

LAMPIRAN

KODING PROGRAM

```

import java.util.logging.Logger;
import javax.swing.JOptionPane;
import javax.swing.table.DefaultTableModel;
import javax.swing.ImageIcon;
import javax.swing.JFileChooser;
import java.sql.ResultSet;
import javax.swing.JOptionPane;
import java.sql.Connection;
import java.sql.SQLException;
import java.sql.Statement;
import java.sql.*;
import javax.swing.table.DefaultTableModel;
import javax.swing.*;
import java.awt.*;
import java.awt.event.*;
import koneksi.database;
import koneksi.koneksi;

public class form_alternative extends javax.swing.JInternalFrame {
String data[]=new String[7];
database dbsetting;
String driver, dtbase, user,pass,userLogin;
    private Connection con;
    private Statement st;
    String Tanggal1;
    ImageIcon imageIcon;
    /** Creates new form form_alternative */
    public form_alternative() {
        initComponents();
        dbsetting = new database();
        driver = dbsetting.SettingPanel("DBDriver");

```

```

dtbase = dbsetting.SettingPanel("DBDatabase");
user = dbsetting.SettingPanel("DBUsername");
pass = dbsetting.SettingPanel("DBPassword");
AutoNumber();
jTable1.setModel(tableModel);
Table(jTable1, new int[]{70,120});
setDefaultTable();
}
public void AutoNumber() {
    try {
        Class.forName(driver);
        Connection kon = DriverManager.getConnection(dtbase, user, pass);
        Statement stt = kon.createStatement();
        String sql = "SELECT * FROM t_alternative ORDER BY kd_alternatif
DESC";
        ResultSet rs = stt.executeQuery(sql);
        if (rs.next()) {
            String kd_barang = rs.getString("kd_alternatif").substring(1);
            String AN = "" + (Integer.parseInt(kd_barang) + 1);
            String Nol = "";
            if(AN.length()==1)
                {Nol = "";}
            jTextField1.setText("A"+Nol + AN);//sesuaikan dengan variable
namanya
        } else {
            jTextField1.setText("A1");//sesuaikan dengan variable namanya
        }
        rs.close();
        kon.close();
    } catch (Exception e) {
        e.printStackTrace();//penanganan masalah
    }
}

```

```

    }
}
int row=0;
private                                     javax.swing.table.DefaultTableModel
tableModel=getDefaultTableModel();
private void Table(javax.swing.JTable tb, int lebar[] ) {
tb.setAutoResizeMode(JTable.AUTO_RESIZE_OFF);
int kolom=tb.getColumnCount();
for(int i=0;i < kolom;i++) {
javax.swing.table.TableColumn tbc=tb.getColumnModel().getColumn(i);
tbc.setPreferredWidth(lebar[i]);
tb.setRowHeight(17);
}}
private javax.swing.table.DefaultTableModel getDefaultTableModel() {
return new javax.swing.table.DefaultTableModel(
new Object[][] {},
new String [] {"Kode Alternative","Nama Alternative"}
){
boolean[] canEdit = new boolean[]{
false, false, false, false
};
@Override
public boolean isCellEditable(int rowIndex, int columnIndex){
return canEdit[columnIndex];
}
};
}
private void setDefaultTable() {
String stat ="";
try {
Class.forName(driver);

```

```

Connection con = DriverManager.getConnection(dtbase,user,pass);
Statement st = con.createStatement();
    String SQL = "SELECT * FROM t_alternative order by kd_alternatif Asc";
    ResultSet res = st.executeQuery(SQL);
    while(res.next()){
    data[0] = res.getString(1);
    data[1] = res.getString(2);
    tableModel.addRow(data);
    }
    } catch (Exception ex) {
    JOptionPane.showMessageDialog(null, "Koneksi database error");
    }
}

public void refresh() throws SQLException{
    String SQL = "select * from t_alternative order by kd_alternatif Asc";
    Connection con = DriverManager.getConnection(dtbase,user,pass);
    Statement st = con.createStatement();
    ResultSet res = st.executeQuery(SQL);
    if (res != null) {
    DefaultTableModel model = (DefaultTableModel) jTable1.getModel();
    for (int i=tableModel.getRowCount()-1; i>=0; i--) {
        model.removeRow(i);
    }
    try {
    while(res.next()){
    data[0] = res.getString(1);
    data[1] = res.getString(2);
    tableModel.addRow(data);
    }
    } catch (Exception ex) {
    JOptionPane.showMessageDialog(null, "Connection database error");
}

```

```

    }
    }}

public void bersih() {
    jTextField2.setText("");
    jTextField2.requestFocus();
}

int row1 = 0;

public void Tampil() {
    row1 = jTable1.getSelectedRow();
    jTextField1.setText(tableModel.getValueAt(row1, 0).toString());
    jTextField2.setText(tableModel.getValueAt(row1, 1).toString());
}

/** 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.
 */

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

    jScrollPane1 = new javax.swing.JScrollPane();
    jTable1 = new javax.swing.JTable();
    jLabel1 = new javax.swing.JLabel();
    jLabel2 = new javax.swing.JLabel();
    jTextField1 = new javax.swing.JTextField();
    jTextField2 = new javax.swing.JTextField();
    jButton1 = new javax.swing.JButton();
    jButton2 = new javax.swing.JButton();
    jButton3 = new javax.swing.JButton();
    jButton4 = new javax.swing.JButton();
    jButton5 = new javax.swing.JButton();
}

```

```
setBackground(new java.awt.Color(153, 153, 255));
```

```
setBorder(null);
```

```
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.setViewportView(jTable1);
```

```
jLabel1.setText("KODE ALTERNATIVE");
```

```
jLabel2.setText("NAMA ALTERNATIVE");
```

```
jButton1.setText("BERSIH");
```

```
jButton1.addActionListener(new java.awt.event.ActionListener() {
```

```
    public void actionPerformed(java.awt.event.ActionEvent evt) {
```

```
        jButton1ActionPerformed(evt);
```

```
    }
```

```
});
```

```
jButton2.setText("SIMPAN");
```

```
jButton2.addActionListener(new java.awt.event.ActionListener() {
```

```
    public void actionPerformed(java.awt.event.ActionEvent evt) {
```

```
        jButton2ActionPerformed(evt);
```

```
    }
```



```
.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
```

```
    .addComponent(jScrollPane1,  
javax.swing.GroupLayout.DEFAULT_SIZE, 733, Short.MAX_VALUE)
```

```
    .addGroup(layout.createSequentialGroup()  
.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
```

```
    .addComponent(jLabel1)  
    .addComponent(jLabel2))
```

```
.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)  
.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)
```

```
    .addComponent(jTextField1)  
    .addComponent(jTextField2)  
    .addGroup(layout.createSequentialGroup()  
        .addComponent(jButton1,  
javax.swing.GroupLayout.PREFERRED_SIZE, 83,  
javax.swing.GroupLayout.PREFERRED_SIZE)
```

```
.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)  
    .addComponent(jButton2)))
```

```
.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)  
    .addComponent(jButton3,  
javax.swing.GroupLayout.PREFERRED_SIZE, 71,  
javax.swing.GroupLayout.PREFERRED_SIZE)
```

```
.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)  
    .addComponent(jButton4)
```

```
.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
    .addComponent(jButton5))
    .addContainerGap()
);
layout.setVerticalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(layout.createSequentialGroup()
        .addGap(25, 25, 25)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BAS
ELINE)
    .addComponent(jLabel1)
    .addComponent(jTextField1,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRA
ILING)
    .addComponent(jLabel2)
    .addComponent(jTextField2,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))
    .addGap(29, 29, 29)
```

```
.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)
```

```
.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
```

```
    .addComponent(jButton2,  
        javax.swing.GroupLayout.PREFERRED_SIZE, 45,  
        javax.swing.GroupLayout.PREFERRED_SIZE)
```

```
    .addComponent(jButton1,  
        javax.swing.GroupLayout.PREFERRED_SIZE, 46,  
        javax.swing.GroupLayout.PREFERRED_SIZE))
```

```
.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
```

```
    .addComponent(jButton4,  
        javax.swing.GroupLayout.PREFERRED_SIZE, 45,  
        javax.swing.GroupLayout.PREFERRED_SIZE)
```

```
    .addComponent(jButton5,  
        javax.swing.GroupLayout.PREFERRED_SIZE, 45,  
        javax.swing.GroupLayout.PREFERRED_SIZE)
```

```
    .addComponent(jButton3,  
        javax.swing.GroupLayout.PREFERRED_SIZE, 45,  
        javax.swing.GroupLayout.PREFERRED_SIZE)))
```

```
.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
```

```
    .addComponent(jScrollPane1,  
        javax.swing.GroupLayout.DEFAULT_SIZE, 214, Short.MAX_VALUE)
```

```
    .addContainerGap())
```

```
);
```

```

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

private void SimpanActionPerformed(java.awt.event.ActionEvent evt) {
    String s_1 = jTextField1.getText();
    String s_2 = jTextField2.getText();
    //validasi isian
    if (s_1.isEmpty()){
        JOptionPane.showMessageDialog(null,"Kode Masih Kosong");
        jTextField1.requestFocus();
        return;
    }
    if (s_2.isEmpty()){
        JOptionPane.showMessageDialog(null,"Nama Masih Kosong");
        jTextField2.requestFocus();
        return;
    }
    try{
        Connection kon = DriverManager.getConnection(dtbase,user,pass);
        Statement stt = kon.createStatement();
        String sql = "INSERT INTO t_alternative VALUES("
            + s_1 + ","
            + s_2 + ")";
        int done = stt.executeUpdate(sql);
        data[0] = jTextField1.getText();
        data[1] = jTextField2.getText();
        tableModel.insertRow(0, data);
        bersih();
        refresh();
        AutoNumber();
    }catch(Exception ex){

```

```

        JOptionPane.showMessageDialog(null,"Error Simpan");
    }
}

private void UpdateActionPerformed(java.awt.event.ActionEvent evt) {
    String s_1 = jTextField1.getText();
    String s_2 = jTextField2.getText();
    //validasi isian
    if (s_1.isEmpty()){
        JOptionPane.showMessageDialog(null,"Kode Masih Kosong");
        jTextField1.requestFocus();
        return;
    }
    if (s_2.isEmpty()){
        JOptionPane.showMessageDialog(null,"Nama Masih Kosong");
        jTextField2.requestFocus();
        return;
    }
    try{
        Connection kon = DriverManager.getConnection(dtbase,user,pass);
        Statement stt = kon.createStatement();
        String sql = "UPDATE t_alternative SET nm_alternative="
            + s_2 + " WHERE kd_alternatif="
            + s_1 + """;
        int done = stt.executeUpdate(sql);
        data[0] = jTextField1.getText();
        data[1] = jTextField2.getText();
        tableModel.removeRow(0);
        tableModel.insertRow(0, data);
        refresh();
        AutoNumber();
    }
}

```

```

        bersih();
    }catch(Exception ex){
        JOptionPane.showMessageDialog(null,"Error Update");
    } // TODO add your handling code here:
}

private void DeleteActionPerformed(java.awt.event.ActionEvent evt) {
    String s_1 = jTextField1.getText();
    //validasi isian
    if (s_1.isEmpty()){
        JOptionPane.showMessageDialog(null,"Kode Masih Kosong");
        jTextField1.requestFocus();
        return;
    }
    try{
        Connection kon = DriverManager.getConnection(dtbase,user,pass);
        Statement stt = kon.createStatement();
        int ok=JOptionPane.showConfirmDialog(rootPane, "Apakah Anda yakin
Akan Menghapus Data Tersebut???", "Question",JOptionPane.OK_OPTION);
        if (ok==0) {
            String sql = "DELETE FROM t_alternative WHERE
kd_alternatif='"+s_1+"'";
            int done = stt.executeUpdate(sql);
            refresh();
            AutoNumber();
            bersih();
            JOptionPane.showMessageDialog(this,
                "Data Berhasil Dihapus", "Informasi", 1);
        }
    }catch(Exception ex){
        JOptionPane.showMessageDialog(null, "Data Tidak Dapat Dihapus!");
    }
}

```

```
    }  
}  
  
private void jButton5ActionPerformed(java.awt.event.ActionEvent evt) {  
    this.dispose();  
}  
  
private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {  
    // TODO add your handling code here:  
}  
  
// Variables declaration - do not modify  
private javax.swing.JButton jButton1;  
private javax.swing.JButton jButton2;  
private javax.swing.JButton jButton3;  
private javax.swing.JButton jButton4;  
private javax.swing.JButton jButton5;  
private javax.swing.JLabel jLabel1;  
private javax.swing.JLabel jLabel2;  
private javax.swing.JScrollPane jScrollPane1;  
private javax.swing.JTable jTable1;  
private javax.swing.JTextField jTextField1;  
private javax.swing.JTextField jTextField2;  
// End of variables declaration  
  
}
```