



REPUBLIK INDONESIA
KEMENTERIAN HUKUM DAN HAK ASASI MANUSIA

SURAT PENCATATAN CIPTAAN

Dalam rangka perlindungan ciptaan di bidang ilmu pengetahuan, seni dan sastra berdasarkan Undang-Undang Nomor 28 Tahun 2014 tentang Hak Cipta, dengan ini menerangkan:

Nomor dan tanggal permohonan : EC00202129754, 24 Juni 2021

Pencipta

Nama : **Wina Yusnaeni, M.Kom, Ratih Yulia Hayuningtyas, M.Kom dkk**
Alamat : Kaliabang Tengah No. 53 Rt. 003 Rw. 001 Kelurahan Kaliabang Tengah
Kecamatan Bekasi Utara, Bekasi, JAWA BARAT, 17125
Kewarganegaraan : Indonesia

Pemegang Hak Cipta

Nama : **Wina Yusnaeni, M.Kom, Ratih Yulia Hayuningtyas, M.Kom dkk**
Alamat : Kaliabang Tengah No. 53 Rt. 003 Rw. 001 Kelurahan Kaliabang Tengah
Kecamatan Bekasi Utara, Bekasi, JAWA BARAT, 17125
Kewarganegaraan : Indonesia

Jenis Ciptaan : **Program Komputer**
Judul Ciptaan : **Sistem Informasi Penilaian Kinerja Karyawan (SI-PEKKA)
Menggunakan Metode WP**

Tanggal dan tempat diumumkan untuk pertama kali : 1 Juni 2021, di Jakarta
di wilayah Indonesia atau di luar wilayah Indonesia

Jangka waktu perlindungan : Berlaku selama 50 (lima puluh) tahun sejak Ciptaan tersebut pertama kali
dilakukan Pengumuman.

Nomor pencatatan : 000256189

adalah benar berdasarkan keterangan yang diberikan oleh Pemohon.

Surat Pencatatan Hak Cipta atau produk Hak terkait ini sesuai dengan Pasal 72 Undang-Undang Nomor 28 Tahun 2014 tentang Hak Cipta.

a.n. MENTERI HUKUM DAN HAK ASASI MANUSIA
DIREKTUR JENDERAL KEKAYAAN INTELEKTUAL



Dr. Freddy Harris, S.H., LL.M., ACCS.
NIP. 196611181994031001

Disclaimer:

Dalam hal pemohon memberikan keterangan tidak sesuai dengan surat pernyataan, menteri berwenang untuk mencabut surat pencatatan permohonan.

LAMPIRAN PENCIPTA

No	Nama	Alamat
1	Wina Yusnaeni, M.Kom	Kaliabang Tengah No. 53 Rt. 003 Rw. 001 Kelurahan Kaliabang Tengah Kecamatan Bekasi Utara
2	Ratih Yulia Hayuningtyas, M.Kom	Perum. Pejuang Pratama Blok No. 21 RT 005/RW 006, Kelurahan Pejuang, Kecamatan Medan Satria
3	Retno Sari, M.Kom	Jalan Talang Ujung Rt. 002 Rw. 03 No. 25 Kelurahan Pegangsaan Kecamatan Menteng
4	Widiarina, M.Kom	Komplek Depsos Blok E2 No. 4 Rt. 013 Rw. 008 Kelurahan Telagaasih Kecamatan Cikarang Barat
5	Puji Astuti, M.Kom	Jalan Akasia Rt. 002 Rw.018 No. 9 Kelurahan Pamulang Timur, Kecamatan Pamulang
6	Nia Nuraeni, M.Kom	Komp DDN Blok B No. 22 Rt. 003 Rw. 010 Kelurahan Jatiwaringin Kecamatan PondokGede

LAMPIRAN PEMEGANG

No	Nama	Alamat
1	Wina Yusnaeni, M.Kom	Kaliabang Tengah No. 53 Rt. 003 Rw. 001 Kelurahan Kaliabang Tengah Kecamatan Bekasi Utara
2	Ratih Yulia Hayuningtyas, M.Kom	Perum. Pejuang Pratama Blok No. 21 RT 005/RW 006, Kelurahan Pejuang, Kecamatan Medan Satria
3	Retno Sari, M.Kom	Jalan Talang Ujung Rt. 002 Rw. 03 No. 25 Kelurahan Pegangsaan Kecamatan Menteng
4	Widiarina, M.Kom	Komplek Depsos Blok E2 No. 4 Rt. 013 Rw. 008 Kelurahan Telagaasih Kecamatan Cikarang Barat
5	Puji Astuti, M.Kom	Jalan Akasia Rt. 002 Rw.018 No. 9 Kelurahan Pamulang Timur, Kecamatan Pamulang
6	Nia Nuraeni, M.Kom	Komp DDN Blok B No. 22 Rt. 003 Rw. 010 Kelurahan Jatiwaringin Kecamatan PondokGede



SISTEM INFORMASI PENILAIAN KINERJA KARYAWAN (SI-PEKKA) MENGUNAKAN METODE WP



Wina Yusnaeni, M.Kom
Ratih Yulia Hayuningtyas, M.Kom
Retno Sari, M.Kom
Widiarina, M.Kom
Puji Astuti, M.Kom
Nia Nuraeni, M.Kom

KATA PENGANTAR

Puji dan Syukur kami panjatkan kepada Allah SWT yang telah memberikan nikmat dan rasa syukur sehingga kami dapat menyelesaikan dengan baik. **Aplikasi Penentuan Peringkat Kinerja Karyawan dengan Metode WP** merupakan aplikasi yang dibuat untuk menentukan peringkat kinerja karyawan.

Modul ini dibuat sebagai panduan atau petunjuk dalam pengoperasian **Aplikasi Penentuan Peringkat Kinerja Karyawan dengan Metode WP**. Dengan adanya modul ini diharapkan dapat membantu pengguna dalam menentukan peringkat kinerja karyawan. Dalam pembuatan modul ini tentunya tidak lepas dari kesalahan dan kekurangan, maka dari itu kami memohon maaf yang sebesar-besarnya dan kedepannya menjadi lebih baik.

Jakarta, 21 Juni 2021

Wina Yusnaeni, M.Kom

BAB I

MENU LOGIN DAN MENU UTAMA

A. Menu Login

Menu login merupakan awal proses untuk masuk ke dalam menu utama, dengan cara memasukkan kode user dan password. Login ini memiliki tujuan untuk keamanan data sehingga yang dapat mengakses hanya user yang memiliki hak akses. Berikut adalah tampilan menu login.



Gambar 1. Menu Login

Source Code Koneksi

```
package spk_new;
import java.sql.*;
import javax.swing.JOptionPane;
public class koneksi {
    Connection conn;
    Statement st;
    ResultSet rs;
    public Connection setKoneksi() {
    try{
        Class.forName("com.mysql.jdbc.Driver");
        conn=DriverManager.getConnection("jdbc:mysql://localhost/spkku","root","");
        st=conn.createStatement();
    }
    catch(Exception e){
        JOptionPane.showMessageDialog(null,"Koneksi Gagal : " +e);
    }
}
```

```
} return conn; } }
```

Source Code Login

```
package spk_new;
import java.awt.event.KeyEvent;
import java.sql.*;
import javax.swing.JFrame;
import javax.swing.JOptionPane;
import java.awt.Dimension;
import java.awt.Toolkit;
public class login extends javax.swing.JFrame {
    koneksi kon=new koneksi();
    public MenuUtama menu=null;
    public login() {
        initComponents();
        Dimension layar=Toolkit.getDefaultToolkit().getScreenSize();
        int x=layar.width/2-this.getSize().width/2;
        int y=layar.height/2 - this.getSize().height/2;
        this.setLocation(x,y);
        kon.setKoneksi(); }
    void bersih(){
        kduser.setText("");
        password.setText("");}
    private void proseslogin(){
        String kodeuser="";
        String pass="";
        kodeuser=kduser.getText();
        try{String    sql="Select    kduser,nmuser,password    from    user    where
        kduser='"+kodeuser+"' AND password='"+password.getText()+"'";
        kon.rs=kon.st.executeQuery(sql);
        if(kon.rs.next()){
            MenuUtama menu = new MenuUtama();
            menu.setLocationRelativeTo(null);
            menu.setVisible(true);
```

```

        this.dispose(); }
    else{
        JOptionPane.showMessageDialog(null, "PASSWORD SALAH");
        bersih();
        kduser.setEnabled(true);
        kduser.requestFocus(); }
} catch(SQLException e){
    JOptionPane.showMessageDialog(null, e); } }

private void bbatalActionPerformed(java.awt.event.ActionEvent evt) {
    dispose(); }

private void formWindowActivated(java.awt.event.WindowEvent evt) {
    password.setEnabled(false);
    kduser.requestFocus();
    blogin.setEnabled(false); }

private void bloginActionPerformed(java.awt.event.ActionEvent evt) {
    proseslogin(); }

private void kduserKeyPressed(java.awt.event.KeyEvent evt) {
    if(evt.getKeyCode()==KeyEvent.VK_ENTER){
        String kodeuser = "";
        kodeuser = kduser.getText();
        try {
            String sql = "Select *from user where kduser='" + kodeuser + "'";
            kon.rs = kon.st.executeQuery(sql);
            if (kon.rs.next()) {
                kduser.setEnabled(false);
                password.setEnabled(true);
                password.requestFocus();
                blogin.setEnabled(true);
                kduser.setEnabled(false);
            } else {
                JOptionPane.showMessageDialog(null, "kode user salah");
                bersih();
                kduser.setEnabled(true);
            }
        }
    }
}

```

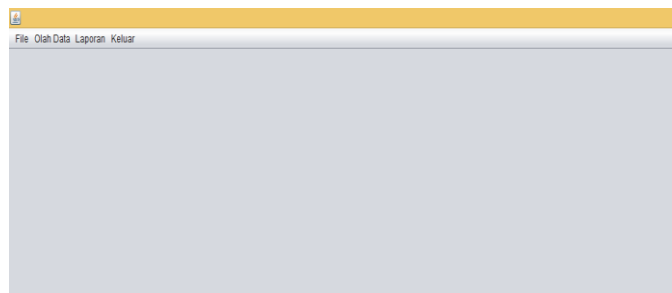
```

        kduser.requestFocus(); }
    } catch (SQLException e) {
        JOptionPane.showMessageDialog(null, e); } } }

```

B. Menu Utama

Menu utama berisikan menu-menu yang akan digunakan dalam aplikasi ini. Terdapat beberapa menu dalam form menu utama yaitu menu file yang terdiri dari menu alternative, kriteria, range kriteria dan user. Pada menu olah data terdiri dari menu penilaian SPK dan Hasil Olahan. Menu Laporan terdiri dari laporan hasil penilaian dan laporan range kriteria, sedangkan menu keluar berisikan logout yang akan kembali ke form login dan menu keluar aplikasi yang akan menutup aplikasi ini. Berikut adalah gambar dari menu utama.



Gambar 2. Menu Utama

Source Code Menu Utama

```

package spk_new;
import java.io.File;
import java.util.HashMap;
import javax.swing.JOptionPane;
import net.sf.jasperreports.engine.JasperFillManager;
import net.sf.jasperreports.engine.JasperPrint;
import net.sf.jasperreports.engine.JasperReport;
import net.sf.jasperreports.engine.util.JRLoader;
import net.sf.jasperreports.view.JasperViewer;
import java.sql.*;
import javax.swing.*;

public class MenuUtama extends javax.swing.JFrame {
    koneksi kon=new koneksi();
    public MenuUtama() {

```

```

        initComponents();
        kon.setKoneksi(); }

private void jMenuItem1ActionPerformed(java.awt.event.ActionEvent evt) {
    range_kriteria range = new range_kriteria();
    range.setLocationRelativeTo(null);
    range.setVisible(true); }

private void mnkriteriaActionPerformed(java.awt.event.ActionEvent evt) {
    Kriteria kr= new Kriteria();
    kr.setLocationRelativeTo(null);
    kr.setVisible(true); }

private void mnpenilaianActionPerformed(java.awt.event.ActionEvent evt) {
    penilaianspk nilai = new penilaianspk();
    nilai.setLocationRelativeTo(null);
    nilai.setVisible(true); }

private void mnolahActionPerformed(java.awt.event.ActionEvent evt) {
    hasil HS = new hasil();
    HS.setLocationRelativeTo(null);
    HS.setVisible(true); }

private void mnalternatifActionPerformed(java.awt.event.ActionEvent evt) {
    alternative alt = new alternative();
    alt.setLocationRelativeTo(null);
    alt.setVisible(true); }

private void mnkeluarActionPerformed(java.awt.event.ActionEvent evt) {
    System.exit(0); }

private void formWindowOpened(java.awt.event.WindowEvent evt) {
    setExtendedState(MenuUtama.MAXIMIZED_BOTH); }

private void mnlogoutActionPerformed(java.awt.event.ActionEvent evt) {
    this.dispose();
    login l=new login();
    l.setLocationRelativeTo(null);
    l.setVisible(true); }

private void jMenuItem3ActionPerformed(java.awt.event.ActionEvent evt) {
    user usr = new user();

```

```

        usr.setLocationRelativeTo(null);
        usr.setVisible(true); }

private void mnlapshlActionPerformed(java.awt.event.ActionEvent evt) {
    try{
        HashMap parameter=new HashMap();
        File file=new File ("src/spk_new/report1.jasper");
        JasperReport jr=(JasperReport) JRLoader.loadObject(file);
        JasperPrint jp=JasperFillManager.fillReport(jr, parameter,kon.setKoneksi());
        JasperViewer.viewReport(jp,false);
        JasperViewer.setDefaultLookAndFeelDecorated(true); }
    catch(Exception e){
        JOptionPane.showMessageDialog(rootPane, "Gagal ! Error : "+e);    } }

private void jMenuItem2ActionPerformed(java.awt.event.ActionEvent evt) {
    try{
        String path="src/SPK_new/report2.jasper";
        HashMap parameter = new HashMap();
        JasperPrint print = JasperFillManager.fillReport(path,parameter, kon.setKoneksi());
        JasperViewer.viewReport(print, false); }
    catch(Exception e){
        JOptionPane.showMessageDialog(rootPane, "Gagal ! Error : "+e);    } }

```

BAB II

MENU FILE

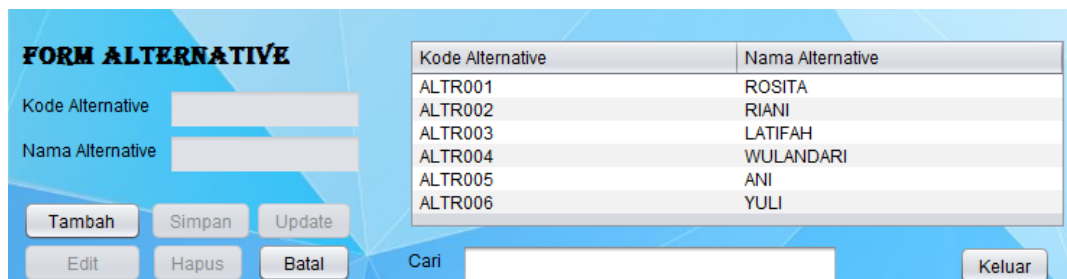
A. Form Alternative

Form alternative ini berisikan kode alternative dan nama alternative yang akan dinilai peringkat kinerjanya. Pada saat form ini di running textbox kode dan nama alternative dalam keadaan nonaktif dan yang aktif button tambah, batal, cari dan keluar.

Textbox cari digunakan untuk mencari kode dan nama alternative yang ada dengan cara mengisikan nama alternative di textbox maka tabel akan tampil nama yang dicari. Kemudian double klik nama alternative maka akan tampil datanya di dalam form alternative. Button edit digunakan untuk mengedit data yang tampil di form alternative dan dapat di update datanya dengan button update, selain itu datanya juga dapat di hapus dengan button hapus.

Button tambah digunakan untuk menambahkan data alternative dengan mengisi kode serta nama alternative, untuk menyimpannya dengan button simpan.

Button keluar digunakan untuk menutup form alternative dan kembali ke menu utama. Berikut adalah tampilan dari form alternative.



Kode Alternative	Nama Alternative
ALTR001	ROSITA
ALTR002	RIANI
ALTR003	LATIFAH
ALTR004	WULANDARI
ALTR005	ANI
ALTR006	YULI

Gambar 3. Form Alternative

Source Code Form Alternative

```
package spk_new;  
import java.awt.event.KeyEvent;  
import java.sql.*;  
import java.util.logging.Level;  
import java.util.logging.Logger;  
import javax.swing.*;  
import javax.swing.table.DefaultTableModel;  
import sun.misc.Cleaner;
```

```

public class alternative extends javax.swing.JFrame {
    koneksi kon=new koneksi();
    private Object [][] dataalternative=null;
    private String[] label={"Kode Alternative","Nama Alternative"};
    public alternative() {
        initComponents();
        kon.setKoneksi();
        nonaktif();
        BacaTabelalternative(); }
    private void BacaTabelalternative(){
        try{
            String sql="Select *From alternative order by kdalternative";
            kon.rs=kon.st.executeQuery(sql);
            ResultSetMetaData m=kon.rs.getMetaData();
            int kolom=m.getColumnCount();
            int baris=0;
            while(kon.rs.next()){
                baris=kon.rs.getRow();
            }
            dataalternative=new Object[baris][kolom];
            int x=0;
            kon.rs.beforeFirst();
            while(kon.rs.next()){
                dataalternative[x][0]=kon.rs.getString("kdalternative");
                dataalternative[x][1]=kon.rs.getString("nmalternative");
                x++;
            }
            tblalternative.setModel(new DefaultTableModel(dataalternative,label));
        } catch(SQLException e){
            JOptionPane.showMessageDialog(null, e); } }
    private void BacaTabelalternative2(){
        try{

```

```

        String sql="select *from alternative where nmalternative like '%"
+carialternative.getText()+"%' ";
        kon.rs=kon.st.executeQuery(sql);
        ResultSetMetaData m=kon.rs.getMetaData();
        int kolom=m.getColumnCount();
        int baris=0;
        while(kon.rs.next()){
            baris=kon.rs.getRow();
        }
        dataalternative=new Object[baris][kolom];
        int x=0;
        kon.rs.beforeFirst();
        while(kon.rs.next()){
            dataalternative[x][0]=kon.rs.getString("kdalternative");
            dataalternative[x][1]=kon.rs.getString("nmalternative");
            x++;
        }
        tblalternative.setModel(new DefaultTableModel(dataalternative,label));
    } catch(SQLException e){
        JOptionPane.showMessageDialog(null, e);    }    }

private void setTable(){
    int row=tblalternative.getSelectedRow();
    kodealternative.setText((String)tblalternative.getValueAt(row,0));
    nmalternative.setText((String)tblalternative.getValueAt(row,1));    }

private void BersihField(){
    kodealternative.setText("");
    nmalternative.setText("");
    carialternative.setText("");    }

private void aktif(){
    kodealternative.setEnabled(true);
    nmalternative.setEnabled(true);    }

private void nonaktif(){
    kodealternative.setEnabled(false);

```

```

nmalternative.setEnabled(false);
edit.setEnabled(false);
update.setEnabled(false);
hapus.setEnabled(false);
simpan.setEnabled(false); }
private void SimpanData(){
    try{
        String sql="insert into alternative
values('"+kodealternative.getText()+"','"+nmalternative.getText()+"')";
        kon.st.executeUpdate(sql);
        JOptionPane.showMessageDialog(null,"Data berhasil disimpan");
        BersihField();
        BacaTabelalternative();
    } catch(SQLException e){
        JOptionPane.showMessageDialog(null,e); } }
private void EditData(){
    try{
        String sql="Update alternative set
kdalternative='"+kodealternative.getText()+"',nmalternative='"+nmalternative.getText(
)+"' where kdalternative='"+kodealternative.getText()+"'";
        kon.st.executeUpdate(sql);
        JOptionPane.showMessageDialog(null,"Data berhasil diupdate");
        BersihField();
        BacaTabelalternative();
        kon.st.close();
    } catch(SQLException e){
        JOptionPane.showMessageDialog(null,e); } }
private void HapusData(){
    try{
        String sql="Delete from alternative where
kdkriteria='"+kodealternative.getText()+"'";
        kon.st.executeUpdate(sql);
        JOptionPane.showMessageDialog(null,"Data berhasil dihapus");
    }
}

```

```

        BersihField();
        BacaTabelalternative();
        kon.st.close();
    } catch(SQLException e){
        JOptionPane.showMessageDialog(null, e);    }    }

private void keluarActionPerformed(java.awt.event.ActionEvent evt) {
    dispose();    }

private void tambahActionPerformed(java.awt.event.ActionEvent evt) {
    kon.setKoneksi();
    aktif();
    batal.setEnabled(true);
    tambah.setEnabled(false);
    simpan.setEnabled(true);
    kodealternative.requestFocus();    }

private void simpanActionPerformed(java.awt.event.ActionEvent evt) {
    kon.setKoneksi();
    if (kodealternative.getText().isEmpty() || nmalternative.getText().isEmpty()) {
        JOptionPane.showMessageDialog(this, "Lengkapi Data", "Konfirmasi",
JOptionPane.INFORMATION_MESSAGE);
        tambah.setEnabled(true);
    } else {
        tambah.setEnabled(true);
        keluar.setEnabled(true);
        SimpanData();
        try {
            kon.st.close();
        } catch (SQLException ex) {
            Logger.getLogger(Kriteria.class.getName()).log(Level.SEVERE, null, ex);    }    }    }

private void carialternativeKeyTyped(java.awt.event.KeyEvent evt) {
    kon.setKoneksi();
    BacaTabelalternative2();    }

private void tblalternativeMouseClicked(java.awt.event.MouseEvent evt) {
    setTable();

```

```

hapus.setEnabled(true);
edit.setEnabled(true);
tambah.setEnabled(false); }

private void hapusActionPerformed(java.awt.event.ActionEvent evt) {
    kon.setKoneksi();
    if (JOptionPane.showConfirmDialog(this, "yakin mau dihapus?", "konfirmasi",
JOptionPane.YES_NO_OPTION) == JOptionPane.YES_OPTION) {
        HapusData();
        tambah.setEnabled(true);
        nonaktif();
        BersihField();
    } else {
        JOptionPane.showMessageDialog(this, "Data Batal Dihapus", "Konfirmasi",
JOptionPane.INFORMATION_MESSAGE);
        tambah.setEnabled(true);
        nonaktif();
        BersihField();
        return;
    } formWindowActivated(null); }

private void editActionPerformed(java.awt.event.ActionEvent evt) {
    aktif();
    kodealternative.setEnabled(false);
    edit.setEnabled(false);
    update.setEnabled(true);
    batal.setEnabled(true);
    hapus.setEnabled(false);
    tambah.setEnabled(false); }

private void updateActionPerformed(java.awt.event.ActionEvent evt) {
    kon.setKoneksi();
    update.setEnabled(false);
    tambah.setEnabled(true);
    EditData(); }

private void formWindowActivated(java.awt.event.WindowEvent evt) {

```

```
Bersihkan();  
nonaktif(); }
```

B. Form Kriteria

Form kriteria ini berisikan kode kriteria, nama kriteria, jenis kriteria dan bobot yang digunakan sebagai kriteria perhitungan dalam menilai peringkat kinerjanya. Pada saat form ini di running textbox kode kriteria, nama kriteria, jenis kriteria dan bobot dalam keadaan nonaktif dan yang aktif button tambah, batal, cari dan keluar.

Textbox cari digunakan untuk mencari kode kriteria, nama kriteria, jenis kriteria dan bobot yang ada dengan cara mengisikan nama kriteria di textbox maka tabel akan tampil nama kriteria yang dicari. Kemudian double klik nama kriteria maka akan tampil datanya di dalam form kriteria. Button edit digunakan untuk mengedit data yang tampil di form kriteria dan dapat di update datanya dengan button update, selain itu datanya juga dapat di hapus dengan button hapus.

Button tambah digunakan untuk menambahkan data kriteria dengan mengisi kode kriteria, nama kriteria, jenis kriteria dan bobot untuk menyimpannya dengan button simpan.

Button keluar digunakan untuk menutup form kriteria dan kembali ke menu utama. Berikut adalah tampilan dari form kriteria.

Kode Kriteria	Nama Kriteria	Jenis Kriteria	Bobot
KR001	Kehadiran	Benefit	3
KR002	Hasil Penjualan	Cost	4
KR003	Attitude	Benefit	4
KR004	Lama Bekerja	Benefit	4
KR005	Latar Belakang...	Benefit	5

Gambar 4. Form Kriteria

Source Code Form Kriteria

```
package spk_new;  
  
import java.awt.event.KeyEvent;  
  
import java.sql.*;  
  
import java.util.logging.Level;  
import java.util.logging.Logger;  
  
import javax.swing.*;  
  
import javax.swing.table.DefaultTableModel;
```

```

import sun.misc.Cleaner;

public class Kriteria extends javax.swing.JFrame {
    koneksi kon=new koneksi();
    private Object [][] datakriteria=null;
    private String[] label={"Kode Kriteria","Nama Kriteria","Jenis Kriteria","Bobot"};
    public Kriteria() {
        initComponents();
        kon.setKoneksi();
        nonaktif();
        BacaTabelkriteria(); }
    private void BacaTabelkriteria(){
        try{
            String sql="Select *From kriteria order by kdkriteria";
            kon.rs=kon.st.executeQuery(sql);
            ResultSetMetaData m=kon.rs.getMetaData();
            int kolom=m.getColumnCount();
            int baris=0;
            while(kon.rs.next()){
                baris=kon.rs.getRow();
            }
            datakriteria=new Object[baris][kolom];
            int x=0;
            kon.rs.beforeFirst();
            while(kon.rs.next()){
                datakriteria[x][0]=kon.rs.getString("kdkriteria");
                datakriteria[x][1]=kon.rs.getString("nmkriteria");
                datakriteria[x][2]=kon.rs.getString("jns");
                datakriteria[x][3]=kon.rs.getString("bobot");
                x++; }
            tblkriteria.setModel(new DefaultTableModel(datakriteria,label));
        } catch(SQLException e){
            JOptionPane.showMessageDialog(null, e); } }
    private void BacaTabelkriteria2(){

```

```

try{
    String sql="select *from kriteria where nmkriteria like '%" +cari.getText()+ "%' ";
    kon.rs=kon.st.executeQuery(sql);
    ResultSetMetaData m=kon.rs.getMetaData();
    int kolom=m.getColumnCount();
    int baris=0;
    while(kon.rs.next()){
        baris=kon.rs.getRow();
    }
    datakriteria=new Object[baris][kolom];
    int x=0;
    kon.rs.beforeFirst();
    while(kon.rs.next()){
        datakriteria[x][0]=kon.rs.getString("kdkriteria");
        datakriteria[x][1]=kon.rs.getString("nmkriteria");
        datakriteria[x][2]=kon.rs.getString("jns");
        datakriteria[x][3]=kon.rs.getString("bobot");
        x++; }
    tblkriteria.setModel(new DefaultTableModel(datakriteria,label));
} catch(SQLException e){
    JOptionPane.showMessageDialog(null, e); } }

private void setTable(){
    int row=tblkriteria.getSelectedRow();
    kdkriteria.setText((String)tblkriteria.getValueAt(row,0));
    nmkriteria.setText((String)tblkriteria.getValueAt(row,1));
    jnskriteria.setText((String)tblkriteria.getValueAt(row,2));
    bobot.setSelectedItem((String) tblkriteria.getValueAt(row, 3)); }

private void BersihField(){
    kdkriteria.setText("");
    nmkriteria.setText("");
    jnskriteria.setText(""); }

private void aktif(){
    kdkriteria.setEnabled(true);

```

```

        nmkriteria.setEnabled(true);
        jnsriteria.setEnabled(true);
        bobot.setEnabled(true); }

private void nonaktif(){
    kdkriteria.setEnabled(false);
    nmkriteria.setEnabled(false);
    jnsriteria.setEnabled(false);
    bobot.setEnabled(false);
    bedit.setEnabled(false);
    bupdate.setEnabled(false);
    bhapus.setEnabled(false);
    bsimpan.setEnabled(false); }

private void SimpanData(){
    try{
        String sql="insert into kriteria
values('"+kdkriteria.getText()+"','"+nmkriteria.getText()+"','"+jnsriteria.getText()+"','"+
+bobot.getSelectedItemAt()+"')";
        kon.st.executeUpdate(sql);
        JOptionPane.showMessageDialog(null,"Data berhasil disimpan");
        BersihField();
        BacaTabelkriteria();
    } catch(SQLException e){
        JOptionPane.showMessageDialog(null,e); } }

private void EditData(){
    try{
        String sql="Update kriteria set
kdkriteria='"+kdkriteria.getText()+"',nmkriteria='"+nmkriteria.getText()+"',jns='"+jnsk
riteria.getText()+"',bobot='"+bobot.getSelectedItemAt()+"' where
kdkriteria='"+kdkriteria.getText()+"'";
        kon.st.executeUpdate(sql);
        JOptionPane.showMessageDialog(null,"Data berhasil diupdate");
        BersihField();
        BacaTabelkriteria();
    }

```

```

        kon.st.close();
    } catch(SQLException e){
        JOptionPane.showMessageDialog(null,e); } }
private void HapusData(){
    try{
        String sql="Delete from kriteria where kdkriteria='"+kdkriteria.getText()+"'";
        kon.st.executeUpdate(sql);
        JOptionPane.showMessageDialog(null,"Data berhasil dihapus");
        BersihField();
        BacaTabelkriteria();
        kon.st.close();
    } catch(SQLException e){
        JOptionPane.showMessageDialog(null, e); } }
private void formWindowActivated(java.awt.event.WindowEvent evt) {
    BersihField();
    nonaktif(); }
private void bkeluarActionPerformed(java.awt.event.ActionEvent evt) {
    dispose(); }
private void btambahActionPerformed(java.awt.event.ActionEvent evt) {
    kon.setKoneksi();
    aktif();
    bbatal.setEnabled(true);
    btambah.setEnabled(false);
    bsimpan.setEnabled(true);
    kdkriteria.requestFocus(); }
private void bbatalActionPerformed(java.awt.event.ActionEvent evt) {
    nonaktif();
    BersihField();
    btambah.setEnabled(true);
    try {
        kon.st.close();
    } catch (SQLException ex) {
        Logger.getLogger(Kriteria.class.getName()).log(Level.SEVERE, null, ex); } }

```

```

private void bsimpanActionPerformed(java.awt.event.ActionEvent evt) {
    kon.setKoneksi();
    if (kdkriteria.getText().isEmpty() || nmkriteria.getText().isEmpty() ||
jnskriteria.getText().isEmpty()) {
        JOptionPane.showMessageDialog(this, "Lengkapi Data", "Konfirmasi",
JOptionPane.INFORMATION_MESSAGE);
        btambah.setEnabled(true);
    } else {
        btambah.setEnabled(true);
        bkeluar.setEnabled(true);
        SimpanData();
        try {
            kon.st.close();
        } catch (SQLException ex) {
            Logger.getLogger(Kriteria.class.getName()).log(Level.SEVERE, null, ex); } } }
private void cariKeyTyped(java.awt.event.KeyEvent evt) {
    kon.setKoneksi();
    BacaTabelkriteria2(); }
private void tblkriteriaMouseClicked(java.awt.event.MouseEvent evt) {
    setTable();
    bhapus.setEnabled(true);
    bedit.setEnabled(true);
    btambah.setEnabled(false); }
private void bhapusActionPerformed(java.awt.event.ActionEvent evt) {
    kon.setKoneksi();
    if (JOptionPane.showConfirmDialog(this, "yakin mau dihapus?", "konfirmasi",
JOptionPane.YES_NO_OPTION) == JOptionPane.YES_OPTION) {
        HapusData();
        btambah.setEnabled(true);
        nonaktif();
        BersihField();
    } else {

```

```

        JOptionPane.showMessageDialog(this, "Data Batal Dihapus", "Konfirmasi",
JOptionPane.INFORMATION_MESSAGE);

        btambah.setEnabled(true);
        nonaktif();
        BersihField();
        return;
    } formWindowActivated(null); }

private void beditActionPerformed(java.awt.event.ActionEvent evt) {
    aktif();
    kdkriteria.setEnabled(false);
    bedit.setEnabled(false);
    bupdate.setEnabled(true);
    bbatal.setEnabled(true);
    bhapus.setEnabled(false);
    btambah.setEnabled(false); }

private void bupdateActionPerformed(java.awt.event.ActionEvent evt) {
    bupdate.setEnabled(false);
    btambah.setEnabled(true);
    EditData(); }

```

C. Form Range Kriteria

Form range kriteria ini berisikan nama kriteria, kode range, keterangan dan bobot range yang digunakan untuk memberikan nilai range disetiap kriteria yang nantinya akan digunakan sebagai perhitungan peringkat kinerjanya. Pada saat form ini di running textbox nama kriteria, kode range, keterangan dan bobot range dalam keadaan nonaktif dan yang aktif button tambah, batal, filter dan keluar.

Combo Filter digunakan untuk mencari nama kriteria sesuai yang dipilih maka data akan terfilter sesuai yang dipilih dan tampil datanya di dalam tabel. Kemudian double klik nama kriteria maka akan tampil datanya di dalam form range kriteria. Button edit digunakan untuk mengedit data yang tampil di form range kriteria dan dapat di update datanya dengan button update, selain itu datanya juga dapat di hapus dengan button hapus.

Button tambah digunakan untuk menambahkan data range kriteria dengan mengisi nama kriteria, kode range, keterangan dan bobot range untuk menyimpannya dengan button simpan.

Button keluar digunakan untuk menutup form range kriteria dan kembali ke menu utama. Berikut adalah tampilan dari form range kriteria.

Kode Range	Nama Kriteria	Bobot Range	Keterangan
KRH1	Kehadiran	Sangat Rendah	1
KRH2	Kehadiran	Rendah	2
KRH3	Kehadiran	Cukup	3
KRH4	Kehadiran	Tinggi	4
KRH5	Kehadiran	Tinggi	5

Gambar 5. Form Range Kriteria

Source Code Form Range Kriteria

```
package spk_new;

import java.awt.event.KeyEvent;
import java.sql.*;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.swing.*;
import javax.swing.table.DefaultTableModel;
import sun.misc.Cleaner;

public class range_kriteria extends javax.swing.JFrame {
    koneksi kon=new koneksi();
    private Object [][] datarange=null;
    private String[] label={"Kode Range","Nama Kriteria","Bobot Range","Keterangan"};
    public range_kriteria() {
        initComponents();
        kon.setKoneksi();
        nonaktif();
        BacaTabelrange();
        tampil_combo();
        tampil_combo2(); }
}
```

```

private void BacaTabelrange(){
    try{
        String sql="Select *From rangekr order by kdrange";
        kon.rs=kon.st.executeQuery(sql);
        ResultSetMetaData m=kon.rs.getMetaData();
        int kolom=m.getColumnCount();
        int baris=0;
        while(kon.rs.next()){
            baris=kon.rs.getRow();
        }
        datarange=new Object[baris][kolom];
        int x=0;
        kon.rs.beforeFirst();
        while(kon.rs.next()){
            datarange[x][0]=kon.rs.getString("kdrange");
            datarange[x][1]=kon.rs.getString("nmkriteriakr");
            datarange[x][2]=kon.rs.getString("bobotrange");
            datarange[x][3]=kon.rs.getString("ket");
            x++; }
        tblrange.setModel(new DefaultTableModel(datarange,label));
    } catch(SQLException e){
        JOptionPane.showMessageDialog(null, e); } }

private void setTable(){
    int row=tblrange.getSelectedRow();
    koderange.setText((String)tblrange.getValueAt(row,0));
    bkode.setSelectedItem((String)tblrange.getValueAt(row,1));
    bbobot.setSelectedItem((String)tblrange.getValueAt(row, 2));
    ket.setText((String)tblrange.getValueAt(row,3)); }

private void tampil_combo(){
    try{
        String sql="Select *From kriteria ";
        kon.rs=kon.st.executeQuery(sql);
        while (kon.rs.next()){

```

```

        bkode.addItem(kon.rs.getString("nmkriteria")); }
    }catch(SQLException e){
        System.out.println("Koneksi Gagal"+ e.toString()); } }
private void tampil_combo2(){
    try{
        String sql="Select *From kriteria ";
        kon.rs=kon.st.executeQuery(sql);
        while (kon.rs.next()){
            bnmkr.addItem(kon.rs.getString("nmkriteria")); }
        }catch(SQLException e){
            System.out.println("Koneksi Gagal"+ e.toString()); } }
private void filterkriteria(){
    try{
        String sql="select *from rangekr where nmkriteriakr like '%"
+bnmkr.getSelectedItemAt()+ "%' ";
        kon.rs=kon.st.executeQuery(sql);
        ResultSetMetaData m=kon.rs.getMetaData();
        int kolom=m.getColumnCount();
        int baris=0;
        while(kon.rs.next()){
            baris=kon.rs.getRow();
        }
        datarange=new Object[baris][kolom];
        int x=0;
        kon.rs.beforeFirst();
        while(kon.rs.next()){
            datarange[x][0]=kon.rs.getString("kdrange");
            datarange[x][1]=kon.rs.getString("nmkriteriakr");
            datarange[x][2]=kon.rs.getString("ket");
            datarange[x][3]=kon.rs.getString("bobotrange");
            x++; }
        tblrange.setModel(new DefaultTableModel(datarange,label));
    } catch(SQLException e){

```

```

        JOptionPane.showMessageDialog(null, e); } }
private void aktif(){
    koderange.setEnabled(true);
    ket.setEnabled(true);
    bbobot.setEnabled(true);
    bkode.setEnabled(true); }
private void nonaktif(){
    koderange.setEnabled(false);
    ket.setEnabled(false);
    bbobot.setEnabled(false);
    bkode.setEnabled(false);
    bedit.setEnabled(false);
    bupdate.setEnabled(false);
    bhapus.setEnabled(false);
    bsimpan.setEnabled(false); }
private void BersihField(){
    koderange.setText("");
    ket.setText("");
    bbobot.addItem("Pilih"); }
private void SimpanData(){
    try{
        String sql="insert into rangekr
values('"+koderange.getText()+"','"+bkode.getSelectedItem()+"','"+bbobot.getSelectedIt
em()+"','"+ket.getText()+"')";
        kon.st.executeUpdate(sql);
        JOptionPane.showMessageDialog(null,"Data berhasil disimpan");
        BersihField();
        BacaTabelrange();
    } catch(SQLException e){
        JOptionPane.showMessageDialog(null,e); } }
private void EditData(){
    try{

```

```

        String sql="Update rangekr set
kdrange='"+koderange.getText()+"",nmkriteriakr='"+bkode.getSelectedItemAt()+"',ket="
+ket.getText()+"',bobotrange='"+bbobot.getSelectedItemAt()+"' where
kdrange='"+koderange.getText()+"";
        kon.st.executeUpdate(sql);
        JOptionPane.showMessageDialog(null,"Data berhasil diupdate");
        BersihField();
        BacaTabelrange();
        kon.st.close();
    } catch(SQLException e){
        JOptionPane.showMessageDialog(null,e); } }

private void HapusData(){
    try{
        String sql="Delete from rangekr where kdrange='"+koderange.getText()+"";
        kon.st.executeUpdate(sql);
        JOptionPane.showMessageDialog(null,"Data berhasil dihapus");
        BersihField();
        BacaTabelrange();
        kon.st.close();
    } catch(SQLException e){
        JOptionPane.showMessageDialog(null, e); } }

private void bkeluarActionPerformed(java.awt.event.ActionEvent evt) {
    dispose(); }

private void btambahActionPerformed(java.awt.event.ActionEvent evt) {
    kon.setKoneksi();
    aktif();
    bbatal.setEnabled(true);
    btambah.setEnabled(false);
    bsimpan.setEnabled(true);
    bkode.requestFocus(); }

private void bupdateActionPerformed(java.awt.event.ActionEvent evt) {
    kon.setKoneksi();
    bupdate.setEnabled(false);

```

```

    btambah.setEnabled(true);
    EditData(); }
private void beditActionPerformed(java.awt.event.ActionEvent evt) {
    aktif();
    koderange.setEnabled(false);
    bedit.setEnabled(false);
    bupdate.setEnabled(true);
    bbatal.setEnabled(true);
    bhapus.setEnabled(false);
    btambah.setEnabled(false); }
private void bhapusActionPerformed(java.awt.event.ActionEvent evt) {
    kon.setKoneksi();
    if (JOptionPane.showConfirmDialog(this, "yakin mau dihapus?", "konfirmasi",
JOptionPane.YES_NO_OPTION) == JOptionPane.YES_OPTION) {
        HapusData();
        btambah.setEnabled(true);
        nonaktif();
        BersihField();
    } else {
        JOptionPane.showMessageDialog(this, "Data Batal Dihapus", "Konfirmasi",
JOptionPane.INFORMATION_MESSAGE);
        btambah.setEnabled(true);
        nonaktif();
        BersihField();
        return;
    } formWindowActivated(null); }
private void formWindowActivated(java.awt.event.WindowEvent evt) {
    BersihField();
    nonaktif(); }
private void bbatalActionPerformed(java.awt.event.ActionEvent evt) {
    nonaktif();
    BersihField();
    btambah.setEnabled(true);

```

```

try {
    kon.st.close();
} catch (SQLException ex) {
    Logger.getLogger(Kriteria.class.getName()).log(Level.SEVERE, null, ex); } }
private void bsimpanActionPerformed(java.awt.event.ActionEvent evt) {
    kon.setKoneksi();
    if (koderange.getText().isEmpty() || ket.getText().isEmpty()) {
        JOptionPane.showMessageDialog(this, "Lengkapi Data", "Konfirmasi",
JOptionPane.INFORMATION_MESSAGE);
        btambah.setEnabled(true);
    } else {
        btambah.setEnabled(true);
        bkeluar.setEnabled(true);
        SimpanData();
        try {
            kon.st.close();
        } catch (SQLException ex) {
            Logger.getLogger(Kriteria.class.getName()).log(Level.SEVERE, null, ex); } } }
private void tblrangeMouseClicked(java.awt.event.MouseEvent evt) {
    setTable();
    bhapus.setEnabled(true);
    bedit.setEnabled(true);
    btambah.setEnabled(false); }
private void bnmkrMouseClicked(java.awt.event.MouseEvent evt) {
    filterkriteria(); }

```

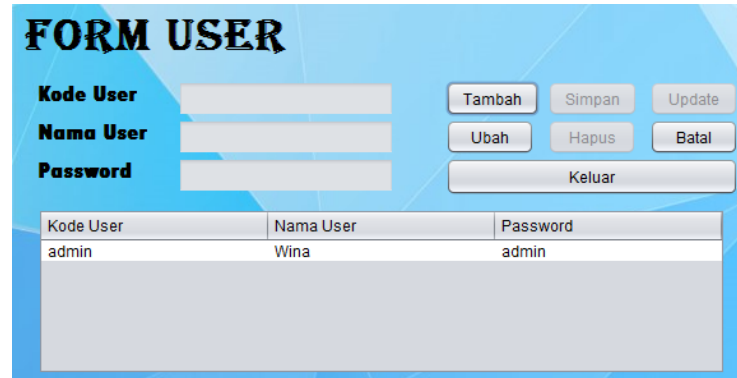
D. Form User

Form user ini berisikan kode user, nama user dan password yang digunakan sebagai hak akses login. Pada saat form ini di running textbox kode user, nama user dan password dalam keadaan nonaktif dan yang aktif button tambah, batal dan keluar.

Button ubah digunakan untuk mengubah data dengan cara mengklik data yang ada di tabel maka akan tampil data user. Data user yang diubah dapat diupdate datanya dengan button update, selain itu datanya juga dapat di hapus dengan button hapus.

Button tambah digunakan untuk menambahkan data user dengan mengisi kode user, nama user dan password untuk menyimpannya dengan button simpan.

Button keluar digunakan untuk menutup form range kriteria dan kembali ke menu utama. Berikut adalah tampilan dari form user.



Kode User	Nama User	Password
admin	Wina	admin

Gambar 6. Form User

Source Code Form User

```
package spk_new;
import java.awt.event.KeyEvent;
import java.sql.*;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.swing.*;
import javax.swing.table.DefaultTableModel;
import sun.misc.Cleaner;

public class user extends javax.swing.JFrame {
    koneksi kon=new koneksi();
    private Object [][] datauser=null;
    private String[] label={"Kode User","Nama User","Password"};
    public user() {
        initComponents();
        kon.setKoneksi();
        nonaktif();
        BacaTabeluser(); }
    private void BacaTabeluser(){
        try{
            String sql="Select *From user order by kduser";
```

```

kon.rs=kon.st.executeQuery(sql);
ResultSetMetaData m=kon.rs.getMetaData();
int kolom=m.getColumnCount();
int baris=0;
while(kon.rs.next()){
    baris=kon.rs.getRow();
}
datauser=new Object[baris][kolom];
int x=0;
kon.rs.beforeFirst();
while(kon.rs.next()){
    datauser[x][0]=kon.rs.getString("kduser");
    datauser[x][1]=kon.rs.getString("nmuser");
    datauser[x][2]=kon.rs.getString("password");
    x++; }
tabeluser.setModel(new DefaultTableModel(datauser,label));
} catch(SQLException e){
    JOptionPane.showMessageDialog(null, e); } }

private void setTable(){
    int row=tabeluser.getSelectedRow();
    tkduser.setText((String)tabeluser.getValueAt(row,0));
    tnmuser.setText((String)tabeluser.getValueAt(row,1));
    tpassword.setText((String)tabeluser.getValueAt(row,2)); }

private void BersihField(){
    tkduser.setText("");
    tnmuser.setText("");
    tpassword.setText(""); }

private void aktif(){
    tkduser.setEnabled(true);
    tnmuser.setEnabled(true);
    tpassword.setEnabled(true); }

private void nonaktif(){
    tkduser.setEnabled(false);

```

```

tnmuser.setEnabled(false);
tpassword.setEnabled(false);
bupdate.setEnabled(false);
bhapus.setEnabled(false);
bsimpan.setEnabled(false); }
private void SimpanData(){
    try{
        String sql="insert into user
values('"+tkduser.getText()+"','"+tnmuser.getText()+"','"+tpassword.getText()+"')";
        kon.st.executeUpdate(sql);
        JOptionPane.showMessageDialog(null,"Data berhasil disimpan");
        BersihField();
        BacaTabeluser();
    } catch(SQLException e){
        JOptionPane.showMessageDialog(null,e);    } }
private void EditData(){
    try{
        String sql="Update user set
kduser='"+tkduser.getText()+"',nmuser='"+tnmuser.getText()+"',password='"+tpasswo
rd.getText()+"' where kduser='"+tkduser.getText()+"'";
        kon.st.executeUpdate(sql);
        JOptionPane.showMessageDialog(null,"Data berhasil diupdate");
        BersihField();
        BacaTabeluser();
        kon.st.close();
    } catch(SQLException e){
        JOptionPane.showMessageDialog(null,e);    } }
private void HapusData(){
    try{
        String sql="Delete from user where kduser='"+tkduser.getText()+"'";
        kon.st.executeUpdate(sql);
        JOptionPane.showMessageDialog(null,"Data berhasil dihapus");
        BersihField();
    }

```

```

        BacaTabeluser();
        kon.st.close();
    } catch(SQLException e){
        JOptionPane.showMessageDialog(null, e);    } }

private void btambahActionPerformed(java.awt.event.ActionEvent evt) {
    kon.setKoneksi();
    aktif();
    bbatal.setEnabled(true);
    btambah.setEnabled(false);
    bsimpan.setEnabled(true);
    tkduser.requestFocus(); }

private void bsimpanActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    kon.setKoneksi();
    if (tkduser.getText().isEmpty() || tnmluser.getText().isEmpty() ||
    tpassword.getText().isEmpty()) {
        JOptionPane.showMessageDialog(this, "Lengkapi Data", "Konfirmasi",
        JOptionPane.INFORMATION_MESSAGE);
        btambah.setEnabled(true);
    } else {
        btambah.setEnabled(true);
        bkeluar.setEnabled(true);
        SimpanData();
        try {
            kon.st.close();
        } catch (SQLException ex) {
            Logger.getLogger(Kriteria.class.getName()).log(Level.SEVERE, null, ex);    }    } }

private void buupdateActionPerformed(java.awt.event.ActionEvent evt) {
    buupdate.setEnabled(false);
    btambah.setEnabled(true);
    EditData(); }

private void bubahActionPerformed(java.awt.event.ActionEvent evt) {
    aktif();

```

```

tkduser.setEnabled(false);
bubah.setEnabled(false);
bupdate.setEnabled(true);
bbatal.setEnabled(true);
bhapus.setEnabled(false);
btambah.setEnabled(false); }

private void bhapusActionPerformed(java.awt.event.ActionEvent evt) {
    kon.setKoneksi();
    if (JOptionPane.showConfirmDialog(this, "yakin mau dihapus?", "konfirmasi",
JOptionPane.YES_NO_OPTION) == JOptionPane.YES_OPTION) {
        HapusData();
        btambah.setEnabled(true);
        nonaktif();
        BersihField();
    } else {
        JOptionPane.showMessageDialog(this, "Data Batal Dihapus", "Konfirmasi",
JOptionPane.INFORMATION_MESSAGE);
        btambah.setEnabled(true);
        nonaktif();
        BersihField();
        return;
    } formWindowActivated(null); }

private void bbatalActionPerformed(java.awt.event.ActionEvent evt) {
    BersihField();
    btambah.setEnabled(true);
    try {
        kon.st.close();
    } catch (SQLException ex) {
        Logger.getLogger(Kriteria.class.getName()).log(Level.SEVERE, null, ex); } }

private void formWindowActivated(java.awt.event.WindowEvent evt) {
    BersihField();
    nonaktif(); }

private void bkeluarActionPerformed(java.awt.event.ActionEvent evt) {

```

```
        dispose(); }  
private void tabeluserMouseClicked(java.awt.event.MouseEvent evt) {  
    setTable();  
    bhapus.setEnabled(true);  
    bubah.setEnabled(true);  
    btambah.setEnabled(false); }  

```

BAB III

MENU OLAH DATA

A. Form Penilaian SPK

Form penilaian SPK ini berisikan kode nilai, kode alternative dan bobot nilai kriteria yang digunakan sebagai penilaian sistem penunjang keputusan. Pada saat form ini di running textbox kode nilai, kode alternative dan bobot nilai kriteria dalam keadaan nonaktif dan yang aktif button tambah, hapus dan keluar.

Button hapus digunakan untuk menghaus data yang ada dengan cara memilih data yang akan dihapus didalam tabel penilaian alternative kemudian pilih button hapus.

Button tambah digunakan untuk menambahkan data penilaian SPK, pada saat di pilih button tambah maka kode nilai akan tampil otomatis kemudian pilih kode alternatif, bobot nilai serta latar belakang pendidikan dan untuk menyimpannya dengan button simpan.

Button keluar digunakan untuk menutup form penilaian SPK dan kembali ke menu utama. Berikut adalah tampilan dari form penilaian SPK.

Kode Nil...	Kode Alt...	K1	K2	K3	K4	K5
NA-001	ALTR001	4	5	5	4	4
NA-002	ALTR002	4	4	4	4	5
NA-003	ALTR003	5	4	4	4	4
NA-004	ALTR004	5	4	4	4	4
NA-005	ALTR005	4	2	4	4	4

Gambar 7. Form Penilaian SPK

Source Code Form Penilaian SPK

```
package spk_new;  
  
import java.awt.event.KeyEvent;  
  
import java.sql.*;  
  
import java.util.logging.Level;  
import java.util.logging.Logger;  
  
import javax.swing.*;
```

```

import javax.swing.table.DefaultTableModel;
import sun.misc.Cleaner;
public class penilaianspk extends javax.swing.JFrame {
    koneksi kon=new koneksi();
    private Object [][] datanilai=null;
    private String[] label={"Kode Nilai","Kode Alternatif","K1","K2","K3","K4","K5"};
    public penilaianspk() {
        initComponents();
        kon.setKoneksi();
        BacaTabelnilai();
        tampil_comboal(); }
    private void tampil_comboal(){
        try{
            String sql="Select *From alternative ";
            kon.rs=kon.st.executeQuery(sql);
            while (kon.rs.next()){
                cal.addItem(kon.rs.getString("kdalternatif")); }
        } catch(SQLException e) {
            System.out.println("Koneksi Gagal"+ e.toString()); } }
    public String getno()
    {
        String urutan=null;
        try{
            kon.rs=kon.st.executeQuery("select right(kdnilai,3)+1 from nilai_alternatif as Nomor
            order by kdnilai desc");
            if(kon.rs.next())
            {
                urutan=kon.rs.getString(1);
                while(urutan.length()<3)
                    urutan="0"+urutan;
                urutan="NA-"+urutan;
            }else
            { urutan="NA-"+ "001"; }
        }
    }

```

```

    }catch(SQLException e){
        JOptionPane.showMessageDialog(null,"No NA Error : "+e);
    } return urutan; }

private void BacaTabelnilai(){
    try{
        String sql="Select *From nilai_alternatif order by kdnilai";
        kon.rs=kon.st.executeQuery(sql);
        ResultSetMetaData m=kon.rs.getMetaData();
        int kolom=m.getColumnCount();
        int baris=0;
        while(kon.rs.next()){
            baris=kon.rs.getRow();
        }
        datanilai=new Object[baris][kolom];
        int x=0;
        kon.rs.beforeFirst();
        while(kon.rs.next()){
            datanilai[x][0]=kon.rs.getString("kdnilai");
            datanilai[x][1]=kon.rs.getString("kdalternative");
            datanilai[x][2]=kon.rs.getString("K1");
            datanilai[x][3]=kon.rs.getString("K2");
            datanilai[x][4]=kon.rs.getString("K3");
            datanilai[x][5]=kon.rs.getString("K4");
            datanilai[x][6]=kon.rs.getString("K5");
            x++; }
        tblnilai.setModel(new DefaultTableModel(datanilai,label));
    } catch(SQLException e){
        JOptionPane.showMessageDialog(null, e);    } }

private void SimpanData(){
    try{
        String sql="insert into nilai_alternatif
values('"+nilai.getText()+"','"+cal.getSelectedItemAt()+"','"+ck1.getSelectedItemAt()+"',"

```

```

        +
        ""+ck2.getSelectedItemAt()+""+ck3.getSelectedItemAt()+""+ck4.getSelectedItemAt()+""+
        + ""+ck5.getSelectedItemAt()+""");
        kon.st.executeUpdate(sql);
        JOptionPane.showMessageDialog(null,"Data berhasil disimpan");
        BacaTabelnilai();
    } catch(SQLException e){
        JOptionPane.showMessageDialog(null,e);    } }

private void delete(){
    try{
        String sql="Delete from nilai_alternatif where kdnilai='"+nilai.getText()+"'";
        kon.st.executeUpdate(sql);
        JOptionPane.showMessageDialog(null,"Data berhasil dihapus");
        BacaTabelnilai();
        kon.st.close();
    } catch(SQLException e){
        JOptionPane.showMessageDialog(null, e);    } }

private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
    cal.setSelectedItemAt("pilih");
    SimpanData(); }

private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
    dispose(); }

private void jButton3ActionPerformed(java.awt.event.ActionEvent evt) {
    delete(); }

private void jButton4ActionPerformed(java.awt.event.ActionEvent evt) {
    nilai.setText(getno());
    cal.setSelectedItemAt("pilih");
    ck1.setSelectedItemAt("pilih");
    ck2.setSelectedItemAt("pilih");
    ck3.setSelectedItemAt("pilih");
    ck4.setSelectedItemAt("pilih");
    ck5.setSelectedItemAt("pilih"); }

private void formWindowActivated(java.awt.event.WindowEvent evt) {

```

```
nilai.setEditable(false); }
```

B. Form Hasil

Form hasil ini berisikan data dari penilaian SPK, bobot Kriteria yang digunakan untuk menghitung nilai Si dan Vi yang nantinya akan menghasilkan suatu nilai yang dijadikan peringkat kinerja karyawan. Pada saat form ini di running button tampil data, button ok, button hitung Si dan VI, button hasil, hapus dan keluar aktif semua.

Pada form hasil ini dimulai dengan memilih button tampil data, maka akan tampil semua penilaian dari alternatif. Setelah itu klik button ok, maka akan menampilkan nilai bobot dari masing-masing kriteria.

Klik button hitung nilai Si dan Vi maka akan tampil hasil nilai Si dan Vi dari setiap alternative. Button hasil digunakan untuk mengurutkan nilai Si dan Vi terbesar ke yang terkecil dari alternatif yang ada.

Button hapus digunakan untuk menghaus data dan button keluar digunakan untuk menutup form hasil dan kembali ke menu utama. Berikut adalah tampilan dari form hasil.

The screenshot displays the 'HASIL PENILAIAN' (Evaluation Results) form. It is divided into several sections:

- TAMPIL DATA**: A table showing evaluation data for alternatives (ALTR001 to ALTR004) across criteria (K1 to K5).
- Kode Kriteria**: A table showing criteria weights (K1 to K5) and their types (Benefit, Cost).
- NILAI SI DAN VI**: A table showing calculated Si and Vi values for each alternative.
- PERINGKAT**: A table showing the final ranking of alternatives based on Si and Vi values.

Buttons for 'TAMPIL DATA', 'Hitung nilai Si dan Vi', 'OK', 'HAPUS', and 'Tutup' are visible. The 'PERINGKAT' table shows the following data:

Kode Alternative	Nilai Si	Nilai Vi
ALTR006	4.781762498950186	0.18801150884025608
ALTR006	4.781762498950186	0.18801150884025608
ALTR001	4.3734482957731124	0.17195722562625654
ALTR001	4.3734482957731124	0.17195722562625654
ALTR002	4.229485053762256	0.16629681351797254
ALTR002	4.229485053762256	0.16629681351797254
ALTR004	4.136152028136661	0.16262710324349802

Gambar 8. Form Hasil

Source Code Form Hasil

```
package spk_new;

import java.awt.event.KeyEvent;
import java.sql.*;
import java.util.logging.Level;
import java.util.List;
import java.util.ArrayList;
import java.util.logging.Logger;
import javax.swing.*;
```

```

import javax.swing.table.DefaultTableModel;
import sun.misc.Cleaner;
import java.lang.Math;
import javax.swing.DefaultListModel;
import java.text.DecimalFormat;
public class hasil extends javax.swing.JFrame {
private DefaultTableModel DftTabMode;
koneksi kon=new koneksi();
private Object [][] dtnilai=null;
private Object [][] dt2=null;
private Object [][] dt3=null;
public hasil() {
    initComponents();
    kon.setKoneksi();
    bacatabelkriteria(); }
private void bacatabel(){
    Object[] Baris={"Kode Nilai","Kode Alternatif","K1","K2","K3","K4","K5"};
    DftTabMode=new DefaultTableModel(null,Baris);
    tabeltmpl.setModel(DftTabMode);
    try{
        String sql="Select *From nilai_alternatif order by kdnilai";
        kon.rs=kon.st.executeQuery(sql);
        ResultSetMetaData m=kon.rs.getMetaData();
        int kolom=m.getColumnCount();
        int baris=0;
        while(kon.rs.next()){
            baris=kon.rs.getRow();
        }
        dtnilai=new Object[baris][kolom];
        int x=0;
        kon.rs.beforeFirst();
        while(kon.rs.next()){
            dtnilai[x][0]=kon.rs.getString("kdnilai");

```

```

        dtnilai[x][1]=kon.rs.getString("kdalternative");
        dtnilai[x][2]=kon.rs.getString("K1");
        dtnilai[x][3]=kon.rs.getString("K2");
        dtnilai[x][4]=kon.rs.getString("K3");
        dtnilai[x][5]=kon.rs.getString("K4");
        dtnilai[x][6]=kon.rs.getString("K5");
        x++;    }
    tabeltmpl.setModel(new DefaultTableModel(dtnilai,Baris));
}    catch(SQLException e){
    JOptionPane.showMessageDialog(null, e);    } }

private void hapusdt(){
    try{
        String sql="Delete from hasil";
        kon.st.executeUpdate(sql);
        JOptionPane.showMessageDialog(null,"Data berhasil dihapus");
        bacatabelhasil();
        kon.st.close();
    }    catch(SQLException e){
        JOptionPane.showMessageDialog(null, e);    } }

private void bacatabelhasil(){
    Object[] Baris={"Kode Alternative","Nilai Si","Nilai Vi"};
    DftTabMode=new DefaultTableModel(null,Baris);
    thasil.setModel(DftTabMode);
    try{
        String sql="Select *From hasil order by nilaivi DESC";
        kon.rs=kon.st.executeQuery(sql);
        ResultSetMetaData m=kon.rs.getMetaData();
        int kolom=m.getColumnCount();
        int baris=0;
        while(kon.rs.next()){
            baris=kon.rs.getRow();
        }
        dt3=new Object[baris][kolom];
    }
}

```

```

int x=0;
kon.rs.beforeFirst();
while(kon.rs.next()){
    dt3[x][0]=kon.rs.getString("kdAlternative");
    dt3[x][1]=kon.rs.getString("nilaisi");
    dt3[x][2]=kon.rs.getString("nilaivi");
    x++; }
thasil.setModel(new DefaultTableModel(dt3,Baris));
} catch(SQLException e){
    JOptionPane.showMessageDialog(null, e); } }

private void bacatabelkriteria(){
    Object[] Baris={"Kode Kriteria","Nama Kriteria","Jenis","Bobot"};
    DftTabMode=new DefaultTableModel(null,Baris);
    tabel2.setModel(DftTabMode);
    try{
        String sql="Select *From kriteria order by kdkriteria";
        kon.rs=kon.st.executeQuery(sql);
        ResultSetMetaData m=kon.rs.getMetaData();
        int kolom=m.getColumnCount();
        int baris=0;
        while(kon.rs.next()){
            baris=kon.rs.getRow();
        }
        dt2=new Object[baris][kolom];
        int x=0;
        kon.rs.beforeFirst();
        while(kon.rs.next()){
            dt2[x][0]=kon.rs.getString("kdkriteria");
            dt2[x][1]=kon.rs.getString("nmkriteria");
            dt2[x][2]=kon.rs.getString("jns");
            dt2[x][3]=kon.rs.getString("bobot");
            x++;
        } tabel2.setModel(new DefaultTableModel(dt2,Baris));
    }

```

```

    } catch(SQLException e){
        JOptionPane.showMessageDialog(null, e);    } }
private void simpandt(){
    try{
        kon.setKoneksi();
        int jumlah=cb.getRowCount();
        for (int q=0; q<jumlah;q++){
            String sql="INSERT INTO hasil(kdalternative,nilaisi,nilaivi)values("
                +"""+cb.getValueAt(q,0)+""","
                +"""+cb.getValueAt(q,1)+""","
                +"""+cb.getValueAt(q,2)+""")";
            kon.st.executeUpdate(sql);
        } int q=0;
    }catch (Exception e){    } }
private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
    bacatabel(); }
private void jButton5ActionPerformed(java.awt.event.ActionEvent evt) {
    dispose(); }
private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
    String hasil=tabel2.getValueAt(0, 3).toString();
    wk1.setText(hasil);
    String hasil2=tabel2.getValueAt(1, 3).toString();
    wk2.setText(hasil2);
    String hasil3=tabel2.getValueAt(2, 3).toString();
    wk3.setText(hasil3);
    String hasil4=tabel2.getValueAt(3, 3).toString();
    wk4.setText(hasil4);
    String hasil5=tabel2.getValueAt(4, 3).toString();
    wk5.setText(hasil5);
    String hsl=tabel2.getValueAt(0, 2).toString();
    jns1.setText(hsl);
    String hsl2=tabel2.getValueAt(1, 2).toString();
    jns2.setText(hsl2);

```

```

String hsl3=tabel2.getValueAt(2, 2).toString();
jns3.setText(hsl3);
String hsl4=tabel2.getValueAt(3, 2).toString();
jns4.setText(hsl4);
String hsl5=tabel2.getValueAt(4, 2).toString();
jns5.setText(hsl5); }

private void jButton3ActionPerformed(java.awt.event.ActionEvent evt) {
    int w1,w2,w3,w4,w5,a1,a2,a3,a4,a5,b1,b2,b3,b4,b5,c1,c2,c3,c4,c5,d1,d2,d3,d4,d5,
        e1,e2,e3,e4,e5,f1,f2,f3,f4,f5;
    double jml,k1,k2,k3,k4,k5,s1,s2,s3,s4,s5,s6,t1,v1,v2,v3,v4,v5,v6;
    w1=Integer.parseInt(wk1.getText());
    w2=Integer.parseInt(wk2.getText());
    w3=Integer.parseInt(wk3.getText());
    w4=Integer.parseInt(wk4.getText());
    w5=Integer.parseInt(wk5.getText());
    jml=w1+w2+w3+w4+w5;
    k1= (w1/jml);
    k2= (w2/jml);
    k3= (w3/jml);
    k4= (w4/jml);
    k5= (w5/jml);
    a1=Integer.parseInt(tabeltmpl.getValueAt(0, 2).toString()) ;
    a2=Integer.parseInt(tabeltmpl.getValueAt(0, 3).toString()) ;
    a3=Integer.parseInt(tabeltmpl.getValueAt(0, 4).toString()) ;
    a4=Integer.parseInt(tabeltmpl.getValueAt(0, 5).toString()) ;
    a5=Integer.parseInt(tabeltmpl.getValueAt(0, 6).toString()) ;
    b1=Integer.parseInt(tabeltmpl.getValueAt(1, 2).toString()) ;
    b2=Integer.parseInt(tabeltmpl.getValueAt(1, 3).toString()) ;
    b3=Integer.parseInt(tabeltmpl.getValueAt(1, 4).toString()) ;
    b4=Integer.parseInt(tabeltmpl.getValueAt(1, 5).toString()) ;
    b5=Integer.parseInt(tabeltmpl.getValueAt(1, 6).toString()) ;
    c1=Integer.parseInt(tabeltmpl.getValueAt(2, 2).toString()) ;
    c2=Integer.parseInt(tabeltmpl.getValueAt(2, 3).toString()) ;

```

```

c3=Integer.parseInt(tabeltmpl.getValueAt(2, 4).toString()) ;
c4=Integer.parseInt(tabeltmpl.getValueAt(2, 5).toString()) ;
c5=Integer.parseInt(tabeltmpl.getValueAt(2, 6).toString()) ;
d1=Integer.parseInt(tabeltmpl.getValueAt(3, 2).toString()) ;
d2=Integer.parseInt(tabeltmpl.getValueAt(3, 3).toString()) ;
d3=Integer.parseInt(tabeltmpl.getValueAt(3, 4).toString()) ;
d4=Integer.parseInt(tabeltmpl.getValueAt(3, 5).toString()) ;
d5=Integer.parseInt(tabeltmpl.getValueAt(3, 6).toString()) ;
e1=Integer.parseInt(tabeltmpl.getValueAt(4, 2).toString()) ;
e2=Integer.parseInt(tabeltmpl.getValueAt(4, 3).toString()) ;
e3=Integer.parseInt(tabeltmpl.getValueAt(4, 4).toString()) ;
e4=Integer.parseInt(tabeltmpl.getValueAt(4, 5).toString()) ;
e5=Integer.parseInt(tabeltmpl.getValueAt(4, 6).toString()) ;
f1=Integer.parseInt(tabeltmpl.getValueAt(5, 2).toString()) ;
f2=Integer.parseInt(tabeltmpl.getValueAt(5, 3).toString()) ;
f3=Integer.parseInt(tabeltmpl.getValueAt(5, 4).toString()) ;
f4=Integer.parseInt(tabeltmpl.getValueAt(5, 5).toString()) ;
f5=Integer.parseInt(tabeltmpl.getValueAt(5, 6).toString()) ;
s1=(Math.pow(a1, k1))*(Math.pow(a2, k2))*(Math.pow(a3, k3))*(Math.pow(a4,
k4))*(Math.pow(a5, k5));
s2=(Math.pow(b1, k1))*(Math.pow(b2, k2))*(Math.pow(b3, k3))*(Math.pow(b4,
k4))*(Math.pow(b5, k5));
s3=(Math.pow(c1, k1))*(Math.pow(c2, k2))*(Math.pow(c3, k3))*(Math.pow(c4,
k4))*(Math.pow(c5, k5));
s4=(Math.pow(d1, k1))*(Math.pow(d2, k2))*(Math.pow(d3, k3))*(Math.pow(d4,
k4))*(Math.pow(d5, k5));
s5=(Math.pow(e1, k1))*(Math.pow(e2, k2))*(Math.pow(e3, k3))*(Math.pow(e4,
k4))*(Math.pow(e5, k5));
s6=(Math.pow(f1, k1))*(Math.pow(f2, k2))*(Math.pow(f3, k3))*(Math.pow(f4,
k4))*(Math.pow(f5, k5));
ttl=s1+s2+s3+s4+s5+s6;
v1=s1/ttl;
v2=s2/ttl;

```

v3=s3/ttl;

v4=s4/ttl;

v5=s5/ttl;

v6=s6/ttl;

String al1=tabeltmpl.getValueAt(0,1).toString();

String al2=tabeltmpl.getValueAt(1, 1).toString();

String al3=tabeltmpl.getValueAt(2, 1).toString();

String al4=tabeltmpl.getValueAt(3, 1).toString();

String al5=tabeltmpl.getValueAt(4, 1).toString();

String al6=tabeltmpl.getValueAt(5, 1).toString();

cb.setValueAt(tabeltmpl.getValueAt(0,1).toString(), 0, 0);

cb.setValueAt(tabeltmpl.getValueAt(1,1).toString(), 1, 0);

cb.setValueAt(tabeltmpl.getValueAt(2,1).toString(), 2, 0);

cb.setValueAt(tabeltmpl.getValueAt(3,1).toString(), 3, 0);

cb.setValueAt(tabeltmpl.getValueAt(4,1).toString(), 4, 0);

cb.setValueAt(tabeltmpl.getValueAt(5,1).toString(), 5, 0);

cb.setValueAt(s1, 0, 1);

cb.setValueAt(s2, 1, 1);

cb.setValueAt(s3, 2, 1);

cb.setValueAt(s4, 3, 1);

cb.setValueAt(s5, 4, 1);

cb.setValueAt(s6, 5, 1);

cb.setValueAt(v1, 0, 2);

cb.setValueAt(v2, 1, 2);

cb.setValueAt(v3, 2, 2);

cb.setValueAt(v4, 3, 2);

cb.setValueAt(v5, 4, 2);

cb.setValueAt(v6, 5, 2);

Total.setText(String.valueOf(ttl));

simpandt();

}

```
private void bhasilActionPerformed(java.awt.event.ActionEvent evt) {  
    bacatabelhasil(); }  
private void jButton4ActionPerformed(java.awt.event.ActionEvent evt) {  
    hapusdt(); }
```