st.executeQuery(query);
60.
61.              Object[] row = new Object[4];
62.              int no = 0;
63.              while(rs.next()){
64.                  //row[0] = ++no;
65.                  row[0] = rs.getString("id_kriteria");
66.                  row[1] = rs.getString("nama_kriteria");
67.                  row[2] = rs.getString("bobot");
68.                  row[3] = rs.getString("flag");
69.                  model.addRow(row);
70.              }
71.          } catch (Exception e) {
72.              e.printStackTrace();
73.          }
74.      }
75.
76.
77.
78.      public void RefreshPage(){
79.          DefaultTableModel model = (DefaultTableModel)tblGuru.getModel();
80.          model.setRowCount(0);
81.          showGuru();
82.          bersih();
83.      }
84.
85.      /**
86.       * This method is called from within the constructor to initialize the form.
87.       * WARNING: Do NOT modify this code. The content of this method is always
```

```
88.      * regenerated by the Form Editor.
89.      */
90.     @SuppressWarnings("unchecked")
91.     // <editor-fold defaultstate="collapsed" desc="Generated Code">//GEN-BEGIN:initComponents
92.     private void initComponents() {
93.
94.         jScrollPane1 = new javax.swing.JScrollPane();
95.         tblGuru = new javax.swing.JTable();
96.         btnAdd = new javax.swing.JButton();
97.         btnEdit = new javax.swing.JButton();
98.         btnDelete = new javax.swing.JButton();
99.         lblNIP = new javax.swing.JLabel();
100.        tfNIP = new javax.swing.JTextField();
101.        lblNama = new javax.swing.JLabel();
102.        tfNama = new javax.swing.JTextField();
103.        lblAgama = new javax.swing.JLabel();
104.        lblNoTelp = new javax.swing.JLabel();
105.        btnReset = new javax.swing.JButton();
106.        jScrollPane2 = new javax.swing.JScrollPane();
107.        tfAgama = new javax.swing.JTextArea();
108.        tfNoTlp = new javax.swing.JComboBox<>();
109.
110.        setFocusCycleRoot(false);
111.        setFocusable(false);
112.        setPreferredSize(new java.awt.Dimension(1200, 580));
113.        setRequestFocusEnabled(false);
114.
115.        tblGuru.setAutoCreateRowSorter(true);
116.        tblGuru.setBorder(javax.swing.BorderFactory.createCompoundBorder());
117.        tblGuru.setModel(new javax.swing.table.DefaultTableModel(
118.            new Object [][] {
119.
120.            },
121.            new String [] {
122.                "ID", "NAMA", "BOBOT", "TIPE"
123.            }
124.        ));
125.        tblGuru.setCursor(new java.awt.Cursor(java.awt.Cursor.HAND_CURSOR));
126.        tblGuru.addMouseListener(new java.awt.event.MouseAdapter() {
127.            public void mouseClicked(java.awt.event.MouseEvent evt) {
128.                tblGuruMouseClicked(evt);
129.            }
130.        });
131.        jScrollPane1.setViewportView(tblGuru);
132.
133.        btnAdd.setFont(new java.awt.Font("Segoe UI", 1, 12)); // NOI18N
134.        btnAdd.setText("ADD");
135.        btnAdd.setCursor(new java.awt.Cursor(java.awt.Cursor.HAND_CURSOR));
136.        btnAdd.addActionListener(new java.awt.event.ActionListener() {
137.            public void actionPerformed(java.awt.event.ActionEvent evt) {
138.                btnAddActionPerformed(evt);
139.            }
140.        });
141.
142.        btnEdit.setFont(new java.awt.Font("Segoe UI", 1, 12)); // NOI18N
143.        btnEdit.setText("EDIT");
144.        btnEdit.setCursor(new java.awt.Cursor(java.awt.Cursor.HAND_CURSOR));
145.        btnEdit.addActionListener(new java.awt.event.ActionListener() {
146.            public void actionPerformed(java.awt.event.ActionEvent evt) {
147.                btnEditActionPerformed(evt);
148.            }
149.        });
150.
151.        btnDelete.setFont(new java.awt.Font("Segoe UI", 1, 12)); // NOI18N
```

```
152.        btnDelete.setText("DELETE");
153.        btnDelete.setCursor(new java.awt.Cursor(java.awt.Cursor.HAND_CURSOR));
154.        btnDelete.addActionListener(new java.awt.event.ActionListener() {
155.            public void actionPerformed(java.awt.event.ActionEvent evt) {
156.                btnDeleteActionPerformed(evt);
157.            }
158.        });
159.
160.        lblNIP.setFont(new java.awt.Font("Segoe UI", 0, 14)); // NOI18N
161.        lblNIP.setText("ID:");
162.
163.        lblNama.setFont(new java.awt.Font("Segoe UI", 0, 14)); // NOI18N
164.        lblNama.setText("NAMA:");
165.
166.        lblAgama.setFont(new java.awt.Font("Segoe UI", 0, 14)); // NOI18N
167.        lblAgama.setText("BOBOT:");
168.
169.        lblNoTelp.setFont(new java.awt.Font("Segoe UI", 0, 14)); // NOI18N
170.        lblNoTelp.setText("TIPE:");
171.
172.        btnReset.setFont(new java.awt.Font("Segoe UI", 1, 12)); // NOI18N
173.        btnReset.setText("RESET");
174.        btnReset.setCursor(new java.awt.Cursor(java.awt.Cursor.HAND_CURSOR));
175.        btnReset.addActionListener(new java.awt.event.ActionListener() {
176.            public void actionPerformed(java.awt.event.ActionEvent evt) {
177.                btnResetActionPerformed(evt);
178.            }
179.        });
180.
181.        tfAgama.setColumns(20);
182.        tfAgama.setRows(5);
183.        jScrollPane2.setViewportView(tfAgama);
184.
185.        tfNoTlp.setModel(new javax.swing.DefaultComboBoxModel<>(new String[] { "benefit", "cost" }));
186.
187.        javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
188.        getContentPane().setLayout(layout);
189.        layout.setHorizontalGroup(
190.            layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
191.            .addGroup(layout.createSequentialGroup()
192.                .addGap(30, 30, 30)
193.                .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
194.                    .addComponent(jScrollPane1, javax.swing.GroupLayout.PREFERRED_SIZE, 662,
    javax.swing.GroupLayout.PREFERRED_SIZE)
195.                    .addGroup(layout.createSequentialGroup()
196.                        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)
197.                            .addGroup(layout.createSequentialGroup()
198.                                .addComponent(lblNama)
199.                                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED,
    javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
200.                                .addComponent(tfNama, javax.swing.GroupLayout.PREFERRED_SIZE, 119,
    javax.swing.GroupLayout.PREFERRED_SIZE))
201.                            .addGroup(layout.createSequentialGroup()
202.                                .addComponent(lblNIP)
203.                                .addGap(52, 52, 52)
204.                                .addComponent(tfNIP, javax.swing.GroupLayout.PREFERRED_SIZE, 119,
    javax.swing.GroupLayout.PREFERRED_SIZE)))
205.                        .addGap(31, 31, 31)
206.                        .addComponent(lblAgama)
207.                        .addGap(10, 10, 10)
208.                        .addComponent(jScrollPane2, javax.swing.GroupLayout.PREFERRED_SIZE,
    javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
209.                        .addGap(18, 18, 18)
210.                        .addComponent(lblNoTelp)
```

```
                    .addGap(18, 18, 18)
                    .addComponent(tfNoTlp, javax.swing.GroupLayout.PREFERRED_SIZE, 142, javax.swing.GroupLayout.PREFERRED_SIZE))
                .addGroup(layout.createSequentialGroup()
                    .addComponent(btnAdd, javax.swing.GroupLayout.PREFERRED_SIZE, 125, javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addGap(18, 18, 18)
                    .addComponent(btnEdit, javax.swing.GroupLayout.PREFERRED_SIZE, 125, javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
                    .addComponent(btnDelete, javax.swing.GroupLayout.PREFERRED_SIZE, 125, javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
                    .addComponent(btnReset, javax.swing.GroupLayout.PREFERRED_SIZE, 125, javax.swing.GroupLayout.PREFERRED_SIZE)))
                .addGap(0, 22, Short.MAX_VALUE))
        );
        layout.setVerticalGroup(
            layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(layout.createSequentialGroup()
                .addGap(20, 20, 20)
                .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                    .addGroup(layout.createSequentialGroup()
                        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                            .addComponent(lblNIP)
                            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                                .addComponent(tfNIP, javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
                                .addComponent(lblAgama))
                            .addGroup(javax.swing.GroupLayout.Alignment.TRAILING, layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                                .addComponent(lblNoTelp)
                                .addComponent(tfNoTlp, javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)))
                        .addGap(10, 10, 10)
                        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                            .addComponent(lblNama)
                            .addComponent(tfNama, javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)))
                    .addComponent(jScrollPane2, javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
                .addGap(28, 28, 28)
                .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                    .addComponent(btnAdd)
                    .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                        .addComponent(btnEdit)
                        .addComponent(btnDelete)
                        .addComponent(btnReset)))
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
                .addComponent(jScrollPane1, javax.swing.GroupLayout.PREFERRED_SIZE, 190, javax.swing.GroupLayout.PREFERRED_SIZE)
                .addContainerGap(64, Short.MAX_VALUE))
        );

        pack();
    }// </editor-fold>//GEN-END:initComponents

    private void tblGuruMouseClicked(java.awt.event.MouseEvent evt) {//GEN-FIRST:event_tblGuruMouseClicked
        DefaultTableModel model = (DefaultTableModel)tblGuru.getModel();
        int i = tblGuru.getSelectedRow();
        String a = model.getValueAt(i, 0).toString();
        String b = model.getValueAt(i, 1).toString();
        String c = model.getValueAt(i, 2).toString();
```

```
263.         String d = model.getValueAt(i, 3).toString();
264.
265.         tfNIP.setText(a);
266.         tfNama.setText(b);
267.         tfAgama.setText(c);
268.         tfNoTlp.setSelectedItem(d);
269.
270.
271. //       tfNPM.setText(model.getValueAt(i, 0).toString());
272. //       tfNama.setText(model.getValueAt(i, 1).toString());
273. //       cbAngkatan.getModel().setSelectedItem(model.getValueAt(i, 2).toString());
274.
275.     }//GEN-LAST:event_tblGuruMouseClicked
276.
277.     private void btnAddActionPerformed(java.awt.event.ActionEvent evt) {//GEN-
    FIRST:event_btnAddActionPerformed
278.         // TODO add your handling code here:
279.         if(checkField() == 1){
280.             try {
281.                 Statement st = con.createStatement();
282.                 String sql = "INSERT into tbl_kriteria (id_kriteria, nama_kriteria, bobot,flag) values
    ('"+tfNIP.getText()+"', '"+Utility.capitalizeWord(tfNama.getText())+"',
    '"+Utility.capitalizeWord(tfAgama.getText())+"', '"+tfNoTlp.getSelectedItem().toString()+"')";
283.                 st.executeUpdate(sql);
284.              RefreshPage();
285.             } catch (SQLException e) {
286.                 JOptionPane.showMessageDialog(null, e);
287.             }
288.
289.         } else{
290.             JOptionPane.showMessageDialog(this, "Isi Semua Field!");
291.         }
292.     }//GEN-LAST:event_btnAddActionPerformed
293.
294.     private void btnEditActionPerformed(java.awt.event.ActionEvent evt) {//GEN-
    FIRST:event_btnEditActionPerformed
295.         // TODO add your handling code here:
296.         if(tfNIP.equals("")){
297.             JOptionPane.showMessageDialog(this, "Pilih Data Yang Ingin Di Edit!");
298.         } else{
299.             int conf;
300.                 conf = JOptionPane.showConfirmDialog(null, "Apakah Anda Yakin Ingin Mengedit Data Ini?",
301.                     "Warning",JOptionPane.WARNING_MESSAGE);
302.             if(conf == 0){
303.                 try {
304.                 Statement st = con.createStatement();
305.                 String sql = "UPDATE `tbl_kriteria` SET `nama_kriteria` = '"
    +Utility.capitalizeWord(tfNama.getText())+"', "+
306.                         "`bobot` = '"+Utility.capitalizeWord(tfAgama.getText())+"', "+
307.                         "`flag` = '"+tfNoTlp.getSelectedItem().toString()+"' "+
308.                         "WHERE `id_kriteria` ='"+tfNIP.getText()+"'";
309.                 st.executeUpdate(sql);
310.   RefreshPage();
311.                 } catch (SQLException e) {
312.                     JOptionPane.showMessageDialog(null, e);
313.                 }
314.
315.             }
316.         }
317.     }//GEN-LAST:event_btnEditActionPerformed
318.
319.     private void btnDeleteActionPerformed(java.awt.event.ActionEvent evt) {//GEN-
    FIRST:event_btnDeleteActionPerformed
320.         // TODO add your handling code here:
```

```
321.        if(tfNIP.equals(""))
322.         {
323.            JOptionPane.showMessageDialog(this, "Pilih Data Yang Ingin Di Hapus!");
324.         }
325.         else{
326.           int conf;
327.           conf = JOptionPane.showConfirmDialog(null, "Apakah Anda Yakin Ingin Menghapus Data Ini?",
328.                 "Warning",JOptionPane.WARNING_MESSAGE);
329.           if(conf == 0){
330.              try {
331.              Statement st = con.createStatement();
332.              String sql = "DELETE from tbl_kriteria where id_kriteria ='"+tfNIP.getText()+"'";
333.              st.executeUpdate(sql);
334.    RefreshPage();
335.              } catch (SQLException e) {
336.                 JOptionPane.showMessageDialog(null, e);
337.              }
338.           }
339.         }
340.    }//GEN-LAST:event_btnDeleteActionPerformed
341.
342.    public void bersih(){
343.    tfNIP.setText("");
344.       tfNama.setText("");
345.       tfAgama.setText("");
346.       tfNoTlp.setSelectedIndex(0);
347.    }
348.
349.    private void btnResetActionPerformed(java.awt.event.ActionEvent evt) {//GEN-
       FIRST:event_btnResetActionPerformed
350.       // TODO add your handling code here:
351.       tfNIP.setText("");
352.       tfNama.setText("");
353.       tfAgama.setText("");
354.       tfNoTlp.setSelectedIndex(0);
355.    }//GEN-LAST:event_btnResetActionPerformed
356.
357.
358.    public int checkField(){
359.       if(
360.             tfNIP.getText().trim().isEmpty() ||
361.             tfNama.getText().trim().isEmpty() ||
362.             tfAgama.getText().trim().isEmpty()
363.             ) {
364.          return 0;
365.       } else{
366.          return 1;
367.       }
368.    }
369.    // Variables declaration - do not modify//GEN-BEGIN:variables
370.    private javax.swing.JButton btnAdd;
371.    private javax.swing.JButton btnDelete;
372.    private javax.swing.JButton btnEdit;
373.    private javax.swing.JButton btnReset;
374.    private javax.swing.JScrollPane jScrollPane1;
375.    private javax.swing.JScrollPane jScrollPane2;
376.    private javax.swing.JLabel lblAgama;
377.    private javax.swing.JLabel lblNIP;
378.    private javax.swing.JLabel lblNama;
379.    private javax.swing.JLabel lblNoTelp;
380.    private javax.swing.JTable tblGuru;
381.    private javax.swing.JTextArea tfAgama;
382.    private javax.swing.JTextField tfNIP;
383.    private javax.swing.JTextField tfNama;
```

```
    private javax.swing.JComboBox<String> tfNoTlp;
    // End of variables declaration//GEN-END:variables
}
```

C. Data Penilaian

```
/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
 * Click nbfs://nbhost/SystemFileSystem/Templates/GUIForms/JInternalFrame.java to edit this template
 */
package Form;

import Koneksi.KoneksiDb;
import Process.PrintPdf;
import Process.ReadWrite;
import Process.ResultData;
import Process.ResultOrder;
import com.itextpdf.text.DocumentException;
import java.awt.Dimension;
import java.awt.event.KeyEvent;
import java.io.FileNotFoundException;
import java.sql.Connection;
import java.sql.ResultSet;
import java.sql.Statement;
import java.text.DateFormat;
import java.text.SimpleDateFormat;
import java.sql.Date;
import java.sql.PreparedStatement;
import java.sql.SQLException;
import java.util.LinkedList;
import javax.swing.JOptionPane;
import javax.swing.plaf.basic.BasicInternalFrameUI;
import javax.swing.table.DefaultTableModel;

/**
 *
 * @author resar
 */

public class Penilaian extends javax.swing.JInternalFrame {

    /**
     * Creates new form Profile
     */
    private PrintPdf printPdf = new PrintPdf();
    private Penilaian tesData;
    private LinkedList<Double> listResult = new LinkedList<>();
    private LinkedList<Double> listResult1 = new LinkedList<>();
    private LinkedList<ResultData> listResultData = new LinkedList<>();
    private ResultData resultData;
    private LinkedList<ResultData> listResultData1 = new LinkedList<>();
    private ResultData resultData1;
    private ResultOrder resultOrder = new ResultOrder();
    private ResultOrder resultOrder1 = new ResultOrder();

    public double[][] data;
    private double[][] dataBagi;
    private double[][] dataPersen;

    private int alternatif = 0;
    private int criteria = 0;

    Connection con;
    Statement st;
```

```
    ResultSet rs;
    PreparedStatement ps;

    public LinkedList<Object> listKriteriaId = new LinkedList<>();
    public LinkedList<Object> listKriteriaNama = new LinkedList<>();
    public LinkedList<String> listHeader = new LinkedList<>();
    public LinkedList<Object> listKriteriaBobot = new LinkedList<>();
    public LinkedList<Object> listKriteriaFlag = new LinkedList<>();

    public LinkedList<Object> listnasabahId = new LinkedList<>();
    public LinkedList<Object> listnasabahNama = new LinkedList<>();

    private ReadWrite readWrite = new ReadWrite();

    public Penilaian() throws SQLException {
        initComponents();
        con = new KoneksiDb().getConnect();
        tampilData();
    }

    /**
     * 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")
    // <editor-fold defaultstate="collapsed" desc="Generated Code">//GEN-BEGIN:initComponents
    private void initComponents() {

        jDialog1 = new javax.swing.JDialog();
        jScrollPane2 = new javax.swing.JScrollPane();
        jTable1 = new javax.swing.JTable();
        jLabel1 = new javax.swing.JLabel();
        jScrollPane3 = new javax.swing.JScrollPane();
        jTable2 = new javax.swing.JTable();
        jLabel2 = new javax.swing.JLabel();
        jButton1 = new javax.swing.JButton();
        jButton2 = new javax.swing.JButton();
        jScrollPane1 = new javax.swing.JScrollPane();
        tblProsesHitung = new javax.swing.JTable();
        btnAdd = new javax.swing.JButton();
        btnEdit = new javax.swing.JButton();
        btnReset = new javax.swing.JButton();

        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"
            }
        ));
        jScrollPane2.setViewportView(jTable1);

        jLabel1.setText("HASIL PERHITUNGAN SAW");

        jTable2.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"
        }
    ));
    jScrollPane3.setViewportView(jTable2);

    jLabel2.setText("NORMALISASI");

    jButton1.setText("Hitung");
    jButton1.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            jButton1ActionPerformed(evt);
        }
    });

    jButton2.setText("Clear");
    jButton2.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            jButton2ActionPerformed(evt);
        }
    });

    javax.swing.GroupLayout jDialog1Layout = new javax.swing.GroupLayout(jDialog1.getContentPane());
    jDialog1.getContentPane().setLayout(jDialog1Layout);
    jDialog1Layout.setHorizontalGroup(
        jDialog1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(jDialog1Layout.createSequentialGroup()
            .addGroup(jDialog1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                .addGroup(jDialog1Layout.createSequentialGroup()
                    .addContainerGap()
                    .addComponent(jScrollPane2, javax.swing.GroupLayout.DEFAULT_SIZE, 770,
Short.MAX_VALUE))
                .addGroup(jDialog1Layout.createSequentialGroup()

.addGroup(jDialog1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                        .addGroup(jDialog1Layout.createSequentialGroup()
                            .addGap(344, 344, 344)
                            .addComponent(jLabel1))
                        .addGroup(jDialog1Layout.createSequentialGroup()
                            .addGap(368, 368, 368)
                            .addComponent(jLabel2))
                        .addGroup(jDialog1Layout.createSequentialGroup()
                            .addContainerGap()
                            .addComponent(jButton1)
                            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                            .addComponent(jButton2)))
                    .addGap(0, 0, Short.MAX_VALUE)))
            .addContainerGap())
        .addGroup(jDialog1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(jDialog1Layout.createSequentialGroup()
                .addContainerGap()
                .addComponent(jScrollPane3, javax.swing.GroupLayout.DEFAULT_SIZE, 770,
Short.MAX_VALUE)
                .addContainerGap()))
    );
    jDialog1Layout.setVerticalGroup(
        jDialog1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(jDialog1Layout.createSequentialGroup()
            .addGap(10, 10, 10)
            .addComponent(jLabel1)
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, 360,
Short.MAX_VALUE)
```

```
183.                 .addComponent(jLabel2)
184.                 .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
185.                 .addComponent(jScrollPane2, javax.swing.GroupLayout.PREFERRED_SIZE, 328,
    javax.swing.GroupLayout.PREFERRED_SIZE)
186.                 .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
187.                 .addGroup(jDialog1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
188.                     .addComponent(jButton1)
189.                     .addComponent(jButton2))
190.                 .addGap(8, 8, 8))
191.             .addGroup(jDialog1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
192.                 .addGroup(jDialog1Layout.createSequentialGroup()
193.                     .addGap(49, 49, 49)
194.                     .addComponent(jScrollPane3, javax.swing.GroupLayout.PREFERRED_SIZE, 328,
    javax.swing.GroupLayout.PREFERRED_SIZE)
195.                     .addContainerGap(423, Short.MAX_VALUE)))
196.         );
197.
198.         setFocusCycleRoot(false);
199.         setFocusable(false);
200.         setPreferredSize(new java.awt.Dimension(1200, 580));
201.         setRequestFocusEnabled(false);
202.
203.         jScrollPane1.setAutoscrolls(true);
204.
205.         tblProsesHitung.setAutoCreateRowSorter(true);
206.         tblProsesHitung.setBorder(javax.swing.BorderFactory.createCompoundBorder());
207.         tblProsesHitung.setModel(new javax.swing.table.DefaultTableModel(
208.             new Object [][] {
209.                 {}
210.             },
211.             new String [] {
212.
213.             }
214.         ));
215.         tblProsesHitung.setCursor(new java.awt.Cursor(java.awt.Cursor.HAND_CURSOR));
216.         jScrollPane1.setViewportView(tblProsesHitung);
217.
218.         btnAdd.setFont(new java.awt.Font("Segoe UI", 1, 12)); // NOI18N
219.         btnAdd.setText("ADD");
220.         btnAdd.setCursor(new java.awt.Cursor(java.awt.Cursor.HAND_CURSOR));
221.         btnAdd.addActionListener(new java.awt.event.ActionListener() {
222.             public void actionPerformed(java.awt.event.ActionEvent evt) {
223.                 btnAddActionPerformed(evt);
224.             }
225.         });
226.
227.         btnEdit.setFont(new java.awt.Font("Segoe UI", 1, 12)); // NOI18N
228.         btnEdit.setText("EDIT");
229.         btnEdit.setCursor(new java.awt.Cursor(java.awt.Cursor.HAND_CURSOR));
230.         btnEdit.addActionListener(new java.awt.event.ActionListener() {
231.             public void actionPerformed(java.awt.event.ActionEvent evt) {
232.                 btnEditActionPerformed(evt);
233.             }
234.         });
235.
236.         btnReset.setFont(new java.awt.Font("Segoe UI", 1, 12)); // NOI18N
237.         btnReset.setText("RESET");
238.         btnReset.setCursor(new java.awt.Cursor(java.awt.Cursor.HAND_CURSOR));
239.         btnReset.addActionListener(new java.awt.event.ActionListener() {
240.             public void actionPerformed(java.awt.event.ActionEvent evt) {
241.                 btnResetActionPerformed(evt);
242.             }
243.         });
244.
```

```
245.        javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
246.        getContentPane().setLayout(layout);
247.        layout.setHorizontalGroup(
248.            layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
249.            .addGroup(layout.createSequentialGroup()
250.                .addGap(30, 30, 30)
251.                .addComponent(jScrollPane1, javax.swing.GroupLayout.PREFERRED_SIZE, 578,
    javax.swing.GroupLayout.PREFERRED_SIZE)
252.                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
253.                .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
254.                    .addComponent(btnAdd, javax.swing.GroupLayout.PREFERRED_SIZE, 125,
    javax.swing.GroupLayout.PREFERRED_SIZE)
255.                    .addComponent(btnEdit, javax.swing.GroupLayout.PREFERRED_SIZE, 125,
    javax.swing.GroupLayout.PREFERRED_SIZE)
256.                    .addComponent(btnReset, javax.swing.GroupLayout.PREFERRED_SIZE, 125,
    javax.swing.GroupLayout.PREFERRED_SIZE))
257.                .addGap(0, 0, Short.MAX_VALUE))
258.        );
259.        layout.setVerticalGroup(
260.            layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
261.            .addGroup(layout.createSequentialGroup()
262.                .addContainerGap()
263.                .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
264.                    .addComponent(jScrollPane1, javax.swing.GroupLayout.PREFERRED_SIZE, 359,
    javax.swing.GroupLayout.PREFERRED_SIZE)
265.                    .addGroup(layout.createSequentialGroup()
266.                        .addComponent(btnAdd)
267.                        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
268.                        .addComponent(btnEdit)
269.                        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
270.                        .addComponent(btnReset)))
271.                .addContainerGap(180, Short.MAX_VALUE))
272.        );
273.
274.        pack();
275.    }// </editor-fold>//GEN-END:initComponents
276.
277.    private void btnAddActionPerformed(java.awt.event.ActionEvent evt) {//GEN-
    FIRST:event_btnAddActionPerformed
278.
279.        // TODO add your handling code here:
280.        boolean check = true;
281.
282.        for (int i = 0; i < tblProsesHitung.getRowCount()-2; i++) {
283.            for (int j = 2; j < tblProsesHitung.getColumnCount(); j++) {
284.                String value = (String) tblProsesHitung.getValueAt(i, j);
285.                //System.out.print("Value("+i+","+j+") : "+isNumeric(value)+" || ");
286.                if (value != null) {
287.                    if(isNumeric(value) == false){
288.                        check = false;
289.                    }
290.                }else if(value == null){
291.                    check = false;
292.                }
293.            }
294.            System.out.println();
295.        }
296.
297.        if (check) {
298.            tblProsesHitung.editCellAt(-1, -1);
299.            tblProsesHitung.setRowSelectionInterval(0, 0);
300.            tblProsesHitung.setEnabled(false);
301.            JOptionPane.showMessageDialog(null, "Data sudah disimpan");
302.            jDialog1.setVisible(true);
```

```
303.             jDialog1.setSize(new Dimension(800, 800));
304.  jDialog1.setResizable(false);
305.  jDialog1.setLocationRelativeTo(null);
306.  
307.         } else {
308.             JOptionPane.showMessageDialog(null, "Mohon cek kembali data yang anda inputkan");
309.         }
310.     }//GEN-LAST:event_btnAddActionPerformed
311.  
312.     private void btnEditActionPerformed(java.awt.event.ActionEvent evt) {//GEN-
    FIRST:event_btnEditActionPerformed
313.         // TODO add your handling code here:
314.         tblProsesHitung.setEnabled(true);
315.     }//GEN-LAST:event_btnEditActionPerformed
316.  
317.     private void btnResetActionPerformed(java.awt.event.ActionEvent evt) {//GEN-
    FIRST:event_btnResetActionPerformed
318.         // TODO add your handling code here:
319.          try {
320.             tblProsesHitung.setEnabled(true);
321.             refresh();
322.             tampilData();
323.         } catch (SQLException ex) {
324.         }
325.     }//GEN-LAST:event_btnResetActionPerformed
326.  
327.     private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {//GEN-
    FIRST:event_jButton2ActionPerformed
328.  
329.  
330.  
331.         // TODO add your handling code here:
332.             DefaultTableModel mdlClear = (DefaultTableModel) jTable1.getModel();
333.         mdlClear.setRowCount(0);
334.         listResult.clear();
335.         listResultData.clear();
336.  
337.         DefaultTableModel mdlClear1 = (DefaultTableModel) jTable2.getModel();
338.         mdlClear1.setRowCount(0);
339.         listResult1.clear();
340.         listResultData1.clear();
341.     }//GEN-LAST:event_jButton2ActionPerformed
342.  
343.     private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {//GEN-
    FIRST:event_jButton1ActionPerformed
344.         // TODO add your handling code here:
345.         DefaultTableModel mdlClear = (DefaultTableModel) jTable2.getModel();
346.         mdlClear.setRowCount(0);
347.         listResult.clear();
348.  
349.         DefaultTableModel mdlClear1 = (DefaultTableModel) jTable1.getModel();
350.         mdlClear1.setRowCount(0);
351.         listResult1.clear();
352.  
353.         //proses menghapus semua isi di folder laporan
354.         printPdf.removeDir("laporan");
355.  
356.  
357.  
358.         alternatif = tblProsesHitung.getRowCount()-2;
359.         criteria = tblProsesHitung.getColumnCount()-2;
360.  
361.         data = new double[alternatif][criteria];
362.         dataBagi = new double[alternatif][criteria];
```

```
363.        dataPersen = new double[alternatif][criteria];
364.
365.        //Proses input data dari table ke dalam array 2D
366.        for (int i = 0; i < alternatif; i++) {
367.            for (int j = 0; j < criteria; j++) {
368.                data[i][j] = Double.parseDouble(tblProsesHitung.getValueAt(i, j+2).toString());
369.            }
370.        }
371.
372.        System.out.println("\n============== DATA ASLI ================");
373.        for (int i = 0; i < alternatif; i++) {
374.            for (int j = 0; j < criteria; j++) {
375.                System.out.print(data[i][j] + "\t");
376.            }
377.            System.out.println("");
378.        }
379.
380.        //PROSES PRINT DATA
381.        try {
382.            printPdf.print("DATA NILAI KRITERIA", "data nilai kriteria", listKriteriaId, listnasabahId,
    listnasabahNama, data);
383.        } catch (FileNotFoundException ex) {
384.        } catch (DocumentException ex) {
385.        }
386.
387.        //Proses Membagi Nilai
388.        for (int i = 0; i < criteria; i++) {
389.            String value = tblProsesHitung.getValueAt(tblProsesHitung.getRowCount() - 1, i + 2).toString();
390.            if (value.equals("benefit")) {
391.                double max = 0;
392.                for (int j = 0; j < alternatif; j++) {
393.                    double nilai = data[j][i];
394.                    if (nilai > max) {
395.                        max = nilai;
396.                    }
397.                }
398.
399.                for (int j = 0; j < alternatif; j++) {
400.                    double val = data[j][i] / max;
401.                    dataBagi[j][i] = Math.round(val*1000.0)/1000.0;
402.                }
403.            } else {
404.                double min = 1000000000;
405.                for (int j = 0; j < alternatif; j++) {
406.                    double nilai = data[j][i];
407.                    if (nilai < min) {
408.                        min = nilai;
409.                    }
410.                }
411.
412.                for (int j = 0; j < alternatif; j++) {
413.                    double val = min / data[j][i];
414.                    dataBagi[j][i] = Math.round(val*1000.0)/1000.0;
415.                }
416.            }
417.        }
418.
419.        System.out.println("\n\n=================DATA SETELAH
    DIBAGI====================");
420.        for (int i = 0; i < alternatif; i++) {
421.            for (int j = 0; j < criteria; j++) {
422.                System.out.print(dataBagi[i][j] + "\t");
423.            }
424.            System.out.println();
```

```
425.        }
426.        //PROSES PRINT DATA
427.        try {
428.            printPdf.print("DATA NILAI TERNORMALISASI", "data nilai ternormalisasi", listKriteriaId,
    listnasabahId, listnasabahNama, dataBagi);
429.        } catch (FileNotFoundException ex) {
430.        } catch (DocumentException ex) {
431.        }
432.
433.        //Proses Menghitung Dengan Persentase
434.        for (int i = 0; i < criteria; i++) {
435.            double persentase = Float.parseFloat(tblProsesHitung.getValueAt(tblProsesHitung.getRowCount() - 2, i
    + 2).toString());
436.            System.out.print(persentase + "%\t");
437.            for (int j = 0; j < alternatif; j++) {
438.                double val = (dataBagi[j][i] * persentase) / 100f;
439.                dataPersen[j][i] = Math.round(val*1000.0)/1000.0;
440.            }
441.        }
442.
443.        System.out.println("\n\n=================DATA PERSENTASE====================");
444.
445.        DefaultTableModel mdl1 = new DefaultTableModel();
446.        mdl1.addColumn("C1");
447.        mdl1.addColumn("C2");
448.        mdl1.addColumn("C3");
449.        mdl1.addColumn("C4");
450.        for (int i = 0; i < alternatif; i++) {
451.            for (int j = 0; j < criteria; j++) {
452.                System.out.print(dataPersen[i][j] + "\t");
453.                mdl1.addRow(new Object[]{
454.                dataPersen[i][0],
455.                dataPersen[i][1],
456.                dataPersen[i][2],
457.                dataPersen[i][3]
458.            });
459.            }
460.            System.out.println();
461.        }
462.        jTable1.setModel(mdl1);
463.
464.        //Proses menjumlahkan setiap kriteria
465.        for (int i = 0; i < alternatif; i++) {
466.            double sum = 0;
467.            for (int j = 0; j < criteria; j++) {
468.                sum += dataPersen[i][j];
469.            }
470.            listResult.add(Math.round(sum*1000.0)/1000.0);
471.        }
472.
473.        //PROSES PRINT DATA
474. //       try {
475. //           printPdf.print("DATA HASIL PERBANDINGAN", "data hasil perbandingan", listKriteriaId,
    listnasabahId, listnasabahNama, dataPersen);
476. //       } catch (FileNotFoundException ex) {
477. //           Logger.getLogger(M_ProsesHitung.class.getName()).log(Level.SEVERE, null, ex);
478. //       } catch (DocumentException ex) {
479. //           Logger.getLogger(M_ProsesHitung.class.getName()).log(Level.SEVERE, null, ex);
480. //       }
481.
482.        for(int i=0; i<listResult.size(); i++){
483.            LinkedList<Double> listValue = new LinkedList<>();
484.            int cols = tblProsesHitung.getColumnCount();
485.            for(int j=2; j<cols; j++){
```

```
            double val = Float.parseFloat(tblProsesHitung.getValueAt(i, j).toString());
            listValue.add(val);
        }
        resultData = new ResultData();
        resultData.setAlternatif("Alternatif "+(i+1));
        resultData.setNamaAlternatif(tblProsesHitung.getValueAt(i, 1).toString());
        resultData.setListValue(listValue);
        resultData.setResult(listResult.get(i));

        listResultData.add(resultData);
    }

    ResultData[] rd = new ResultData[listResultData.size()];
    for(int i=0; i<rd.length; i++){
        rd[i] = listResultData.get(i);
    }

    LinkedList<ResultData> listFinalResult = resultOrder.getResultData(rd);
    String sqln = "TRUNCATE table tbl_hasil";
    try {
        ps = con.prepareStatement(sqln);
        ps.execute();
    } catch (SQLException ex) {
    }


    String sql = "insert into tbl_hasil(alternatif, nama, c1, c2,c3,c4,hasil) values(?,?,?,?,?,?,?)";

    try {
        ps = con.prepareStatement(sql);
    } catch (SQLException ex) {
    }


    for(int i=0; i<listFinalResult.size(); i++){
        try {
            ps.setString(1, listFinalResult.get(i).getAlternatif());
            ps.setString(2, listFinalResult.get(i).getNamaAlternatif());
            System.out.print(listFinalResult.get(i).getAlternatif()+"||");
        System.out.print(listFinalResult.get(i).getNamaAlternatif()+"||");
        int c = 3;
        for(int j=0; j<listFinalResult.get(i).getListValue().size(); j++){
            System.out.print(listFinalResult.get(i).getListValue().get(j)+"||");
                ps.setDouble(c, listFinalResult.get(i).getListValue().get(j));
            c++;
        }
        System.out.println(listFinalResult.get(i).getResult());
        ps.setDouble(c, listFinalResult.get(i).getResult());
        ps.execute();
        } catch (SQLException ex) {
        }



    }

    //Proses menampilkan di tabel
    DefaultTableModel mdl = new DefaultTableModel();
    mdl.addColumn("Alternatif");
    mdl.addColumn("Nama");
    mdl.addColumn("Hasil");
    for (int i = 0; i < listResult.size(); i++) {
        mdl.addRow(new Object[]{
```

```
                listFinalResult.get(i).getAlternatif(),
                listFinalResult.get(i).getNamaAlternatif(),
                listFinalResult.get(i).getResult()
            });
        }
        jTable2.setModel(mdl);

        //PROSES PRINT DATA
//        try {
//            printPdf.print("DATA NILAI HASIL PERANKINGAN", "data nilai hasil perankingan", listFinalResult);
//        } catch (FileNotFoundException ex) {
//            Logger.getLogger(M_ProsesHitung.class.getName()).log(Level.SEVERE, null, ex);
//        } catch (DocumentException ex) {
//            Logger.getLogger(M_ProsesHitung.class.getName()).log(Level.SEVERE, null, ex);
//        }
        //PROSES PRINT DATA
        try {
            printPdf.print("DATA NILAI HASIL PERANKINGAN", "data nilai hasil perankingan", listKriteriaId, listFinalResult);
        } catch (FileNotFoundException ex) {
        } catch (DocumentException ex) {
        }

    }//GEN-LAST:event_jButton1ActionPerformed


    void tampilData() throws SQLException {
        DefaultTableModel mdl = new DefaultTableModel();
        tblProsesHitung.setModel(mdl);

        //proses menentukan header
        mdl.addColumn("<html><b>Alternatif</b><html>");
        mdl.addColumn("<html><b>Nama</b><html>");
        try {
            st = con.createStatement();
            rs = st.executeQuery("select * from tbl_kriteria");
            while (rs.next()) {
                listKriteriaId.add(rs.getString("id_kriteria"));
                listKriteriaNama.add(rs.getString("nama_kriteria"));
                listKriteriaBobot.add(rs.getString("bobot"));
                listKriteriaFlag.add(rs.getString("flag"));
            }
        } catch (Exception e) {
            System.out.println("#ERROR " + e.getMessage());
        }

        for (int i = 0; i < listKriteriaId.size(); i++) {
            listHeader.add("<html><b>" + (String) listKriteriaId.get(i) + " (" + (String) listKriteriaNama.get(i) + ")</b><html>");
        }

        for (String i : listHeader) {
            mdl.addColumn(i);
        }

        //Proses Menentukan Isi
        try {
            st = con.createStatement();
            rs = st.executeQuery("select * from tbl_alternatif ORDER BY LENGTH(id_alternatif), id_alternatif");
            while (rs.next()) {
                listnasabahId.add(rs.getString("id_alternatif"));
                listnasabahNama.add(rs.getString("nama"));
                mdl.addRow(new Object[]{
```

```
                rs.getString("id_alternatif"),
                rs.getString("nama")
            });
            tblProsesHitung.setModel(mdl);
        }
    } catch (Exception e) {
        System.out.println("#ERROR " + e.getMessage());
    }

    //Proses Menentukan Bobot dan Flag
    mdl.addRow(new Object[]{
        "<html><b>Bobot (%)</b></html>",
        " - "
    });

    mdl.addRow(new Object[]{
        "<html><b>Flag</b><html>",
        " - "
    });
    //mdl.isCellEditable(0, 0);

    //proses agar tabel ada scroll
    tblProsesHitung.setModel(mdl);
    for (int i = 0; i < mdl.getColumnCount(); i++) {
        tblProsesHitung.getColumnModel().getColumn(i).setPreferredWidth(150);
    }

    int rows = tblProsesHitung.getRowCount() - 1;
    int cols = tblProsesHitung.getColumnCount();

    for (int i = 2; i < cols; i++) {
        tblProsesHitung.setValueAt(listKriteriaBobot.get(i - 2), rows - 1, i);
        tblProsesHitung.setValueAt(listKriteriaFlag.get(i - 2), rows, i);
    }

    tblProsesHitung.setShowGrid(true);
}

public void refresh() {
    listHeader.clear();
    listKriteriaBobot.clear();
    listKriteriaFlag.clear();
    listKriteriaId.clear();
    listKriteriaNama.clear();

    listnasabahId.clear();
    listnasabahNama.clear();
}

public boolean isNumeric(String str) {
    try {
        double d = Double.parseDouble(str);
    } catch (NumberFormatException nfe) {
        return false;
    }
    return true;
}
// Variables declaration - do not modify//GEN-BEGIN:variables
private javax.swing.JButton btnAdd;
private javax.swing.JButton btnEdit;
private javax.swing.JButton btnReset;
private javax.swing.JButton jButton1;
private javax.swing.JButton jButton2;
private javax.swing.JDialog jDialog1;
```

```
675.     private javax.swing.JLabel jLabel1;
676.     private javax.swing.JLabel jLabel2;
677.     private javax.swing.JScrollPane jScrollPane1;
678.     private javax.swing.JScrollPane jScrollPane2;
679.     private javax.swing.JScrollPane jScrollPane3;
680.     private javax.swing.JTable jTable1;
681.     private javax.swing.JTable jTable2;
682.     public javax.swing.JTable tblProsesHitung;
683.     // End of variables declaration//GEN-END:variables
684. }
```