# **PRAKTIKUM**

**PEMROGRAMAN BERORIENTASI OBJEK**

**LAPORAN PROYEK AKHIR**

**PENDATAAN IMIGRASI**

****

**DISUSUN OLEH :**

|  |  |  |
| --- | --- | --- |
| **ANGGOTA** | **:** | **ISNAINI KHAIRIAH 123180051**  **RIVANKA DESYA FAWWAZ A 123180054** |
| **KELAS/PLUG** | **:** |  |
| **NAMA ASISTEN** | **:** |  |
|  |  |  |

### PROGRAM STUDI INFORMATIKA

### JURUSAN TEKNIK INFORMATIKA

### FAKULTAS TEKNIK INDUSTRI

### UNIVERSITAS PEMBANGUNAN NASIONAL ”VETERAN”

#### YOGYAKARTA

#### 2021

# **HALAMAN PENGESAHAN**

**LAPORAN PROYEK AKHIR**

****

Disusun oleh :

*Isnaini Khairiah*  123190004

*Rivanka Desya Fawwaz Assyraaj* 123190018

Telah Diperiksa dan Disetujui oleh Asisten Praktikum Implementasi Basis Data

Pada Tanggal : …………………

|  |  |
| --- | --- |
| **Menyetujui,**  **Asisten Praktikum**  **Monica Ameiliana Andista M**  **NIM. 124180041** | **Asisten Praktikum**  **Amira Salsabila**  **NIM. 124180067** |

|  |
| --- |
| **Mengetahui,**  **Ka. Lab. Geoinformatka**  **Budi Santosa, S.Si.,M.T.**  **NIK. 2 7009 02 0234 1** |

# **KATA PENGANTAR**

Puji syukur kami panjatkan kepada Tuhan Yang Maha Esa yang senantiasa mencurahkan rahmat dan hidayah-Nya sehingga kami dapat menyelesaikan praktikum Pemrograman Berorientasi Objek serta laporan proyek akhir praktikum yang berjudul Pendataan Imigrasi. Adapun laporan ini berisi tentang proyek akhir yang kami pilih dari hasil pembelajaran selama praktikum berlangsung.

Tidak lupa ucapan terima kasih kepada asisten dosen yang selalu membimbing dan mengajari kami dalam melaksanakan praktikum dan dalam menyusun laporan ini. Laporan ini masih sangat jauh dari kesempurnaan, oleh karena itu kritik serta saran yang membangun kami harapkan untuk menyempurnakan laporan akhir ini.

Atas perhatian dari semua pihak yang membantu penulisan ini, kami ucapkan terima kasih. Semoga laporan ini dapat dipergunakan seperlunya.

Yogyakarta, Juli 2020

Penyusun

# **DAFTAR ISI**

[**HALAMAN PENGESAHAN** ii](#_Toc25323565)

[**KATA PENGANTAR** ii](#_Toc25323566)

[**DAFTAR ISI** ii](#_Toc25323567)

[**BAB I PENDAHULUAN** 2](#_Toc25323568)

[1.1 Latar Belakang 2](#_Toc25323569)

[1.2 Tujuan 2](#_Toc25323570)

[**BAB II PEMBAHASAN** 2](#_Toc25323571)

[2.1 Dasar Teori 2](#_Toc25323572)

[2.2 Deskripsi Umum Proyek Akhir 2](#_Toc25323573)

[2.3 Inti Pembahasan 2](#_Toc25323574)

[**BAB III JADWAL PENGERJAAN DAN PEMBAGIAN TUGAS** 2](#_Toc25323575)

[3.1 Jadwal Pengerjaan 2](#_Toc25323576)

[3.2 Pembagian Tugas 2](#_Toc25323577)

[**BAB IV KESIMPULAN DAN SARAN** 2](#_Toc25323578)

[4.1 Kesimpulan 2](#_Toc25323579)

[4.2 Saran 2](#_Toc25323580)

[**DAFTAR PUSTAKA** 2](#_Toc25323581)

# **BAB I**

**PENDAHULUAN**

## **Latar Belakang**

Dengan semakin berkembangnya ilmu pengetahuan dan teknologi, serta semakin kompleksnya permasalahan-permasalahan yang ditemui pada segi kehidupan, membuat manusia untuk senantiasa menyelesaikan permasalahan yang dihadapinya dengan memanfaatkan kecanggihan teknologi. Komputer sebagai salah satu alat bantu pengambilan keputusan, sehingga nantinya akan mendapatkan hasil yang lebih efektif, efisien, dan berdayaguna tinggi.

Pemanfaatan teknologi terutama mengacu pada teknologi komputer tersebut salah satunya adalah dalam menyelesaikan permasalahan kependudukan di dalam kantor Pendataan Imigrasi, terutama yang memerlukan efektivitas dalam pekerjaan dan dalam pembuatan laporan yang diperlukan. Dengan bantuan program komputer pada proses penyelesaian dan pembuatan laporan kependudukan tersebut serta dapat memberikan laporan secara tepat cepat dan mengenai informasi pelayanan kependudukan dalam kantor tersebut. Untuk meningkatkan efektivitas dan efisiensi dalam informasi kependudukan maka perlu dipikirkan langkah-langkah penerapan sistem baru yang diharapkan akan membantu menyelesaikan tugas-tugas yang berkaitan dengan data-data kependudukan sehingga setiap pekerjaan dapat diselesaikan dengan baik, efektif, dan efisien. Berdasarkan uraian diatas maka ditetapkan kantor ini sebagai obyek dalam penulisan tugas akhir dengan judul “Pendataan Imigrasi”.

## **Tujuan**

Berkaitan dengan rumusan masalah pada penelitian ini, tujuan penelitian adalah membuat sistem informasi jasa pelayanan Pendataan Imigrasi kependudukan. Berbasis Program Berorientasi Objek

# **BAB II**

**PEMBAHASAN**

## **2.1 Dasar Teori**

**2.1.1 Bahasa Pemrograman Java**

Java adalah bahasa pemrograman berorientasi objek. Sebagai bahasa pemrograman berorientasi objek, Java menggunakan kelas-kelas untuk mengorganisasikan kode program dalam modul-modul logikal.

**2.1.2 Pemrograman Berorientasi Objek**

Dalam bahasa pemrograman berorientasi objek, program dibagi menjadi objek-objek yang saling berinteraksi satu sama lain. Objek tersebut dapat memiliki wujud yang nyata, maupun tidak nyata (abstrak). Konsep-konsep dasar dari pemrograman berorientasi objek adalah inheritance (pewarisan), encapsulation (pengkapsulan), dan polymorphism.

**2.1.3 *Overriding***

Overriding adalah kemampuan suatu subclass untuk menulis code milik method() yang abstract milik super class, atau memodifikasi isi attribute atau isi method() yang tidak abstract, tidak private, tidak final, dan non static milik super class. Method-method yg overloading wajib berada di dalam class yg sama atau dalam satu keluarga (sebagian method()-method() tersebut di dalam superclass dan yg lain di dalam subclass).

**2.1.4 *Overloading***

Overriding adalah kemampuan suatu subclass untuk menulis code milik method() yang abstract milik super class, atau memodifikasi isi attribute atau isi method() yang tidak abstract, tidak private, tidak final, dan non static milik super class.

Modifikasi modifier akses pada attribute, method(), dan constructor() miliki super class, boleh dilakukan dengan cara memperluas daerah aksesnya.

**2.1.5 *Encapsulation***

Encapsulation atau pengkapsulan digunakan untuk menyembunyikan atribut atau method pada suatu objek, dari objek lainnya, atau dengan kata lain yakni Information hiding. Pengkapsulan dilakukan dengan memberikan modifier private pada atribut maupun method. Kemudian, untuk memberikan nilai pada atribut private, digunakan method ‘setter’, dan untuk mengambil nilai dari atribut private, digunakan method ‘getter’. Method setter dan getter ini sama seperti method biasa, hanya berbeda pada pemanfaatan yang dituliskan programmernya saja.

**2.1.6 *Inheritance***

Inheritance adalah penurunan atribut dan method dari suatu kelas induk (superclass) ke kelas anak (subclass). Pewarisan bersifat menyeluruh dan beruntun, sehingga jika A merupakan superclass dari B, B merupakan superclass dari C, maka A juga merupakan superclass dari C. Jika ada atribut atau method dari C dan B yang ingin dimodifikasi, kita hanya perlu memodifikasi atribut/method pada A. Pernyataan extends digunakan untuk melakukan pewarisan ini.

**2.1.7 *Interface***

Interface merupakan suatu kelas yang berisi method-method tanpa implementasi, namun tanpa modifier abstract. Apabila terdapat atribut, maka atribut tersebut berlaku sebagai konstanta (static final). Interface bersifat seperti kontrak yang bisa digunakan oleh setiap kelas yang nyata, dan penggunaannya bukan diwariskan (extends) melainkan diimplementasikan (implements).

**2.1.8 *Exception***

Exception merupakan suatu objek Error khusus yang dibuat ketika terjadi kesalahan pada suatu program. Setelah suatu objek exception dibuat oleh java, objek tersebut dikirimkan ke program, proses ini disebut throwing an exception. Objek exception tersebut kemudian diterima oleh program, dan penanganan exception selanjutnya terserah pada program.

Penanganan exception dalam java dibuat dengan menuliskan kode program exception-handling pada Source code. Exception-handling berguna untuk mencegah program menjadi “Hang”. Code program exception handling secara umum terdiri dari 3 bagian yaitu try, catch dan finally.

**2.1.9 SWING**

GUI merupakan suatu metode untuk antar-muka komputer berbasis grafis. GUI digunakan dalam pembuatan program aplikasi dengan mempertimbangkan dua aspek yaitu keindahan tampilan dan kemudahan penggunaan program. Swing adalah kumpulan class–class yang digunakan untuk membuat GUI. Swing adalah GUI yang lightweight, karena lebih multiplatform dan sedikit menggunakan Native class.

**2.1.10 Java *Database Connectivity***

JDBC (Java *DataBase Connectivity*) merupakan teknologi Java yang populer sebagai standar pengaksesan database. Java tidak membedakan cara pemrograman database yang satu dengan lainnya. Sebagai contoh: cara Java mengolah tabel di Oracle dan MySQL sama saja, yang berbeda hanyalah alamat URL koneksi dan driver JDBC-nya. Sehingga kita dapat membuat sebuah program yang sama di Java untuk mengolah database yang ada di MySQL, Oracle, SQL Server dan lain-lain asalkan nama dan struktur tabelnya sama.

**2.1.3 *Model View Controller***

Model View Control pertama kali diperkenalkan oleh Trygve Reenskaug, seorang pengembang perangkat lunak Smalltalk pada Xerox Palo Alto Research Center pada tahun 1979. Konsep ini memisahkan antara akses data, logika bisnis, dan antarmuka pada perangkat lunak, secara detail MVC dibagi 3, yaitu :

1. Model, merupakan bagian dari MVC yang bertugas menghubungkan langsung antara tugas-tugas dengan proses di Database (Create, Read, Update, Delete). Sebagai contoh pada Java, dibuat kelas sendiri antara koneksi dengan query akses data ke database. Jadi untuk menggunakan fungsi Insert, Update, Delete, dan Select kita tinggal memanggil method-method yang menangani hal tersebut.
2. View, merupakan bagian dari MVC yang bertugas menghubungkan antara pengguna dengan tugas perangkat lunak, dengan kata lain View menangani bagian Interface atau Antarmuka untuk mempermudah si pengguna menggunakan perangkat lunak.
3. Controller, menjadi tulang punggung perangkat lunak. Controller menghubungkan antara View dan Model yang meng-handle semua proses-proses yang terjadi didalamnya. Sebagai contoh kasus User Login, tugas Controller adalah mencocokkan data yang di-inputkan user dengan data yang ada di database

## **2.2 Deskripsi Umum Proyek Akhir**

Program ini digunakan untu mendata para imigran yang datang dari daerah tertentu dengan menggunakan program Berorientasi Objek. Dengan begitu Pendataan penduduk tidak dilakukan lagi secara manual tetapi secara dinamis menggunakan program.Program ini terhubung dengan database yang ada di localhost dengan nama databse imigran. Database imigran mempunyai 2 entitas yaitu entitas admin dan imigran. Entitas admin untuk menyimpan info login admin/operator program dan entitas imigran digunakan untuk menyimpan data imigran.

Program ini diawali dengan menu login. Menu login ini digunakan operator untuk masuk kehalaman utama dalam program. Data dari admin dimasukan dalam database dengan entitas admin. Admin/operator program harus memasukkan username dan password yang ada didalam database. Jika username dan password salah maka akan muncul pesan eror atau username atau password salah. Jika login berhasil maka program akan mengarahkan ke menu utama yaitu pendataan imigrasi. Di pendataan imigrasi ini ada fitur input imigran, ubah data imigran, cari imigran, hapus data imigran, dan reset inputan.

## **2.3 Inti Pembahasan**

## **2.3.1 *Listing* Program**

|  |
| --- |
| public class Connector{  String DBurl = "jdbc:mysql://localhost/imigrasi";  String DBusername = "root";  String DBpassword = "";  public Connection koneksi;  public Statement statement;  public Connector() {  try{  Class.forName("com.mysql.jdbc.Driver");  koneksi = (Connection) DriverManager.getConnection(DBurl,DBusername,DBpassword);  System.out.println("Koneksi Berhasil");  }catch(ClassNotFoundException | SQLException ex){  System.out.println("Koneksi gagal");  }  }  Statement createStatement() {  throw new UnsupportedOperationException("Not supported yet."); //To change body of generated methods, choose Tools | Templates.  }  } |

***Listing* Program 2.1** Connector database

|  |
| --- |
| package Controller;  import DAO.DAOImigran;  import DAOInterface.IDAOImigran;  import Model.Imigran;  import Model.TabelModelImigran;  import View.Menu;  import java.util.List;  import javax.swing.JOptionPane;  import java.text.ParseException;  import java.text.SimpleDateFormat;  public class ControllerImigran {    public ControllerImigran(Menu frmImigran){  this.frmImigran = frmImigran;  iImigran = new DAOImigran();  }    // menampilkan data di dalam tabel gui  public void isiTable(){    listImigran = iImigran.getAll();  TabelModelImigran tabelImg = new TabelModelImigran(listImigran);  frmImigran.getTableData().setModel(tabelImg);  }  public void insert(){    Imigran a = new Imigran();  a.setKode(frmImigran.gettxtKode().getText());  a.setNama(frmImigran.gettxtNama().getText());  a.setAlamat(frmImigran.gettxtAlamat().getText());  a.setWarga(frmImigran.gettxtWarga().getText());  a.setJenis\_kelamin(frmImigran.getcomboJenis().getSelectedItem().toString()); //khusus selected karena menggunakan combo box lalu di ubah menjadi string  String tanggal1 = frmImigran.getdateTanggal().getDate().toString().substring(4, 10);  String tanggal2 = frmImigran.getdateTanggal().getDate().toString().substring(23, 28);  a.setTanggal(tanggal1 + ","+ tanggal2); // khusus dateformat karena menggunakan datechooser lalu di ubah menjadi string  iImigran.insert(a); //perintah insert  JOptionPane.showMessageDialog(null, "input sukses"); //alert jika berhasil  }  public void reset(){ //untuk mereset saat setelah menyimpan atau mengubah data agar kolom form kosong    if(!frmImigran.gettxtKode().isEnabled()){ //fungsi pengendalian jika kolom input kode terkunci  frmImigran.gettxtKode().setEnabled(true); //perintah untuk membuka kembali kunci input kode paspor  }  frmImigran.gettxtKode().setText("");  frmImigran.gettxtNama().setText("");  frmImigran.gettxtAlamat().setText("");  frmImigran.gettxtWarga().setText("");  frmImigran.getdateTanggal().setDate(null);  }  //fungsi dari throws Parse Exception adalah agar fungsi parse mengubah string ke date berhasil  public void isiField(int row) throws ParseException{ //untuk actionListener mouse clicked saat salah satu data di tabel maka akan muncul di form data yang telah di pilih  frmImigran.gettxtKode().setText(listImigran.get(row).getKode());  frmImigran.gettxtKode().setEnabled(false); //berfungsi untuk mengunci primary key / kode paspor agar tidak dapat di ubah  frmImigran.gettxtNama().setText(listImigran.get(row).getNama());  frmImigran.gettxtAlamat().setText(listImigran.get(row).getAlamat()); frmImigran.gettxtWarga().setText(listImigran.get(row).getWarga();  SimpleDateFormat sdf = new SimpleDateFormat("MMM d, y");  String date = listImigran.get(row).getTanggal();  frmImigran.getdateTanggal().setDate(sdf.parse(date));  frmImigran.getcomboJenis().setSelectedItem(listImigran.get(row).getJenis\_kelamin());  }  public void update(){  Imigran a = new Imigran();  a.setKode(frmImigran.gettxtKode().getText());  a.setNama(frmImigran.gettxtNama().getText());  a.setAlamat(frmImigran.gettxtAlamat().getText());  a.setWarga(frmImigran.gettxtWarga().getText());  a.setJenis\_kelamin(frmImigran.getcomboJenis().getSelectedItem().toString()); //khusus selected karena menggunakan combo box lalu di ubah menjadi string  String tanggal1 = frmImigran.getdateTanggal().getDate().toString().substring(4, 10);  String tanggal2 = frmImigran.getdateTanggal().getDate().toString().substring(23, 28);  a.setTanggal(tanggal1 + ","+ tanggal2); // khusus dateformat karena menggunakan datechooser lalu di ubah menjadi string  iImigran.update(a); //perintah insert  JOptionPane.showMessageDialog(null, "Update sukses"); //alert jika berhasil  }  public void delete(){  iImigran.delete(frmImigran.gettxtKode().getText()); //perintah hapus  JOptionPane.showMessageDialog(null, "Hapus sukses"); //alert jika berhasil  }    public void cari(){  listImigran = iImigran.getAllByNameKode(frmImigran.gettxtCari().getText()); //mengambil nama dari form search //dan karena return dari DAOImigran  TabelModelImigran tabelImg = new TabelModelImigran(listImigran);  frmImigran.getTableData().setModel(tabelImg);  }      Menu frmImigran;  IDAOImigran iImigran;  List<Imigran>listImigran;    } |

***Listing* Program 2.2** ControllerImigran.java

|  |
| --- |
| package DAO;  import Helper.Connector;  import DAOInterface.IDAOImigran;  import Model.Imigran;  import com.mysql.jdbc.PreparedStatement;  import java.beans.Statement;  import java.sql.Connection;  import java.sql.ResultSet;  import java.util.List;  import java.util.ArrayList;  import java.sql.SQLException;  import java.sql.ResultSet;  import java.util.logging.Level;  import java.util.logging.Logger;  public class DAOImigran implements IDAOImigran{  Connector connector = new Connector();  public DAOImigran()  {  Connector connector = new Connector();  }  //read data  @Override  public List<Imigran> getAll() {  List<Imigran> listImigran =null;  try{  listImigran = new ArrayList<Imigran>();  connector.statement = connector.koneksi.createStatement();  String queryRead= "select \* from imigran";  ResultSet result = connector.statement.executeQuery(queryRead);  while(result.next()){  Imigran img = new Imigran();  img.setKode(result.getString("kode"));  img.setNama(result.getString("nama"));  img.setAlamat(result.getString("alamat"));  img.setWarga(result.getString("warga"));  img.setJenis\_kelamin(result.getString("jenis\_kelamin")); img.setTanggal(result.getString("tanggal"));  listImigran.add(img);  }  }catch(SQLException ex){  System.out.println("error");    return listImigran;  }  //write data  @Override  public void insert(Imigran a) {  try{  String queryWrite = "insert into imigran (kode,nama,alamat,warga,jenis\_kelamin,tanggal) values ('"+a.getKode()+"','"+a.getNama()+"','"+a.getAlamat()+"','"+a.getWarga()+"','"+a.getJenis\_kelamin()+"','"+a.getTanggal().toString()+"');";  connector.statement = connector.koneksi.createStatement();  connector.statement.executeUpdate(queryWrite);  }catch(SQLException e){  System.out.println("gagal input: " + e);//error ketika di awal  }  }  @Override  public void update(Imigran a) {  try {  String kodeUpdate = a.getKode();  String queryUpdate = "update imigran set kode='"+a.getKode()+"', nama='"+a.getNama()+"', alamat='"+a.getAlamat()+"', warga='"+a.getWarga()+"', jenis\_kelamin='"+a.getJenis\_kelamin()+"', tanggal='"+a.getTanggal().toString()+"' where kode='"+kodeUpdate+"';";  connector.statement = connector.koneksi.createStatement();  connector.statement.executeUpdate(queryUpdate);    } catch (Exception e) {  System.out.println("gagal update: " + e);  }  }  @Override  public void delete(String kode) {  try {  String queryDelete = "delete from imigran where kode='"+kode+"';";  connector.statement = connector.koneksi.createStatement();  connector.statement.executeUpdate(queryDelete);  } catch (Exception e) {  System.out.println("gagal delete: " + e);  }  }  @Override  public List<Imigran> getAllByNameKode(String cari) {  List<Imigran> listImigran =null;  try{  listImigran = new ArrayList<Imigran>();  connector.statement = connector.koneksi.createStatement();  String temp = "%"+cari+"%";  String querySearch = "select \* from imigran where nama like '"+temp+"' or kode like '"+temp+"';";  ResultSet result = connector.statement.executeQuery(querySearch);  while(result.next()){  Imigran img = new Imigran();  img.setKode(result.getString("kode"));  img.setNama(result.getString("nama"));  img.setAlamat(result.getString("alamat"));  img.setWarga(result.getString("warga"));  img.setJenis\_kelamin(result.getString("jenis\_kelamin") img.setTanggal(result.getString("tanggal"));  listImigran.add(img);  }  }catch(SQLException ex){  System.out.println("error");  }  return listImigran; //digunakan saat controller  }  } |

***Listing* Program 2.3** DaoImigran.java

|  |
| --- |
| package Model;  public class Imigran {  public String getNama() {  return nama;  }  public void setNama(String nama) {  this.nama = nama;  }  public String getAlamat() {  return alamat;  }  public void setAlamat(String alamat) {  this.alamat = alamat;  }  public String getTanggal() {  return tanggal;  }  public void setTanggal(String tanggal) {  this.tanggal = tanggal;  }  public String getKode() {  return kode;  }  public void setKode(String kode) {  this.kode = kode;  }  public String getWarga() {  return warga;  }  public void setWarga(String warga) {  this.warga = warga;  }  public String getJenis\_kelamin() {  return jenis\_kelamin;  }  public void setJenis\_kelamin(String jenis\_kelamin) {  this.jenis\_kelamin = jenis\_kelamin;  }  private String nama, alamat, tanggal, kode,warga,jenis\_kelamin;  } |

***Listing* Program 2.4** Imigran.java

|  |
| --- |
| package Model;  import java.util.List;  import javax.swing.table.AbstractTableModel;  public class TabelModelImigran extends AbstractTableModel {  public TabelModelImigran(List<Imigran> listOrder){  this.listImigran=listOrder;  }  @Override  public int getRowCount() {  return this.listImigran.size();  }  @Override  public int getColumnCount() {  return 6;  }  @Override  public String getColumnName(int column){  return switch (column) {  case 0 -> "kode Paspor";  case 1 -> "nama";  case 2 -> "alamat";  case 3 -> "warga";  case 4 -> "jenis kelamin";  case 5 -> "tanggal Lahir";  default -> null;  };  }  @Override  public Object getValueAt(int rowIndex, int columnIndex) {  return switch (columnIndex) {  case 0 -> listImigran.get(rowIndex).getKode();  case 1 -> listImigran.get(rowIndex).getNama();  case 2 -> listImigran.get(rowIndex).getAlamat();  case 3 -> listImigran.get(rowIndex).getWarga();  case 4 -> listImigran.get(rowIndex).getJenis\_kelamin();  case 5 -> listImigran.get(rowIndex).getTanggal();  default -> null;  };  }  List<Imigran> listImigran;  } |

***Listing* Program 2.5** TableModelImigran.java

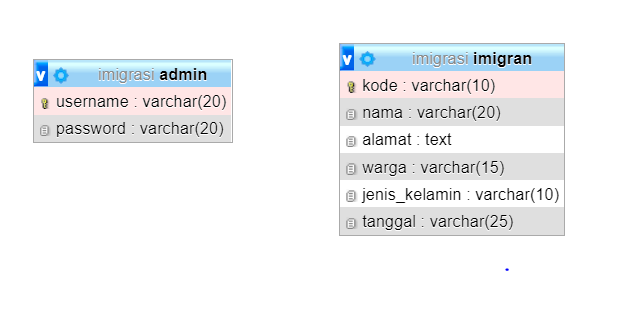
|  |
| --- |
| package View;  import Helper.Connector;  import java.sql.ResultSet;  import java.sql.SQLException;  import java.util.logging.Level;  import java.util.logging.Logger;  public class Login extends javax.swing.JFrame {  public Login() {  initComponents();  }  @SuppressWarnings("unchecked")  // <editor-fold defaultstate="collapsed" desc="Generated Code">  private void initComponents() {  jColorChooser1 = new javax.swing.JColorChooser();  jColorChooser2 = new javax.swing.JColorChooser();  jLabel1 = new javax.swing.JLabel();  jLabel2 = new javax.swing.JLabel();  jLabel4 = new javax.swing.JLabel();  jTextField1 = new javax.swing.JTextField();  jPasswordField1 = new javax.swing.JPasswordField();  jButton1 = new javax.swing.JButton();  jLabel3 = new javax.swing.JLabel();  jPanel1 = new javax.swing.JPanel();  setDefaultCloseOperation(javax.swing.WindowConstants.EXIT\_ON\_CLOSE);  setBackground(new java.awt.Color(153, 153, 255));  getContentPane().setLayout(new org.netbeans.lib.awtextra.AbsoluteLayout());  jLabel1.setFont(new java.awt.Font("News701 BT", 0, 18)); // NOI18N  jLabel1.setText("LOGIN");  getContentPane().add(jLabel1, new org.netbeans.lib.awtextra.AbsoluteConstraints(190, 30, -1, 42));  jLabel2.setFont(new java.awt.Font("News706 BT", 0, 14)); // NOI18N  jLabel2.setText("Username :");  getContentPane().add(jLabel2, new org.netbeans.lib.awtextra.AbsoluteConstraints(50, 110, -1, -1));  jLabel4.setFont(new java.awt.Font("News706 BT", 0, 14)); // NOI18N  jLabel4.setText("Password :");  getContentPane().add(jLabel4, new org.netbeans.lib.awtextra.AbsoluteConstraints(50, 150, -1, -1));  jTextField1.addActionListener(new java.awt.event.ActionListener() {  public void actionPerformed(java.awt.event.ActionEvent evt) {  jTextField1ActionPerformed(evt);  }  });  getContentPane().add(jTextField1, new org.netbeans.lib.awtextra.AbsoluteConstraints(140, 110, 209, 30));  jPasswordField1.addActionListener(new java.awt.event.ActionListener() {  public void actionPerformed(java.awt.event.ActionEvent evt) {  jPasswordField1ActionPerformed(evt);  }  });  getContentPane().add(jPasswordField1, new org.netbeans.lib.awtextra.AbsoluteConstraints(140, 150, 209, 30));  jButton1.setText("Login");  jButton1.addActionListener(new java.awt.event.ActionListener() {  public void actionPerformed(java.awt.event.ActionEvent evt) {  jButton1ActionPerformed(evt);  }  });  getContentPane().add(jButton1, new org.netbeans.lib.awtextra.AbsoluteConstraints(270, 210, 66, 39));  jLabel3.setIcon(new javax.swing.ImageIcon("C:\\Users\\RIVANKA\\Documents\\NetBeansProjects\\Imigrasi\\img\\logo.jpg")); // NOI18N  getContentPane().add(jLabel3, new org.netbeans.lib.awtextra.AbsoluteConstraints(100, 0, 80, 80));  jPanel1.setBackground(new java.awt.Color(255, 255, 255));  getContentPane().add(jPanel1, new org.netbeans.lib.awtextra.AbsoluteConstraints(0, 0, 410, 310));  pack();  }// </editor-fold>  private void jTextField1ActionPerformed(java.awt.event.ActionEvent evt) {  // TODO add your handling code here:  }  private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {  try {  Connector connector = new Connector();  String query = "Select \* from admin where username='"+jTextField1.getText()+"' and password='"+String.valueOf(jPasswordField1.getText())+"'";  connector.statement = connector.koneksi.createStatement();  ResultSet resultSet = connector.statement.executeQuery(query);  if(resultSet.next()){  Menu menu = new Menu();  menu.setVisible(true);  dispose();  }else{    JOptionPane.showMessageDialog(rootPane, "Username Atau Password salah!");  }  } catch (SQLException ex) {  Logger.getLogger(Login.class.getName()).log(Level.SEVERE, null, ex);  }  }  private void jPasswordField1ActionPerformed(java.awt.event.ActionEvent evt) {  // TODO add your handling code here:  }  /\*\*  \* @param args the command line arguments  \*/  public static void main(String args[]) {  /\* Set the Nimbus look and feel \*/  //<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">  /\* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and feel.  \* For details see http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html  \*/  try {  for (javax.swing.UIManager.LookAndFeelInfo info : javax.swing.UIManager.getInstalledLookAndFeels()) {  if ("Nimbus".equals(info.getName())) {  javax.swing.UIManager.setLookAndFeel(info.getClassName());  break;  }  }  } catch (ClassNotFoundException ex) {  java.util.logging.Logger.getLogger(Login.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);  } catch (InstantiationException ex) {  java.util.logging.Logger.getLogger(Login.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);  } catch (IllegalAccessException ex) {  java.util.logging.Logger.getLogger(Login.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);  } catch (javax.swing.UnsupportedLookAndFeelException ex) {  java.util.logging.Logger.getLogger(Login.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);  }  //</editor-fold>  //</editor-fold>  /\* Create and display the form \*/  java.awt.EventQueue.invokeLater(new Runnable() {  public void run() {  new Login().setVisible(true);  }  });  }  // Variables declaration - do not modify  private javax.swing.JButton jButton1;  private javax.swing.JColorChooser jColorChooser1;  private javax.swing.JColorChooser jColorChooser2;  private javax.swing.JLabel jLabel1;  private javax.swing.JLabel jLabel2;  private javax.swing.JLabel jLabel3;  private javax.swing.JLabel jLabel4;  private javax.swing.JPanel jPanel1;  private javax.swing.JPasswordField jPasswordField1;  private javax.swing.JTextField jTextField1;  // End of variables declaration  } |

***Listing* Program 2.6** Login.java

|  |
| --- |
| package View;  import Controller.ControllerImigran;  import com.toedter.calendar.JDateChooser;  import java.text.ParseException;  import java.util.logging.Level;  import java.util.logging.Logger;  import javax.swing.JComboBox;  import javax.swing.JTable;  import javax.swing.JTextArea;  import javax.swing.JTextField;  public class Menu extends javax.swing.JFrame {  public Menu() {  initComponents();  ctImg = new ControllerImigran(this); //view  ctImg.isiTable();// perintah untuk mengisi tabel dengan metode controller  }  /\*\*  \* 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">  private void initComponents() {  jPanel1 = new javax.swing.JPanel();  jLabel1 = new javax.swing.JLabel();  jLabel2 = new javax.swing.JLabel();  txtKode = new javax.swing.JTextField();  jLabel3 = new javax.swing.JLabel();  txtNama = new javax.swing.JTextField();  jLabel4 = new javax.swing.JLabel();  jLabel5 = new javax.swing.JLabel();  txtWarga = new javax.swing.JTextField();  jLabel6 = new javax.swing.JLabel();  jLabel7 = new javax.swing.JLabel();  jScrollPane1 = new javax.swing.JScrollPane();  txtAlamat = new javax.swing.JTextArea();  dateTanggal = new com.toedter.calendar.JDateChooser();  btnSimpan = new javax.swing.JButton();  btnUbah = new javax.swing.JButton();  btnHapus = new javax.swing.JButton();  btnReset = new javax.swing.JButton();  jScrollPane2 = new javax.swing.JScrollPane();  tabelData = new javax.swing.JTable();  jLabel8 = new javax.swing.JLabel();  txtCariNama = new javax.swing.JTextField();  btnCari = new javax.swing.JButton();  comboJenis = new javax.swing.JComboBox<>();  jLabel9 = new javax.swing.JLabel();  setDefaultCloseOperation(javax.swing.WindowConstants.EXIT\_ON\_CLOSE);  jPanel1.setBackground(new java.awt.Color(255, 255, 255));  jPanel1.setLayout(new org.netbeans.lib.awtextra.AbsoluteLayout());  jLabel1.setFont(new java.awt.Font("News706 BT", 0, 18)); // NOI18N  jLabel1.setText("DATABASE IMIGRASI");  jPanel1.add(jLabel1, new org.netbeans.lib.awtextra.AbsoluteConstraints(410, 30, -1, 41));  jLabel2.setFont(new java.awt.Font("Tahoma", 0, 14)); // NOI18N  jLabel2.setText("Kode Paspor :");  jPanel1.add(jLabel2, new org.netbeans.lib.awtextra.AbsoluteConstraints(20, 170, -1, -1));  jPanel1.add(txtKode, new org.netbeans.lib.awtextra.AbsoluteConstraints(140, 170, 181, -1));  jLabel3.setFont(new java.awt.Font("Tahoma", 0, 14)); // NOI18N  jLabel3.setText("Nama : ");  jPanel1.add(jLabel3, new org.netbeans.lib.awtextra.AbsoluteConstraints(20, 210, -1, -1));  jPanel1.add(txtNama, new org.netbeans.lib.awtextra.AbsoluteConstraints(140, 210, 181, -1));  jLabel4.setFont(new java.awt.Font("Tahoma", 0, 14)); // NOI18N  jLabel4.setText("Alamat :");  jPanel1.add(jLabel4, new org.netbeans.lib.awtextra.AbsoluteConstraints(20, 250, -1, -1));  jLabel5.setFont(new java.awt.Font("Tahoma", 0, 14)); // NOI18N  jLabel5.setText("Warga Negara :");  jPanel1.add(jLabel5, new org.netbeans.lib.awtextra.AbsoluteConstraints(20, 370, -1, -1));  jPanel1.add(txtWarga, new org.netbeans.lib.awtextra.AbsoluteConstraints(140, 370, 181, -1));  jLabel6.setFont(new java.awt.Font("Tahoma", 0, 14)); // NOI18N  jLabel6.setText("Jenis kelamin :");  jPanel1.add(jLabel6, new org.netbeans.lib.awtextra.AbsoluteConstraints(20, 410, -1, -1));  jLabel7.setFont(new java.awt.Font("Tahoma", 0, 14)); // NOI18N  jLabel7.setText("Tanggal lahir :");  jPanel1.add(jLabel7, new org.netbeans.lib.awtextra.AbsoluteConstraints(20, 450, -1, -1));  txtAlamat.setColumns(20);  txtAlamat.setRows(5);  jScrollPane1.setViewportView(txtAlamat);  jPanel1.add(jScrollPane1, new org.netbeans.lib.awtextra.AbsoluteConstraints(140, 250, 180, -1));  dateTanggal.setMaxSelectableDate(new java.util.Date(253370743317000L));  dateTanggal.setMinSelectableDate(new java.util.Date(-62135791083000L));  jPanel1.add(dateTanggal, new org.netbeans.lib.awtextra.AbsoluteConstraints(140, 450, 179, -1));  btnSimpan.setText("Simpan");  btnSimpan.addActionListener(new java.awt.event.ActionListener() {  public void actionPerformed(java.awt.event.ActionEvent evt) {  btnSimpanActionPerformed(evt);  }  });  jPanel1.add(btnSimpan, new org.netbeans.lib.awtextra.AbsoluteConstraints(40, 490, -1, -1));  btnUbah.setText("Ubah");  btnUbah.addActionListener(new java.awt.event.ActionListener() {  public void actionPerformed(java.awt.event.ActionEvent evt) {  btnUbahActionPerformed(evt);  }  });  jPanel1.add(btnUbah, new org.netbeans.lib.awtextra.AbsoluteConstraints(120, 490, -1, -1));  btnHapus.setText("Hapus");  btnHapus.addActionListener(new java.awt.event.ActionListener() {  public void actionPerformed(java.awt.event.ActionEvent evt) {  btnHapusActionPerformed(evt);  }  });  jPanel1.add(btnHapus, new org.netbeans.lib.awtextra.AbsoluteConstraints(190, 490, -1, -1));  btnReset.setText("Reset");  btnReset.addActionListener(new java.awt.event.ActionListener() {  public void actionPerformed(java.awt.event.ActionEvent evt) {  btnResetActionPerformed(evt);  }  });  jPanel1.add(btnReset, new org.netbeans.lib.awtextra.AbsoluteConstraints(260, 490, -1, -1));  tabelData.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"  }  ));  tabelData.addMouseListener(new java.awt.event.MouseAdapter() {  public void mouseClicked(java.awt.event.MouseEvent evt) {  tabelDataMouseClicked(evt);  }  });  jScrollPane2.setViewportView(tabelData);  jPanel1.add(jScrollPane2, new org.netbeans.lib.awtextra.AbsoluteConstraints(350, 170, 620, 350));  jLabel8.setFont(new java.awt.Font("Tahoma", 0, 14)); // NOI18N  jLabel8.setText("Cari Kode / Nama Paspor");  jPanel1.add(jLabel8, new org.netbeans.lib.awtextra.AbsoluteConstraints(560, 130, -1, 20));  txtCariNama.addActionListener(new java.awt.event.ActionListener() {  public void actionPerformed(java.awt.event.ActionEvent evt) {  txtCariNamaActionPerformed(evt);  }  });  jPanel1.add(txtCariNama, new org.netbeans.lib.awtextra.AbsoluteConstraints(739, 130, 160, -1));  btnCari.setText("Cari");  btnCari.addActionListener(new java.awt.event.ActionListener() {  public void actionPerformed(java.awt.event.ActionEvent evt) {  btnCariActionPerformed(evt);  }  });  jPanel1.add(btnCari, new org.netbeans.lib.awtextra.AbsoluteConstraints(920, 130, -1, -1));  comboJenis.setModel(new javax.swing.DefaultComboBoxModel<>(new String[] { "Pria", "Wanita" }));  comboJenis.addActionListener(new java.awt.event.ActionListener() {  public void actionPerformed(java.awt.event.ActionEvent evt) {  comboJenisActionPerformed(evt);  }  });  jPanel1.add(comboJenis, new org.netbeans.lib.awtextra.AbsoluteConstraints(140, 410, 179, -1));  jLabel9.setIcon(new javax.swing.ImageIcon("C:\\Users\\RIVANKA\\Documents\\NetBeansProjects\\Imigrasi\\img\\logo.jpg")); // NOI18N  jLabel9.setMaximumSize(new java.awt.Dimension(128, 128));  jLabel9.setMinimumSize(new java.awt.Dimension(128, 128));  jLabel9.setPreferredSize(new java.awt.Dimension(200, 200));  jPanel1.add(jLabel9, new org.netbeans.lib.awtextra.AbsoluteConstraints(310, 20, 80, 70));  javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());  getContentPane().setLayout(layout);  layout.setHorizontalGroup(  layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)  .addGroup(javax.swing.GroupLayout.Alignment.TRAILING, layout.createSequentialGroup()  .addGap(0, 0, 0)  .addComponent(jPanel1, javax.swing.GroupLayout.DEFAULT\_SIZE, 987, Short.MAX\_VALUE))  );  layout.setVerticalGroup(  layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)  .addGroup(javax.swing.GroupLayout.Alignment.TRAILING, layout.createSequentialGroup()  .addGap(0, 0, Short.MAX\_VALUE)  .addComponent(jPanel1, javax.swing.GroupLayout.PREFERRED\_SIZE, 581, javax.swing.GroupLayout.PREFERRED\_SIZE))  );  pack();  }// </editor-fold>  private void comboJenisActionPerformed(java.awt.event.ActionEvent evt) {  // TODO add your handling code here:  }  private void btnSimpanActionPerformed(java.awt.event.ActionEvent evt) {  ctImg.insert(); // perintah jika saat meklik button simpan  ctImg.isiTable();// setelah insert otomatis tabel akan terupdate  ctImg.reset(); // perintah untuk mengkosongkan kolom input setelah penyimpanan data  }  private void btnUbahActionPerformed(java.awt.event.ActionEvent evt) {  ctImg.update(); // perintah jika saat meklik button ubah  ctImg.isiTable();// setelah update otomatis tabel akan terupdate  ctImg.reset(); // perintah untuk mengkosongkan kolom input setelah update data  }  private void tabelDataMouseClicked(java.awt.event.MouseEvent evt) {  int row = tabelData.getSelectedRow();  try {  ctImg.isiField(row);  } catch (ParseException ex) {  Logger.getLogger(Menu.class.getName()).log(Level.SEVERE, null, ex);  }  }  private void btnResetActionPerformed(java.awt.event.ActionEvent evt) {  ctImg.reset(); // perintah untuk mengkosongkan kolom input setelah penyimpanan data  }  private void btnHapusActionPerformed(java.awt.event.ActionEvent evt) {  ctImg.delete(); // perintah jika saat meklik button ubah  ctImg.isiTable();// setelah update otomatis tabel akan terupdate  ctImg.reset(); // perintah untuk mengkosongkan kolom input setelah update data  }  private void btnCariActionPerformed(java.awt.event.ActionEvent evt) {  ctImg.cari();  ctImg.reset();  }  private void txtCariNamaActionPerformed(java.awt.event.ActionEvent evt) {  // TODO add your handling code here:  }  /\*\*  \* @param args the command line arguments  \*/  public static void main(String args[]) {  /\* Set the Nimbus look and feel \*/  //<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">  /\* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and feel.  \* For details see http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html  \*/  try {  for (javax.swing.UIManager.LookAndFeelInfo info : javax.swing.UIManager.getInstalledLookAndFeels()) {  if ("Nimbus".equals(info.getName())) {  javax.swing.UIManager.setLookAndFeel(info.getClassName());  break;  }  }  } catch (ClassNotFoundException ex) {  java.util.logging.Logger.getLogger(Menu.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);  } catch (InstantiationException ex) {  java.util.logging.Logger.getLogger(Menu.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);  } catch (IllegalAccessException ex) {  java.util.logging.Logger.getLogger(Menu.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);  } catch (javax.swing.UnsupportedLookAndFeelException ex) {  java.util.logging.Logger.getLogger(Menu.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);  }  //</editor-fold>  /\* Create and display the form \*/  java.awt.EventQueue.invokeLater(new Runnable() {  @Override  public void run() {  new Menu().setVisible(true);  }  });  }  // Variables declaration - do not modify  private javax.swing.JButton btnCari;  private javax.swing.JButton btnHapus;  private javax.swing.JButton btnReset;  private javax.swing.JButton btnSimpan;  private javax.swing.JButton btnUbah;  private javax.swing.JComboBox<String> comboJenis;  private com.toedter.calendar.JDateChooser dateTanggal;  private javax.swing.JLabel jLabel1;  private javax.swing.JLabel jLabel2;  private javax.swing.JLabel jLabel3;  private javax.swing.JLabel jLabel4;  private javax.swing.JLabel jLabel5;  private javax.swing.JLabel jLabel6;  private javax.swing.JLabel jLabel7;  private javax.swing.JLabel jLabel8;  private javax.swing.JLabel jLabel9;  private javax.swing.JPanel jPanel1;  private javax.swing.JScrollPane jScrollPane1;  private javax.swing.JScrollPane jScrollPane2;  private javax.swing.JTable tabelData;  private javax.swing.JTextArea txtAlamat;  private javax.swing.JTextField txtCariNama;  private javax.swing.JTextField txtKode;  private javax.swing.JTextField txtNama;  private javax.swing.JTextField txtWarga;  // End of variables declaration  ControllerImigran ctImg;  public JTable getTableData() { //mengambil JTable dari java swing  return tabelData;  } // alasannya karena tabel pada gui Private      //berfungsi untuk mengambil data dari field kode karena bersifat private  public JTextField gettxtKode(){  return txtKode;  }    //berfungsi untuk mengambil data dari field Nama karena bersifat private  public JTextField gettxtNama(){  return txtNama;  }    //berfungsi untuk mengambil data dari Text Area alamat karena bersifat private  public JTextArea gettxtAlamat(){  return txtAlamat;  }    //berfungsi untuk mengambil data dari field Warga negara karena bersifat private  public JTextField gettxtWarga(){  return txtWarga;  }    //berfungsi untuk mengambil data dari Combo box jenis kelamin karena bersifat private  public JComboBox getcomboJenis(){  return comboJenis;  }    //berfungsi untuk mengambil data dari dateChooser tanggal lahir karena bersifat private  public JDateChooser getdateTanggal(){  return dateTanggal;  }  //berfungsi untuk mengambil data dari search karena bersifat private  public JTextField gettxtCari(){  return txtCariNama;  }  } |

***Listing* Program 2.7** Menu.Java

## **2.3.2 Penjelasan Program**

1. ***Database***

**Gambar 2.3.1** Rancangan *Database*

Pada program ini terdapat empat entitas yaitu *admin*  dan Imigran. Entitas admin digunakan untuk data login dari seorang adminstrator. Sedangkan entitas imigran digunakan untuk menyimpan data-data dari Imigran

1. **Login**

Pada login akan diminta untuk menginputkan username dan password. Kemudian data yang diinputkan akan diambil oleh controller seperti Setelah itu data yang diinputkan akan diperiksa pada login model. Jika data ada, akan dilanjutkan ke halaman menu . Dan jika data tidak ada, maka akan menampilkan pesan eror atau username dan password salah.

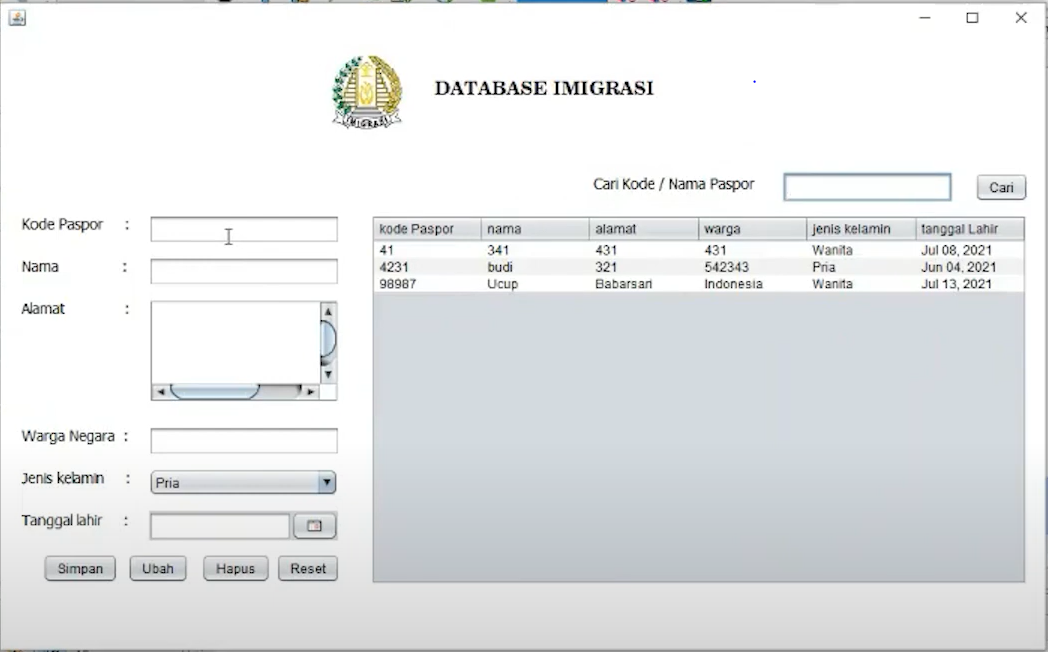


Gambar 2.3.2 Tampilan Login

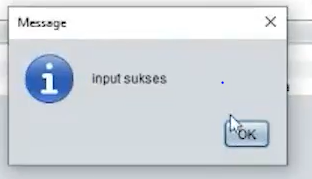
Gambar 2.3.3 Tampilan Login jika username dan password salah

1. **Input *Imigran***

Pada menu ini pengguna akan diminta untuk menginputkan data – data seperti kode paspor nama imigran alamat imigran,kewarganegaraan, jenis kelamin, dan tanggal lahir. setelah diinputkan dengan mengeklik tombol simpan maka data akan tersimpan di tabel data imigran seperti di **Gambar 2.3.5**

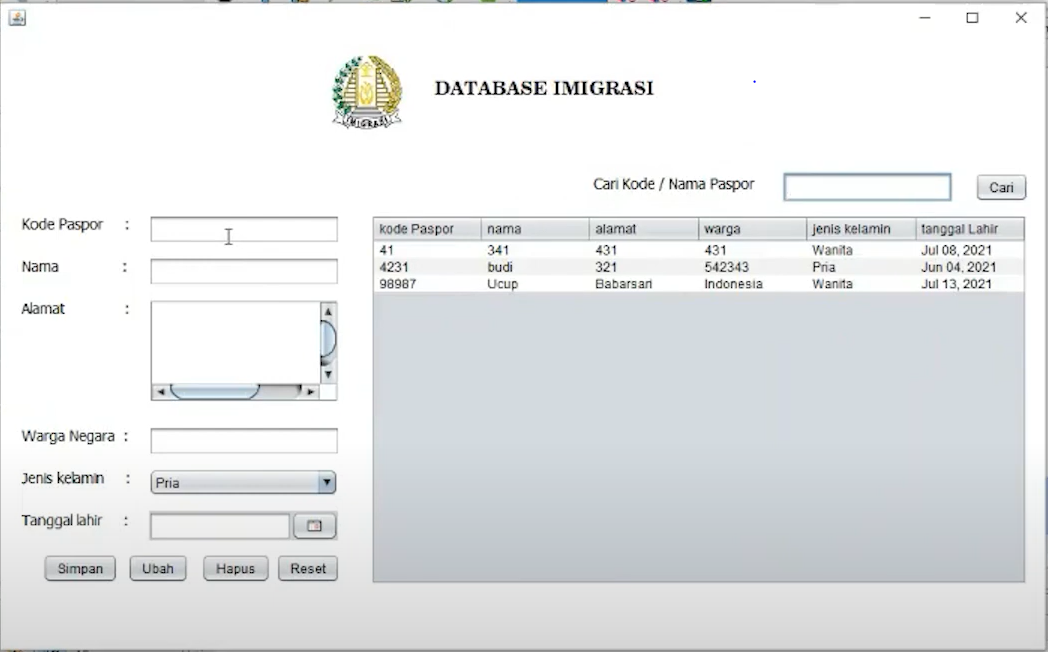


**Gambar 2.3.4** Tampilan Input Data Review



1. **Data Imigran**

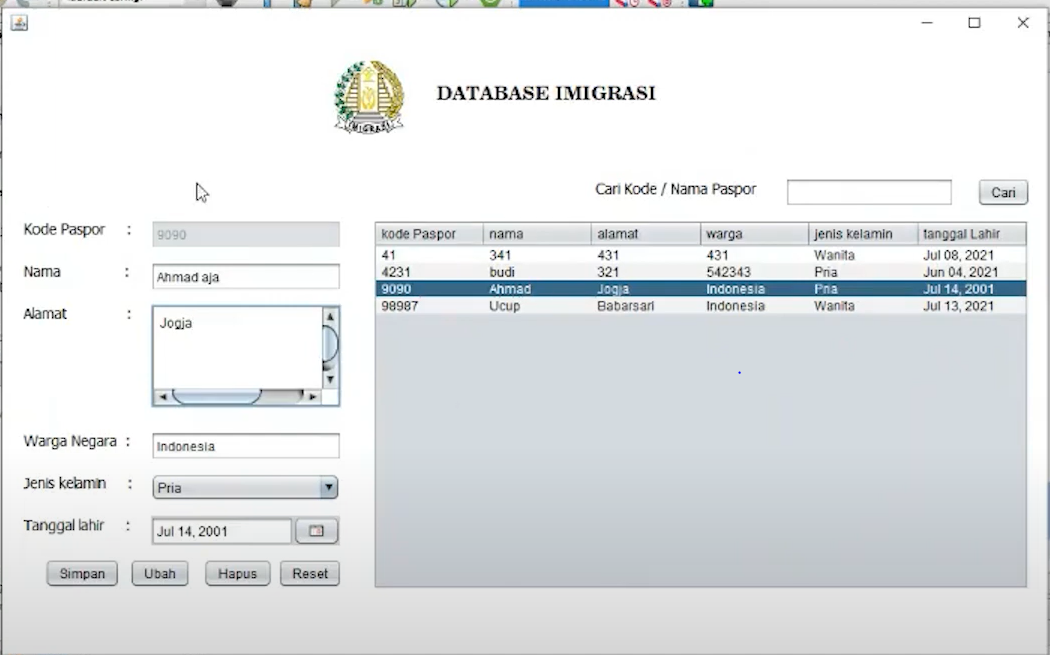
Pada bagian ini akan ditampilkan table yang berisi data – data Imigran yang sudah pernah diinput. Data-data yang akan ditampilkan adalah kode paspor imigran, nama imgiran, alamat, warga negara, jenis kelamin dan juga tanggal lahir. lalu jika ingin mengubah dan menghapus data tinggal mengklik salah satu data yang akan diubah. Maka data tersebut akan keluar diinputan yang ada disebelah kiri.



**Gambar 2.3.5** Tampilan Table data imigran

1. **Ubah Data Imigran**

jika ingin mengubah dan menghapus data tinggal mengklik salah satu data yang akan diubah. Maka data tersebut akan keluar diinputan yang ada disebelah kiri. Dengan begitu kita bisa langsung mengubah data dinputan tersebut dengan mengklik tombol ubah data maka data akan langsung berubah dan tampil kembali di tabel data. Data yang tidak bisa diubah yaitu data no passport karena data no passport merupakan primarykey yang ada didatabase dan menjadi identitas utama yang ada didalam program.



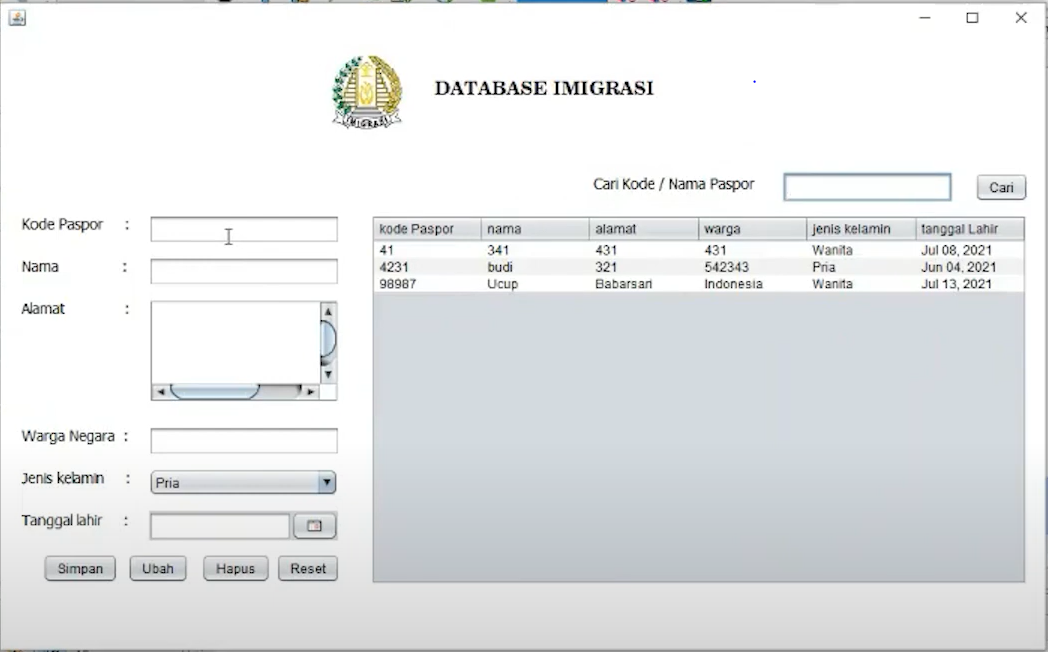
**Gambar 2.5** Tampilan Ubah data

Update digunakan untuk memperbarui nilai review yang pengguna inputkan sebelumnya. Pertama program akan mengambil nilai dari input user. Kemudian program akan menghitung rata – rata baru sesuai data yang diinputkan kemudian mengambil nilai dari table movie dan keseluruhan dari table review seperti pada listing program 2.7. Selanjutnya program akan mencari nilai movie sebelum pengguna menginputkan review dari film yang dipilih seperti listing program 2.8. Setelah mendapatkan nilai baru program akan meng-*update* data yang ada.

Pada Delete, proses yang digunakan hamper sama, namun program hanya melakukan perhitungan untuk mendapat nilai movie sebelum pengguna menginputkan review dari film yang dipilih seperti listing program 2.8. Setelah mendapatkan nilai tersebut, data review pengguna pada table review akan dihapus, kemudian data nilai pada table movie akan di-*update* menggunakan nilai yang didapatkan pada listing program 2.8. Lalu kontribusi dari pengguna akan dikurangi satu.

**2.3.3 Output Program**

**Gambar 2.7** Tampilan halaman login yang akan pertama kali tampil setelah di run



**Gambar 2.8** Jika berhasil login maka akan muncul dialog

# **BAB III**

**JADWAL PENGERJAAN DAN PEMBAGIAN TUGAS**

## **3.1 Jadwal Pengerjaan**

Jadwal pengerjaan proyek Praktikum Pemrograman Berorientasi Objek *Pendataan Imigrasi* ini sebagai berikut:

**Tabel 3.1** Tabel Jadwal Pengerjaan

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| no | Nama Tugas | Minggu ke- | | | |
| Mei | | Juni | |
| 3 | 4 | 1 | 2 |
|  | Menentukan Judul Proyek |  |  |  |  |
|  | Perancangan Konsep Proyek |  |  |  |  |
|  | Halaman Login |  |  |  |  |
|  | Halaman Input Data |  |  |  |  |
|  | Data imigran |  |  |  |  |
|  | Ubah data imigran |  |  |  |  |
|  | Hapus Data Imigran |  |  |  |  |
|  | Cari data Imigran |  |  |  |  |
|  | Pembuatan Laporan |  |  |  |  |

## **3.2 Pembagian Tugas**

Pembagian tugas proyek Praktikum Pemrograman Berorientasi Objek *Pendataan Imigrasi* ini sebagai berikut:

**Tabel 3.2** Tabel Pembagian Tugas

|  |  |  |
| --- | --- | --- |
| no | aktivitas | penanggung jawab |
| 1 | Menentukan Judul Proyek | Isnaini & Rivanka |
| 2 | Perancangan Konsep Proyek | Isnaini & Rivanka |
| 3 | Halaman Login | Isnaini |
| 4 | Halaman Input Data | Isnaini |
| 5 | Data imigran | Rivanka |
| 6 | Ubah data imigran | Rivanka |
| 7 | Hapus Data Imigran | Rivanka |
| 8 | Cari data Imigran | Rivanka |
| 9 | Pembuatan Laporan | Isnaini |

# **BAB IV**

**KESIMPULAN DAN SARAN**

## **4.1 Kesimpulan**

Kesimpulan yang dapat diambil dari pengerjaan ini adalah proyek *Pendataan Imigrasi* yang telah dibuat dapat melakukan penginputan Data imigran lebih cepat dan bisa diubah dan lebih dinamis sehingga aperator tidak perlu khawatir kehilangan daata. Selain itu program dapat digunakan untuk meng-*update* data ulasan jika sewaktu-waktu memerlukan perubahan.

## **4.2 Saran**

Adapun saran yang dapat dipertimbangkan untuk penyempurnaan dan pengembangan program *Pendataan Imigrasi* lebih lanjut adalah sebagai berikut:

1. Sistem *Pendataan Imigrasi* diharapkan dapat bekerja dengan lebih optimal dengan menyediakan fitur-fitur maupun menu-menu yang lebih banyak serta lebih kompleks, sehingga pengguna dapat lebih nyaman dalam melakukan aktivitas.
2. Tampilan pada yang diharapkan lebih baik dan kedepannya bisa diperbarui.
3. Penambahan Export data menjadi Excel atau doc

# **DAFTAR PUSTAKA**

Anonim.2019.*Modul Praktikum Pemrograman Berorientasi Objek Program Studi Teknik Infromatika Jurusan Teknik Informatika Fakultas Tekik Industri UPN “Veteran” Yogyakarta*. Yogyakarta: UPN “Veteran” Yogyakarta.

