

ACTIVITY 3

Nama : Alif Fathur Rahman

NPM : 51421577

Kelas : 4IA28

CODE

MahasiswaController

```
/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this
 * license
 * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this template
 */
package com.mahasiswa.controller;

import com.mahasiswa.model.MahasiswaDAO;
import com.mahasiswa.model.ModelMahasiswa;
import java.util.List;
/**
 *
 * @author alif
 */
public class MahasiswaController {
    private MahasiswaDAO mahasiswaDAO;

    public MahasiswaController(MahasiswaDAO mahasiswaDAO){
        this.mahasiswaDAO = mahasiswaDAO;
    }

    public void displayMahasiswaList(List<ModelMahasiswa> mahasiswaList){
        if(mahasiswaList.isEmpty()){
            System.out.println("Tidak ada data mahasiswa");
        } else {
            System.out.println("");
            System.out.println("=====");
            for(ModelMahasiswa m: mahasiswaList){
                System.out.println("ID      : " + m.getId());
                System.out.println("NPM      : " + m.getNpm());
                System.out.println("NAMA      : " + m.getNama());
                System.out.println("SEMESTER  : " + m.getSemester());
                System.out.println("IPK       : " + m.getIpk());
                System.out.println("=====");
            }
        }
    }
}
```

```

    }
}

public void displayMessage(String message){
    System.out.println(message);
}

public void checkDatabaseConnection(){
    boolean isConnected = mahasiswaDAO.checkConnection();
    if (isConnected){
        displayMessage("Koneksi ke db berhasil");
    } else{
        displayMessage("Koneksi DB Gagal");
    }
}

// READ ALL (Menampilkan semua mahasiswa)
public void displayAllMahasiswa(){
    List<ModelMahasiswa> mahasiswaList = mahasiswaDAO.getAllMahasiswa();
    displayMahasiswaList(mahasiswaList);
}

public void addMahasiswa(String npm, String nama, int semester, float ipk){
    ModelMahasiswa mahasiswaBaru = new ModelMahasiswa(0, npm, nama, semester, ipk);
    System.out.println("Controller Data:  " + npm + nama + semester + ipk);
    System.out.println(mahasiswaBaru);
    mahasiswaDAO.addMahasiswa(mahasiswaBaru);
    displayMessage("Mahasiswa berhasil ditambahkan!");
}

public void updateMahasiswa(int id, String npm, String nama, int semester, float ipk){
    ModelMahasiswa mahasiswaBaru = new ModelMahasiswa(id, npm, nama, semester,
    ipk);
    mahasiswaDAO.updateMahasiswa(mahasiswaBaru);
    displayMessage("Mahasiswa berhasil diperbarui!");
}

public void deleteMahasiswa(int id){
    mahasiswaDAO.deleteMahasiswa(id);
    displayMessage("Mahasiswa Berhasil Dihapus!");
}

```

```
public void closeConnection() {  
    mahasiswaDAO.closeConnection();  
}  
}
```

ModelMahasiswa.java

```
/*  
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this  
 license  
 * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this template  
 */  
package com.mahasiswa.model;  
  
/**  
 *  
 * @author alif  
 */  
public class ModelMahasiswa {  
    private int id;  
    private String npm;  
    private String nama;  
    private int semester;  
    private float ipk;  
  
    public ModelMahasiswa(int id, String npm, String nama, int semester, float ipk){  
        this.id = id;  
        this.npm = npm;  
        this.nama = nama;  
        this.semester = semester;  
        this.ipk = ipk;  
    }  
  
    public int getId() {  
        return id;  
    }  
  
    public void setId(int id) {  
        this.id = id;  
    }  
  
    public String getNpm() {  
        return npm;  
    }  
  
    public void setNpm(String npm) {
```

```

        this.npm = npm;
    }

    public String getNama() {
        return nama;
    }

    public void setNama(String nama) {
        this.nama = nama;
    }

    public int getSemester() {
        return semester;
    }

    public void setSemester(int semester) {
        this.semester = semester;
    }

    public float getIpk() {
        return ipk;
    }

    public void setIpk(float ipk) {
        this.ipk = ipk;
    }
}

```

MahasiswaDAO

```

/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this
 license
 * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this template
 */
package com.mahasiswa.model;

import java.sql.*;
import java.util.ArrayList;
import java.util.List;
/**
 *
 * @author alif
 */

```

```

public class MahasiswaDAO {
    private Connection connection;

    public MahasiswaDAO (){
        try {
            Class.forName("com.mysql.cj.jdbc.Driver");
            connection =
DriverManager.getConnection("jdbc:mysql://localhost:3306/kukuh_mvc", "root", "");
        } catch (Exception e) {
            e.printStackTrace();
        }
    }

    public boolean checkConnection(){
        try {
            if(connection != null && !connection.isClosed()){
                return true;
            }
        } catch(SQLException e){
            e.printStackTrace();
        }
        return false;
    }

    public void addMahasiswa(ModelMahasiswa mahasiswa){
        String sql = "INSERT INTO mahasiswa (npm, nama, semester, ipk) VALUES (?, ?, ?,
?)"
);
        try{
            PreparedStatement pstmt = connection.prepareStatement(sql);
            pstmt.setString(1, mahasiswa.getNpm());
            pstmt.setString(2, mahasiswa.getNama());
            pstmt.setInt(3, mahasiswa.getSemester());
            pstmt.setFloat(4, mahasiswa.getIpk());
            pstmt.executeUpdate();
        } catch(SQLException e){
            e.printStackTrace();
        }
    }

    public List<ModelMahasiswa> getAllMahasiswa(){
        List<ModelMahasiswa> mahasiswaList = new ArrayList<>();
        String sql = "SELECT * FROM mahasiswa";
        try{
            Statement stmt = connection.createStatement();
            ResultSet rs = stmt.executeQuery(sql);
            while(rs.next()){
                mahasiswaList.add(new ModelMahasiswa(
                    rs.getInt("id"),

```

```

        rs.getString("npm"),
        rs.getString("nama"),
        rs.getInt("semester"),
        rs.getFloat("ipk")
    ));
    }
    } catch(SQLException e){
        e.printStackTrace();
    }
    return mahasiswaList;
}

public void updateMahasiswa(ModelMahasiswa mahasiswa){
    String sql = "UPDATE mahasiswa SET npm = ?, nama = ?, semester = ?, ipk = ?
WHERE id = ?";
    try{
        PreparedStatement pstmt = connection.prepareStatement(sql);
        pstmt.setString(1, mahasiswa.getNpm());
        pstmt.setString(2, mahasiswa.getNama());
        pstmt.setInt(3, mahasiswa.getSemester());
        pstmt.setFloat(4, mahasiswa.getIpk());
        pstmt.setInt(5, mahasiswa.getId());
        pstmt.executeUpdate();
    } catch(SQLException e){
        e.printStackTrace();
    }
}

public void deleteMahasiswa (int id){
    String sql = "DELETE from mahasiswa where id = ?";
    try {
        PreparedStatement pstmt = connection.prepareStatement(sql);
        pstmt.setInt(1, id);
        pstmt.executeUpdate();
    } catch (SQLException e) {
        e.printStackTrace();
    }
}

public void closeConnection(){
    try {
        if (connection != null){
            connection.close();
        }
    } catch (SQLException e) {
        e.printStackTrace();
    }
}

```

```
}  
}  
}
```

MahasiswaView

```
/*  
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this  
 license  
 * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this template  
 */  
package com.mahasiswa.view;  
  
import com.mahasiswa.controller.MahasiswaController;  
import com.mahasiswa.model.MahasiswaDAO;  
import java.util.Scanner;  
/**  
 *  
 * @author alif  
 */  
  
public class MahasiswaView {  
    public static void main(String[] args){  
        MahasiswaDAO mahasiswaDAO = new MahasiswaDAO();  
        MahasiswaController mahasiswaController = new  
MahasiswaController(mahasiswaDAO);  
  
        Scanner scanner = new Scanner(System.in);  
        int pilihan;  
  
        while(true){  
            System.out.println("Menu:");  
            System.out.println("1. Tampilkan Semua Mahasiswa");  
            System.out.println("2. Tambah Mahasiswa");  
            System.out.println("3. Update Mahasiswa");  
            System.out.println("4. Hapus Mahasiswa");  
            System.out.println("5. Cek Koneksi Database");  
            System.out.println("6. Keluar");  
            System.out.print("PILIH OPSI: ");  
            pilihan = scanner.nextInt();  
            scanner.nextLine();  
  
            switch (pilihan){  
                case 1:  
                    mahasiswaController.displayAllMahasiswa();  
                    break;
```

case 2:

```
// tambah mhs
System.out.println("Masukkan NPM: ");
String npm = scanner.next();
System.out.println("Masukkan Nama: ");
String nama = scanner.next();
System.out.println("Masukkan Semester: ");
int semester = scanner.nextInt();
System.out.println("Masukkan IPK: ");
float ipk = scanner.nextFloat();
System.out.println(npm + nama + semester + ipk);

mahasiswaController.addMahasiswa(npm, nama, semester, ipk);
break;
```

case 3:

```
System.out.print("Masukkan ID mahasiswa: ");
int id = scanner.nextInt();
scanner.nextLine();

System.out.println("Masukkan NPM: ");
String npmBaru = scanner.next();
System.out.println("Masukkan Nama: ");
String namaBaru = scanner.next();
System.out.println("Masukkan Semester: ");
int semesterBaru = scanner.nextInt();
System.out.println("Masukkan IPK: ");
float ipkBaru = scanner.nextFloat();

mahasiswaController.updateMahasiswa(id, npmBaru, namaBaru, semesterBaru,
ipkBaru);
break;
```

case 4:

```
System.out.print("Masukkan ID Mahasiswa: ");
int idHapus = scanner.nextInt();
mahasiswaController.deleteMahasiswa(idHapus);
```

case 5:

```
mahasiswaController.checkDatabaseConnection();
break;
```

case 6:

```
// Keluar
mahasiswaController.closeConnection();
System.out.println("Program selesai.");
return;
```

default:


```

        System.out.println("Input Tidak valid");
    }
}
}
}

```

OUTPUT

```

Output x
rahma - C:\Users\rahma x Run (MahasiswaView) x
Menu:
1. Tampilkan Semua Mahasiswa
2. Tambah Mahasiswa
3. Update Mahasiswa
4. Hapus Mahasiswa
5. Cek Koneksi Database
6. Keluar
PILIH OPSI: 2
Masukkan NPM:
51421577
Masukkan Nama:
alif
Masukkan Semester:
7
Masukkan IPK:
3,9
51421577alif73.9
Controller Data: 51421577alif73.9
com.mahasiswa.model.ModelMahasiswa@1990a65e
Mahasiswa berhasil ditambahkan!
Menu:
1. Tampilkan Semua Mahasiswa
2. Tambah Mahasiswa
3. Update Mahasiswa
4. Hapus Mahasiswa
5. Cek Koneksi Database
6. Keluar
PILIH OPSI: 2
Masukkan NPM:
5144788
Masukkan Nama:
fathur
Masukkan Semester:

```

```

rahma - C:\Users\rahma x Run (MahasiswaView) x
Masukkan IPK:
3,8
5144788fathur83.8
Controller Data: 5144788fathur83.8
com.mahasiswa.model.ModelMahasiswa@68ceda24
Mahasiswa berhasil ditambahkan!
Menu:
1. Tampilkan Semua Mahasiswa
2. Tambah Mahasiswa
3. Update Mahasiswa
4. Hapus Mahasiswa
5. Cek Koneksi Database
6. Keluar
PILIH OPSI: 2
Masukkan NPM:
6789077
Masukkan Nama:
rahman
Masukkan Semester:
7
Masukkan IPK:
3,9
6789077rahman73.9
Controller Data: 6789077rahman73.9
com.mahasiswa.model.ModelMahasiswa@281e3708
Mahasiswa berhasil ditambahkan!
Menu:
1. Tampilkan Semua Mahasiswa
2. Tambah Mahasiswa
3. Update Mahasiswa
4. Hapus Mahasiswa
5. Cek Koneksi Database
6. Keluar

```

```
rahma - C:\Users\rahma x Run (MahasiswaView) x
PILIH OPSI: 1

=====
ID      : 2
NPM     : 51421577
NAMA    : alif
SEMESTER : 7
IPK     : 3.9
=====
ID      : 3
NPM     : 51421599
NAMA    : fathur
SEMESTER : 8
IPK     : 3.8
=====
ID      : 4
NPM     : 51421577
NAMA    : alif
SEMESTER : 7
IPK     : 3.9
=====
ID      : 5
NPM     : 5144788
NAMA    : fathur
SEMESTER : 8
IPK     : 3.8
=====
ID      : 6
NPM     : 6789077
NAMA    : rahman
SEMESTER : 7
IPK     : 3.9
=====
```

Koneksi database

```
Output - Run (MahasiswaView) x
--- compiler:3.13.0:compile (default-compile) @ mkukuh-mvc ---
Nothing to compile - all classes are up to date.

--- exec:3.1.0:exec (default-cli) @ mkukuh-mvc ---
Menu:
1. Tampilkan Semua Mahasiswa
2. Tambah Mahasiswa
3. Update Mahasiswa
4. Hapus Mahasiswa
5. Cek Koneksi Database
6. Keluar
PILIH OPSI: 5
Koneksi ke db berhasil
```

ACTIVITY 4

Nama : Alif Fathur Rahman

NPM : 51421577

Kelas : 4IA28

MahasiswaController.java

```
/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this
 license
 * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Interface.java to edit this template
 */
package com.mahasiswa.controller;

import com.mahasiswa.model.ModelMahasiswa;
import java.util.List;
/**
 *
 * @author alif
 */
public interface MahasiswaController {
    public void addMhs(ModelMahasiswa mhs);
    public ModelMahasiswa getMhs(int id);
    public void updateMhs(ModelMahasiswa mhs);
    public void deleteMhs(int id);
    public List<ModelMahasiswa> getAllMahasiswa();
}
```

MahasiswaControllerImpl.java

```
/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this
 license
 * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this template
 */
package com.mahasiswa.controller;

import com.mahasiswa.model.HibernateUtil;
import com.mahasiswa.model.ModelMahasiswa;
import java.util.List;
import org.hibernate.Session;
import org.hibernate.Transaction;
import org.hibernate.query.Query;
```

```

/**
 *
 * @author alif
 */
public class MahasiswaControllerImpl implements MahasiswaController {

    @Override
    public void addMhs(ModelMahasiswa mhs){
        Transaction trx = null;

        try (Session session = HibernateUtil.getSessionFactory().openSession()){
            trx = session.beginTransaction();
            session.save(mhs);
            trx.commit();
        } catch (Exception e){
            if (trx != null){
                trx.rollback();
            }
            e.printStackTrace();
        }
    }

    @Override
    public ModelMahasiswa getMhs(int id) {
        throw new UnsupportedOperationException("Not supported yet."); // Generated from
nbfs://nbhost/SystemFileSystem/Templates/Classes/Code/GeneratedMethodBody
    }

    @Override
    public void updateMhs(ModelMahasiswa mhs) {
        Transaction trx = null;

        try (Session session = HibernateUtil.getSessionFactory().openSession()){
            trx = session.beginTransaction();
            session.update(mhs);
            trx.commit();
        } catch (Exception e){
            if (trx != null){
                trx.rollback();
            }
            e.printStackTrace();
        }
    }

    @Override

```

```

public void deleteMhs(int id) {
    Transaction trx = null;

    try (Session session = HibernateUtil.getSessionFactory().openSession()){
        trx = session.beginTransaction();
        ModelMahasiswa mhs = session.get(ModelMahasiswa.class, id);
        if(mhs != null){
            session.delete(mhs);
            System.out.println("Berhasil hapus");
        }
        trx.commit();
    } catch (Exception e){
        if (trx != null){
            trx.rollback();
        }
        e.printStackTrace();
    }
}

@Override
public List<ModelMahasiswa> getAllMahasiswa() {
    Transaction trx = null;
    List<ModelMahasiswa> listMhs = null;

    try (Session session = HibernateUtil.getSessionFactory().openSession()){
        trx = session.beginTransaction();
        // Using HQL (Hibernate Query Language) to fetch all records
        Query<ModelMahasiswa> query = session.createQuery("from ModelMahasiswa",
ModelMahasiswa.class);
        listMhs = query.list(); // Fetch all results

        trx.commit(); // Commit transaction
    } catch (Exception e) {
        if (trx != null) {
            trx.rollback(); // Rollback transaction in case of error
        }
        e.printStackTrace();
    }

    // Return the fetched list
    return listMhs;
}
}

```

HibernateUtil.java

```
/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this
 * license
 * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this template
 */
package com.mahasiswa.model;

import org.hibernate.Session;
import org.hibernate.SessionFactory;
import org.hibernate.cfg.Configuration;

/**
 *
 * @author alif
 */
public class HibernateUtil {
    private static SessionFactory sessionFactory;

    static {
        try {
            // Create the SessionFactory from hibernate.cfg.xml
            sessionFactory = new Configuration().configure().buildSessionFactory();
        } catch (Throwable ex) {
            // Make sure you log the exception, as it might be swallowed
            System.err.println("Initial SessionFactory creation failed." + ex);
            throw new ExceptionInInitializerError(ex);
        }
    }

    public static SessionFactory getSessionFactory() {
        return sessionFactory;
    }

    public static void testConnection() {
        try (Session session = sessionFactory.openSession()) {
            System.out.println("Connection to the database was successful!");
        } catch (Exception e) {
            System.err.println("Failed to connect to the database.");
            e.printStackTrace();
        }
    }
}
```

ModelMahasiswa.java

```
/*
```

* Click <nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt> to change this license

* Click <nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java> to edit this template

*/

```
package com.mahasiswa.model;
```

```
import jakarta.persistence.*;
```

```
/**
```

```
*
```

```
* @author alif
```

```
*/
```

```
@Entity
```

```
@Table(name = "mahasiswa")
```

```
public class ModelMahasiswa {
```

```
    @Id
```

```
    @GeneratedValue(strategy = GenerationType.IDENTITY)
```

```
    public int getId() {
```

```
        return id;
```

```
    }
```

```
    public void setId(int id) {
```

```
        this.id = id;
```

```
    }
```

```
    public int getSemester() {
```

```
        return semester;
```

```
    }
```

```
    public void setSemester(int semester) {
```

```
        this.semester = semester;
```

```
    }
```

```
    public String getNpm() {
```

```
        return npm;
```

```
    }
```

```
    public void setNpm(String npm) {
```

```
        this.npm = npm;
```

```
    }
```

```
    public String getNama() {
```

```
        return nama;
```

```
    }
```

```

public void setNama(String nama) {
    this.nama = nama;
}

public float getIpk() {
    return ipk;
}

public void setIpk(float ipk) {
    this.ipk = ipk;
}

@Column(name = "semester")
private int id;
private int semester;
@Column(name = "npm", nullable = false, length = 8)
private String npm;
@Column(name = "nama", nullable = false, length = 8)
private String nama;
@Column(name = "ipk")
private float ipk;

public ModelMahasiswa(int _id, String _npm, String _nama, int _semester, float _ipk) {
    this.id = _id;
    this.npm = _npm;
    this.nama = _nama;
    this.semester = _semester;
    this.ipk = _ipk;
}

public ModelMahasiswa() {

}
}

```

ModelTabelMahasiswa.java

```

/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this
 * license
 * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this template
 */
package com.mahasiswa.model;

import java.util.List;
import javax.swing.table.AbstractTableModel;

/**

```



```

*
* @author alif
*/
public class ModelTabelMahasiswa extends AbstractTableModel {
    private List<ModelMahasiswa> mahasiswaList;
    private String[] columnNames = {"ID", "NPM", "Nama", "Semester", "IPK"};

    public ModelTabelMahasiswa(List<ModelMahasiswa> mahasiswaList) {
        this.mahasiswaList = mahasiswaList;
    }

    @Override
    public int getRowCount() {
        return mahasiswaList.size(); // Jumlah baris sesuai dengan jumlah data mahasiswa
    }

    @Override
    public int getColumnCount() {
        return columnNames.length; // Jumlah kolom sesuai dengan jumlah elemen dalam
columnNames
    }

    @Override
    public Object getValueAt(int rowIndex, int columnIndex) {
        ModelMahasiswa mahasiswa = mahasiswaList.get(rowIndex);
        switch (columnIndex) {
            case 0:
                return mahasiswa.getId();
            case 1:
                return mahasiswa.getNpm();
            case 2:
                return mahasiswa.getNama();
            case 3:
                return mahasiswa.getSemester();
            case 4:
                return mahasiswa.getIpk();
            default:
                return null;
        }
    }

    @Override
    public String getColumnName(int column) {
        return columnNames[column]; // Mengatur nama kolom
    }
}

```

```

@Override
public boolean isCellEditable(int rowIndex, int columnIndex) {
    return false; // Semua sel tidak dapat diedit
}

// Method untuk menambahkan atau memodifikasi data, jika dibutuhkan
public void setMahasiswaList(List<ModelMahasiswa> mahasiswaList) {
    this.mahasiswaList = mahasiswaList;
    fireTableDataChanged(); // Memberitahu JTable bahwa data telah berubah
}
}

```

MahasiswaView.java

```

/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this
 * license
 * Click nbfs://nbhost/SystemFileSystem/Templates/GUIForms/JFrame.java to edit this
 * template
 */
package com.mahasiswa.view;

import com.mahasiswa.controller.MahasiswaControllerImpl;
import com.mahasiswa.model.HibernateUtil;
import com.mahasiswa.model.ModelMahasiswa;
import com.mahasiswa.model.ModelTabelMahasiswa;
import java.util.List;
import javax.swing.JLabel;
import javax.swing.JOptionPane;
import javax.swing.JPanel;
import javax.swing.JTextField;

/**
 *
 * @author alif
 */
public class MahasiswaView extends javax.swing.JFrame {

    /**
     * Creates new form MahasiswaView
     */
    private MahasiswaControllerImpl controller;
    public MahasiswaView() {
        initComponents();
        controller = new MahasiswaControllerImpl();
        HibernateUtil.testConnection();
        loadMahasiswaTable();
    }
}

```

```

public void loadMahasiswaTable() {
// Ambil data dari controller
List<ModelMahasiswa> listMahasiswa = controller.getAllMahasiswa();

// Buat model tabel kustom dengan data mahasiswa
ModelTabelMahasiswa tableModel = new ModelTabelMahasiswa(listMahasiswa);

// Set model pada JTable
dataTable.setModel(tableModel);
}
/**
 * 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() {

    jScrollPane1 = new javax.swing.JScrollPane();
    jTable1 = new javax.swing.JTable();
    NamaField = new javax.swing.JTextField();
    NpmField = new javax.swing.JTextField();
    SemesterField = new javax.swing.JTextField();
    IpkField = new javax.swing.JTextField();
    simpanButton = new javax.swing.JButton();
    refreshButton = new javax.swing.JButton();
    buangButton = new javax.swing.JButton();
    jLabel1 = new javax.swing.JLabel();
    jLabel2 = new javax.swing.JLabel();
    jLabel3 = new javax.swing.JLabel();
    jLabel4 = new javax.swing.JLabel();
    jScrollPane2 = new javax.swing.JScrollPane();
    dataTable = new javax.swing.JTable();

    jTable1.setModel(new javax.swing.table.DefaultTableModel(
        new Object [][] {
            { null, null, null, null },
            { null, null, null, null },
            { null, null, null, null },
            { null, null, null, null }
        },
        new String [] {
            "Title 1", "Title 2", "Title 3", "Title 4"
        }
    ));

```

```
jScrollPane1.setViewportViewView(jTable1);

setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);

simpanButton.setText("Save");
simpanButton.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        simpanButtonActionPerformed(evt);
    }
});

refreshButton.setText("Refresh");
refreshButton.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        refreshButtonActionPerformed(evt);
    }
});

buangButton.setText("Buang");
buangButton.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        buangButtonActionPerformed(evt);
    }
});

jLabel1.setText("Nama");

jLabel2.setText("NPM");

jLabel3.setText("Semester");

jLabel4.setText("IPK");

dataTable.setModel(new javax.swing.table.DefaultTableModel(
    new Object [][] {
        {null, null, null, null},
        {null, null, null, null},
        {null, null, null, null},
        {null, null, null, null}
    },
    new String [] {
        "Title 1", "Title 2", "Title 3", "Title 4"
    }
));
jScrollPane2.setViewportViewView(dataTable);
```

```
javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
getContentPane().setLayout(layout);
layout.setHorizontalGroup(
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(layout.createSequentialGroup()
            .add(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                .addComponent>NamaField,
                .addGap(7, 7, 7))
            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)
                .addComponent(simpanButton)
                .addComponent(refreshButton)))
        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addComponent(NpmField)
            .addComponent(SemesterField)
            .addComponent(IpkField))
        .addGap(68, 68, 68)
        .addComponent(jScrollPane2,
            javax.swing.GroupLayout.PREFERRED_SIZE, 500,
            javax.swing.GroupLayout.PREFERRED_SIZE))
        .addGroup(layout.createSequentialGroup()
            .add(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                .addComponent(buangButton))
            .addContainerGap(235, Short.MAX_VALUE))
    );
layout.setVerticalGroup(
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
```

```

        .addGroup(layout.createSequentialGroup())
        .addGap(39, 39, 39)

        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
            .addComponent>NamaField, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
            .addComponent(jLabel1))
        .addGap(18, 18, 18)

        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
            .addComponent(NpmField, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
            .addComponent(jLabel2))
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
            .addComponent>SemesterField, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
            .addComponent(jLabel3))
        .addGap(18, 18, 18)

        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
            .addComponent>IpkField, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
            .addComponent(jLabel4))
        .addGap(32, 32, 32)

        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
            .addComponent(simpanButton)
            .addComponent(refreshButton))
        .addGap(18, 18, 18)
        .addComponent(buangButton)
        .addGap(18, 18, 18)
        .addComponent(jScrollPane2, javax.swing.GroupLayout.PREFERRED_SIZE, 250,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addContainerGap(183, Short.MAX_VALUE))
    );

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

private void simpanButtonActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    String npm = NpmField.getText();
    String nama =>NamaField.getText();
    int semester = Integer.parseInt(SemesterField.getText());

```

```

float ipk = Float.parseFloat(IpkField.getText());
ModelMahasiswa mahasiswa = new ModelMahasiswa(0, npm, nama, semester, ipk);
System.out.println(mahasiswa.getIpk());
System.out.println(mahasiswa.getNama());
System.out.println(mahasiswa.getSemester());
System.out.println(mahasiswa.getNpm());

controller.addMhs(mahasiswa);
loadMahasiswaTable();
}

private void refreshButtonActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    loadMahasiswaTable();
}

private void buangButtonActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    // Membuat JTextField untuk memasukkan ID
    JTextField idField = new JTextField(5);

    // Membuat panel untuk menampung JTextField
    JPanel panel = new JPanel();
    panel.add(new JLabel("Masukkan ID yang ingin dihapus:"));
    panel.add(idField);

    // Menampilkan dialog box dengan JTextField, tombol OK, dan Cancel
    int result = JOptionPane.showConfirmDialog(null, panel,
        "Hapus Mahasiswa", JOptionPane.OK_CANCEL_OPTION,
        JOptionPane.PLAIN_MESSAGE);

    // Jika tombol OK ditekan
    if (result == JOptionPane.OK_OPTION) {
        try {
            // Mengambil input ID dan memanggil metode deleteMhs
            int id = Integer.parseInt(idField.getText());
            controller.deleteMhs(id);
            JOptionPane.showMessageDialog(null, "Data berhasil dihapus.", "Sukses",
                JOptionPane.INFORMATION_MESSAGE);
        } catch (NumberFormatException e) {
            // Menangani error jika ID yang dimasukkan bukan angka
            JOptionPane.showMessageDialog(null, "ID harus berupa angka.", "Error",
                JOptionPane.ERROR_MESSAGE);
        }
    }
}

```

```

/**
 * @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(MahasiswaView.class.getName()).log(java.util.logging.Le
        vel.SEVERE, null, ex);
    } catch (InstantiationException ex) {

        java.util.logging.Logger.getLogger(MahasiswaView.class.getName()).log(java.util.logging.Le
        vel.SEVERE, null, ex);
    } catch (IllegalAccessException ex) {

        java.util.logging.Logger.getLogger(MahasiswaView.class.getName()).log(java.util.logging.Le
        vel.SEVERE, null, ex);
    } catch (javax.swing.UnsupportedLookAndFeelException ex) {

        java.util.logging.Logger.getLogger(MahasiswaView.class.getName()).log(java.util.logging.Le
        vel.SEVERE, null, ex);
    }
    //</editor-fold>

    /* Create and display the form */
    java.awt.EventQueue.invokeLater(new Runnable() {
        public void run() {
            new MahasiswaView().setVisible(true);
        }
    });
}

```



```
// Variables declaration - do not modify
private javax.swing.JTextField IpKField;
private javax.swing.JTextField NamaField;
private javax.swing.JTextField NpmField;
private javax.swing.JTextField SemesterField;
private javax.swing.JButton buangButton;
private javax.swing.JTable dataTable;
private javax.swing.JLabel jLabel1;
private javax.swing.JLabel jLabel2;
private javax.swing.JLabel jLabel3;
private javax.swing.JLabel jLabel4;
private javax.swing.JScrollPane jScrollPane1;
private javax.swing.JScrollPane jScrollPane2;
private javax.swing.JTable jTable1;
private javax.swing.JButton refreshButton;
private javax.swing.JButton simpanButton;
// End of variables declaration
}
```

Design

