

LAMPIRAN

KODING PROGRAM

```

package recordingsapi;
import javax.swing.ImageIcon;
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.koneksidb;
/**
 *
 * @author User
 */
public class form_sapi extends javax.swing.JInternalFrame {
String data[]=new String[7];
koneksidb dbsetting;
String driver, dtbase, user,pass,userLogin;
private Connection con;
private Statement st;
String Tanggal1;
ImageIcon imageIcon;
/** Creates new form form_sapi */
public form_sapi() {
initComponents();
dbsetting = new koneksi.koneksidb();
driver = dbsetting.SettingPanel("DBDriver");
dtbase = dbsetting.SettingPanel("DBDatabase");
user = dbsetting.SettingPanel("DBUsername");

```

```

pass = dbsetting.SettingPanel("DBPassword");
AutoNumber();
jTable1.setModel(tableModel);
Table(jTable1, new int[] {100,200,100,100,100,100,100});
setDefaultTable();
tampil_jens();
tampil_kandang();

Dimension screenSize = Toolkit.getDefaultToolkit().getScreenSize();
Dimension frameSize = getSize();
setLocation(
(screenSize.width - frameSize.width) / 2,
(screenSize.height - frameSize.height) / 15);
}

public void AutoNumber() {
    try {
        Class.forName(driver);
        Connection kon = DriverManager.getConnection(dtbase, user, pass);
        Statement stt = kon.createStatement();
        String sql = "SELECT * FROM tb_sapi ORDER BY id_sapi DESC";
        ResultSet rs = stt.executeQuery(sql);
        if (rs.next()) {
            String kd_barang = rs.getString("id_sapi").substring(1);
            String AN = "" + (Integer.parseInt(kd_barang) + 1);
            String Nol = "";
            if(AN.length()==1)
                {Nol = "000";}
            else if(AN.length()==2)
                {Nol = "00";}
            else if(AN.length()==3)
                {Nol = "0";}
            else if(AN.length()==4)
                {Nol = "";}
        }
    }
}

```

```

        jTextField1.setText("S"+Nol + AN);//sesuaikan dengan variable
namenya
    } else {
        jTextField1.setText("S0001");//sesuaikan dengan variable namenya
    }

    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 Sapi","Nama Sapi","Gender","Berat Awal","Berat
Target","KD Jenis","KD Kandang"}
        ){
            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 tb_sapi order by id_sapi Asc";
        ResultSet res = st.executeQuery(SQL);
        while(res.next()){
            data[0] = res.getString(1);
            data[1] = res.getString(2);
            data[2] = res.getString(3);
            data[3] = res.getString(4);
            data[4] = res.getString(5);
            data[5] = res.getString(6);
            data[6] = res.getString(7);
            tableModel.addRow(data);
        }
    } catch (Exception ex) {
        JOptionPane.showMessageDialog(null, "Koneksi database error");
    }
}

public void refresh() throws SQLException{
    String SQL = "select * from tb_sapi order by id_sapi 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);
data[2] = res.getString(3);
data[3] = res.getString(4);
data[4] = res.getString(5);
data[5] = res.getString(6);
data[6] = res.getString(7);
tableModel.addRow(data);
}
} catch (Exception ex) {
JOptionPane.showMessageDialog(null, "Connection database error");
}
}}

public void bersih(){
jTextField1.setText("");
jTextField2.setText("");
jTextField3.setText("");
jTextField4.setText("");
jTextField5.setText("");
jTextField6.setText("");
jTextField4.setText("");
jComboBox1.setSelectedItem("-Pilih Jenis-");
jComboBox2.setSelectedItem("-Pilih Kandang-");
jRadioButton1.setSelected(false);

```

```

jRadioButton2.setSelected(false);
jRadioButton1.setEnabled(true);
jRadioButton2.setEnabled(true);
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());
    if (tableModel.getValueAt(row,2).toString().equals("Jantan")) {
        jRadioButton1.setSelected(true);
        jRadioButton2.setSelected(false);
        jRadioButton2.setEnabled(false);
    }else{
        jRadioButton1.setSelected(false);
        jRadioButton2.setSelected(false);
        jRadioButton2.setEnabled(true);
    }
    jTextField3.setText(tableModel.getValueAt(row1, 3).toString());
    jTextField4.setText(tableModel.getValueAt(row1, 4).toString());
    jTextField5.setText(tableModel.getValueAt(row1, 5).toString());
    jTextField6.setText(tableModel.getValueAt(row1, 6).toString());
}

public void simpan_data(){
    String s_1 = jTextField1.getText();
    String s_2 = jTextField2.getText();
    String s_3 = "";
    String s_4 = jTextField3.getText();
    String s_5 = jTextField4.getText();
    String s_6 = jTextField5.getText();
}

```

```
String s_7 = jTextField6.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;
}
if (jRadioButton1.isSelected()){
    s_3= "Jantan";
}
else if (jRadioButton2.isSelected()){
    s_3 = "Betina";
}
if (s_4.isEmpty()){
    JOptionPane.showMessageDialog(null,"Bobot Awal Masih Kosong");
    jTextField3.requestFocus();
    return;
}
if (s_5.isEmpty()){
    JOptionPane.showMessageDialog(null,"Bobot Target Masih Kosong");
    jTextField4.requestFocus();
    return;
}
if (s_6.isEmpty()){
    JOptionPane.showMessageDialog(null,"KD Jenis Masih Kosong");
    jTextField5.requestFocus();
    return;
}
```

```

    }
    if (s_7.isEmpty()){
        JOptionPane.showMessageDialog(null,"Kode Kandang Masih Kosong");
        jTextField6.requestFocus();
        return;
    }
    try{
        Connection kon = DriverManager.getConnection(dtbase,user,pass);
        Statement stt = kon.createStatement();
        String sql = "INSERT INTO tb_sapi VALUES("
            + s_1 + ","
            + s_2 + ","
            + s_3 + ","
            + s_4 + ","
            + s_5 + ","
            + s_6 + ","
            + s_7 + ")";
        int done = stt.executeUpdate(sql);
        data[0] = jTextField1.getText();
        data[1] = jTextField2.getText();
        data[2] = s_3;
        data[3] = jTextField3.getText();
        data[4] = jTextField4.getText();
        data[5] = jTextField5.getText();
        data[6] = jTextField6.getText();
        tableModel.insertRow(0, data);
        JOptionPane.showMessageDialog(null, "Data BERHASIL tersimpan" ,
        "Informasi", JOptionPane.INFORMATION_MESSAGE);
        bersih();
        refresh();
        AutoNumber();
    }catch(Exception ex){

```

```
        JOptionPane.showMessageDialog(null, "Data GAGAL tersimpan" ,
        "Informasi", JOptionPane.INFORMATION_MESSAGE);
    }
}
```

```
public void ubah_data(){
    String s_1 = jTextField1.getText();
    String s_2 = jTextField2.getText();
    String s_3 = "";
    String s_4 = jTextField3.getText();
    String s_5 = jTextField4.getText();
    String s_6 = jTextField5.getText();
    String s_7 = jTextField6.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;
    }
    if (jRadioButton1.isSelected()){
        s_3= "Jantan";
    }
    else if (jRadioButton2.isSelected()){
        s_3 = "Betina";
    }
    if (s_4.isEmpty()){
        JOptionPane.showMessageDialog(null,"Bobot Awal Masih Kosong");
```

```

        jTextField3.requestFocus();
        return;
    }
    if (s_5.isEmpty()){
        JOptionPane.showMessageDialog(null,"Bobot Target Masih Kosong");
        jTextField4.requestFocus();
        return;
    }
    if (s_6.isEmpty()){
        JOptionPane.showMessageDialog(null,"KD Jenis Masih Kosong");
        jTextField5.requestFocus();
        return;
    }
    if (s_7.isEmpty()){
        JOptionPane.showMessageDialog(null,"Kode Kandang Masih Kosong");
        jTextField6.requestFocus();
        return;
    }
    try{
        Connection kon = DriverManager.getConnection(dtbase,user,pass);
        Statement stt = kon.createStatement();
        String sql = "UPDATE tb_sapi SET nama="
            + s_2 + ",gander="
            + s_3 + ",berat_awal="
            + s_4 + ",berat_target="
            + s_5 + ",kd_jns="
            + s_6 + ",kd_kandang="
            + s_7 + " WHERE id_sapi="
            + s_1 + """;
        int done = stt.executeUpdate(sql);
        data[0] = jTextField1.getText();
        data[1] = jTextField2.getText();
    }
}

```

```

        data[2] = s_3;
        data[3] = jTextField3.getText();
        data[4] = jTextField4.getText();
        data[5] = jTextField5.getText();
        data[6] = jTextField6.getText();
        tableModel.removeRow(0);
        tableModel.insertRow(0, data);
        JOptionPane.showMessageDialog(null, "Data BERHASIL Diubah" ,
        "Informasi", JOptionPane.INFORMATION_MESSAGE);
        bersih();
        refresh();
        AutoNumber();
    }catch(Exception ex){
        JOptionPane.showMessageDialog(null, "Data GAGAL Diubah" ,
        "Informasi", JOptionPane.INFORMATION_MESSAGE);
    }
}

public void hapus_data(){
    try{
        Connection kon = DriverManager.getConnection(dtbase,user,pass);
        Statement stt = kon.createStatement();
        int ok=JOptionPane.showConfirmDialog(rootPane, "Apakah Anda yakin
        Akan Menghapus Data Ini???", "Question",JOptionPane.OK_OPTION);
        if (ok==0) {
            String sql ="DELETE FROM tb_sapi WHERE
            id_sapi="+jTextField1.getText()+"";
            int done = stt.executeUpdate(sql);
            bersih();
            refresh();
            JOptionPane.showMessageDialog(this,
            "Data Berhasil Dihapus", "Informasi", 1);
            AutoNumber();
        }
    }
}

```

```

    }
    }catch(Exception ex){
        JOptionPane.showMessageDialog(null, "Data Tidak Dapat Dihapus!");
    }
}

public void tampil_jens(){
try {
    Class.forName(driver);
    Connection kon = DriverManager.getConnection(dtbase,user,pass);
    Statement stt = kon.createStatement();
    String SQL = "SELECT * FROM tb_jenis ORDER BY nm_jns ASC";
    ResultSet res = stt.executeQuery(SQL);
    while(res.next()){
        jComboBox1.addItem(res.getString("nm_jns"));
    }
} catch (Exception e) {
    JOptionPane.showMessageDialog(null, "Error Connection"+e);
}
}

public void tampil_kandang(){
try {
    Class.forName(driver);
    Connection kon = DriverManager.getConnection(dtbase,user,pass);
    Statement stt = kon.createStatement();
    String SQL = "SELECT * FROM tb_kandang ORDER BY nama_kandang
ASC";
    ResultSet res = stt.executeQuery(SQL);
    while(res.next()){
        jComboBox2.addItem(res.getString("nama_kandang"));
    }
} catch (Exception e) {
    JOptionPane.showMessageDialog(null, "Error Connection"+e);
}
}

```

```

    }
}

public void hsltampljens(){
    try{
        String status="";
        Connection kon = DriverManager.getConnection(dtbase,user,pass);
        Statement stt = kon.createStatement();
        String sql = "SELECT * FROM tb_jenis where nm_jns="
        "+jComboBox1.getSelectedItem()+""";
        ResultSet rsProdi = stt.executeQuery(sql);
        if(rsProdi.next()){
            jTextField5.setText(rsProdi.getString("kd_jns"));
        }catch(Exception ex){
        }
    }
}

public void hsltamplkandang(){
    try{
        String status="";
        Connection kon = DriverManager.getConnection(dtbase,user,pass);
        Statement stt = kon.createStatement();
        String sql = "SELECT * FROM tb_kandang where nama_kandang="
        "+jComboBox2.getSelectedItem()+""";
        ResultSet rsProdi = stt.executeQuery(sql);
        if(rsProdi.next()){
            jTextField6.setText(rsProdi.getString("kd_kandang"));
        }catch(Exception ex){
        }
    }
}

public void hsltamplnamakandang(){
    try{
        String status="";
        Connection kon = DriverManager.getConnection(dtbase,user,pass);

```

```

Statement stt = kon.createStatement();
String sql = "SELECT * FROM tb_kandang where kd_kandang=
"+jTextField6.getText()+"";
ResultSet rsProdi = stt.executeQuery(sql);
if(rsProdi.next()){
jComboBox2.setSelectedItem(rsProdi.getString("nama_kandang"));
}}catch(Exception ex){
}
}
public void hsltamplnamajens(){
try{
String status="";
Connection kon = DriverManager.getConnection(dtbase,user,pass);
Statement stt = kon.createStatement();
String sql = "SELECT * FROM tb_jenis where kd_jns= "+
jTextField5.getText()+"";
ResultSet rsProdi = stt.executeQuery(sql);
if(rsProdi.next()){
jComboBox1.setSelectedItem(rsProdi.getString("nm_jns"));
}}catch(Exception ex){
}
}
private void jRadioButton1ActionPerformed(java.awt.event.ActionEvent evt) {
if (jRadioButton1.isSelected()== true){
jRadioButton2.setSelected(false);
jRadioButton2.setEnabled(false);
}else if (jRadioButton1.isSelected()== false){
jRadioButton2.setSelected(false);
jRadioButton2.setEnabled(true);
}
}
}

```

```
private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {  
    if (jRadioButton2.isSelected()== true){  
        jButton1.setSelected(false);  
        jButton1.setEnabled(false);  
    }else if (jRadioButton2.isSelected()== false){  
        jButton1.setSelected(false);  
        jButton1.setEnabled(true);  
    }  
}  
  
private void jMenuItem1StateChanged(java.awt.event.ItemEvent evt) {  
    hsltampljens();  
}  
  
private void jMenuItem2StateChanged(java.awt.event.ItemEvent evt) {  
    hsltamplkandang();  
}  
  
private void jTextField5CaretUpdate(javax.swing.event.CaretEvent evt) {  
    hsltamplnamajens();  
}  
  
private void jTextField6CaretUpdate(javax.swing.event.CaretEvent evt) {  
    hsltamplnamakandang();  
}  
  
private void jButton4ActionPerformed(java.awt.event.ActionEvent evt) {  
    this.dispose();  
}  
  
private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {  
    simpan_data();  
}
```

```

    }

    private void button2ActionPerformed(java.awt.event.ActionEvent evt) {
ubah_data();
    }

    private void button3ActionPerformed(java.awt.event.ActionEvent evt) {
hapus_data();
    }

    private void jTable1MouseClicked(java.awt.event.MouseEvent evt) {
        if (evt.getClickCount()==1) {
            Tampil();
        }
    }

    private void button5ActionPerformed(java.awt.event.ActionEvent evt) {
bersih();
AutoNumber();
    }

    private void jTextField3KeyTyped(java.awt.event.KeyEvent evt) {
        if(jTextField3.getText().length(>4)
            evt.consume();
            FilterHanyaAngka(evt);    // TODO add your handling code here:
    }

    private void jTextField4KeyTyped(java.awt.event.KeyEvent evt) {
        if(jTextField4.getText().length(>4)
            evt.consume();
            FilterHanyaAngka(evt);    // TODO add your handling code here:
    }

```

```

private void jTextField2KeyTyped(java.awt.event.KeyEvent evt) {
    if(jTextField2.getText().length()>24)
        evt.consume();    // TODO add your handling code here:
    }
void filterhanyahurut (KeyEvent b){
    if (Character.isDigit(b.getKeyChar())) {
        b.consume();
        JOptionPane.showMessageDialog(null, "Maaf Data Yang Anda Inputkan Harus
Huruf");
    }
}
public void FilterHanyaAngka(java.awt.event.KeyEvent evt){
    char c = evt.getKeyChar();
    if (!( (c>='0' && c<='9' || (c == KeyEvent.VK_BACK_SPACE) || (c ==
KeyEvent.VK_DELETE)))){
        evt.consume();
        JOptionPane.showMessageDialog(null, "Maaf Data Yang Anda Inputkan
Harus Angka");
    }
}
}
}
}

```