Nama : Khoerul Akbr

NPM : 51421607

Kelas : 4IA28

Materi : Spring

Praktikum : Rekayasa Perangkat Lunak 2

Activity Pertemuan 5

Pertemuan5SpringBootApplication.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;

import com.mahasiswa.controller.MahasiswaController;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.boot.CommandLineRunner;
import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;


@SpringBootApplication
public class Pertemuan5SpringBootApplication implements CommandLineRunner{

    @Autowired
    private MahasiswaController mhsController;
    public static void main(String[] args) {
        SpringApplication.run(Pertemuan5SpringBootApplication.class, args);
    }

    @Override
    public void run(String... args) throws Exception {
        mhsController.tampilkanMenu();
    }

}
```

MahasiswaController.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.ModelMahasiswa;
import com.mahasiswa.repository.MahasiswaRepository;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Controller;

import java.util.List;
import java.util.Scanner;

@Controller


public class MahasiswaController {

    @Autowired
    private MahasiswaRepository mahasiswaRepository;

    public void tampilkanMenu() {
        Scanner scanner = new Scanner(System.in);
        int opsi;

        do {
            System.out.println("\nMenu:");
            System.out.println("1. Tampilkan semua mahasiswa");
            System.out.println("2. Tambah mahasiswa baru");
            System.out.println("3. Cek koneksi database");
            System.out.println("4. Keluar");
            System.out.print("Pilih opsi: ");
            opsi = scanner.nextInt();
            scanner.nextLine(); // menangkap newline

            switch (opsi) {
                case 1:
                    tampilkanSemuaMahasiswa();
                    break;
                case 2:
```

```
                tambahMahasiswa(scanner);
                break;
            case 3:
                cekKoneksi();
                break;
            case 4:
                System.out.println("Keluar dari program.");
                break;
            default:
                System.out.println("Opsi tidak valid, coba lagi.");
        }

    } while (opsi != 4);
}

private void tampilkanSemuaMahasiswa() {
    List<ModelMahasiswa> mahasiswaList = mahasiswaRepository.findAll();
    if (mahasiswaList.isEmpty()) {
        System.out.println("Tidak ada data mahasiswa.");
    } else {
        mahasiswaList.forEach(mahasiswa -> System.out.println(mahasiswa));
    }
}

private void tambahMahasiswa(Scanner scanner) {
    System.out.print("Masukkan NPM : ");
    String npm = scanner.nextLine();
    System.out.print("Masukkan Nama : ");
    String nama = scanner.nextLine();
    System.out.print("Masukkan Semester : ");
    int semester = scanner.nextInt();
    System.out.print("Masukkan IPK : ");
    float ipk = scanner.nextFloat();

    ModelMahasiswa mahasiswa = new ModelMahasiswa(0, npm, nama, semester, ipk);
    mahasiswaRepository.save(mahasiswa);
    System.out.println("Mahasiswa berhasil ditambahkan.");
}

private void cekKoneksi() {
    try {
        mahasiswaRepository.findAll();
        System.out.println("Koneksi ke database berhasil.");
    } catch (Exception e) {
            System.out.println("Gagal terhubung ke database.");
        }
    }
}
```

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.*;


@Entity
@Table(name = "mahasiswa")

public class ModelMahasiswa {

    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    @Column(name = "id")
    private int id;

    @Column(name = "npm", nullable = false, length = 8)
    private String npm;

    @Column(name = "nama", nullable = false, length = 50)
    private String nama;

    @Column(name = "semester")
    private int semester;

    @Column(name = "ipk")
    private float ipk;

    public ModelMahasiswa(){

    }
    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;
    }

    @Override
    public String toString() {
        return "Mahasiswa{" +
                "id =" + id +
                ", npm ='" + npm + '\'' +
                ", nama ='" + nama + '\'' +
                ", semester ='" + semester + '\'' +
                ", ipk ='" + ipk + '\'' +
                '}';
    }
}
```

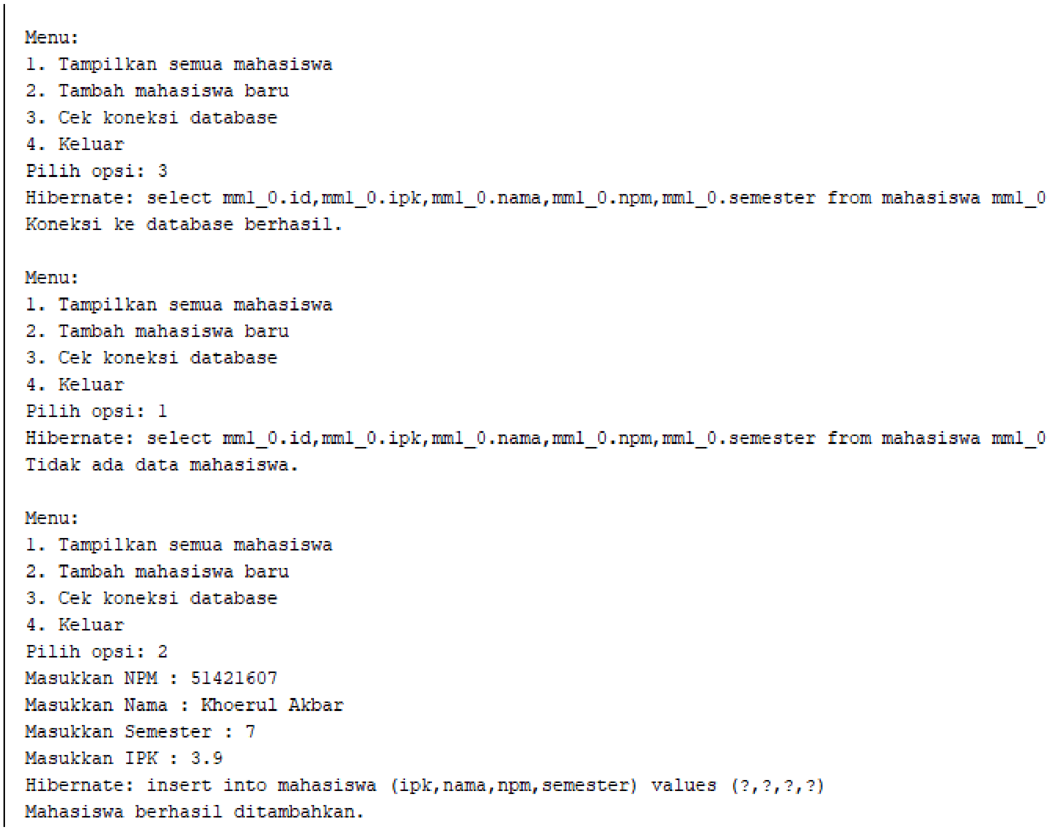
MahasiswaRepository.java

```
package com.mahasiswa.repository;

import com.mahasiswa.model.ModelMahasiswa;
import org.springframework.data.jpa.repository.JpaRepository;
import org.springframework.stereotype.Repository;

@Repository
public interface MahasiswaRepository extends JpaRepository<ModelMahasiswa, Long> {

}
```

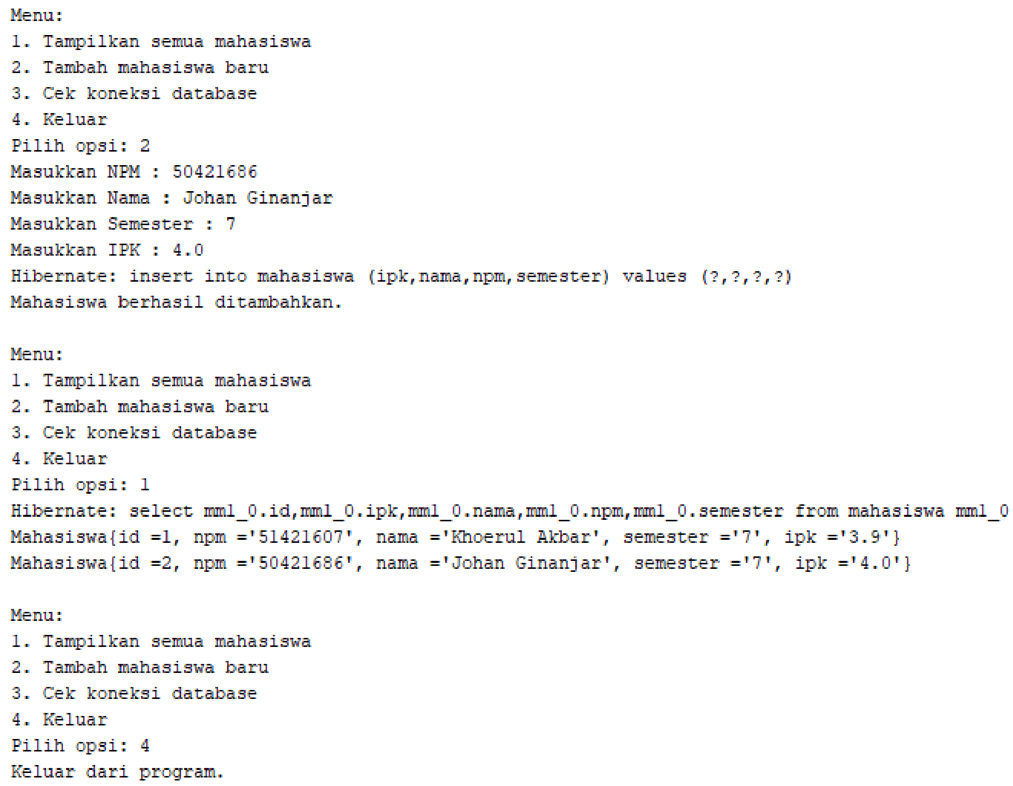
Application.properties

```
# Konfigurasi MySQL Hibernate
spring.datasource.url=jdbc:mysql://localhost:3306/rpl_pert5?useSSL=false&serverTimezone=
UTC
spring.datasource.username=root
spring.datasource.password=
spring.datasource.driver-class-name=com.mysql.cj.jdbc.Driver

# Hibernate settings
spring.jpa.hibernate.ddl-auto=update
spring.jpa.show-sql=true
```

Output:



```
Menu:
1. Tampilkan semua mahasiswa
2. Tambah mahasiswa baru
3. Cek koneksi database
4. Keluar
Pilih opsi: 3
Hibernate: select m1_0.id,m1_0.ipk,m1_0.nama,m1_0.npm,m1_0.semester from mahasiswa m1_0
Koneksi ke database berhasil.

Menu:
1. Tampilkan semua mahasiswa
2. Tambah mahasiswa baru
3. Cek koneksi database
4. Keluar
Pilih opsi: 1
Hibernate: select m1_0.id,m1_0.ipk,m1_0.nama,m1_0.npm,m1_0.semester from mahasiswa m1_0
Tidak ada data mahasiswa.

Menu:
1. Tampilkan semua mahasiswa
2. Tambah mahasiswa baru
3. Cek koneksi database
4. Keluar
Pilih opsi: 2
Masukkan NPM : 51421607
Masukkan Nama : Khoerul Akbar
Masukkan Semester : 7
Masukkan IPK : 3.9
Hibernate: insert into mahasiswa (ipk,nama,npm,semester) values (?,?,?,?)
Mahasiswa berhasil ditambahkan.
```



```
Menu:
1. Tampilkan semua mahasiswa
2. Tambah mahasiswa baru
3. Cek koneksi database
4. Keluar
Pilih opsi: 2
Masukkan NPM : 50421686
Masukkan Nama : Johan Ginanjar
Masukkan Semester : 7
Masukkan IPK : 4.0
Hibernate: insert into mahasiswa (ipk,nama,npm,semester) values (?,?,?,?)
Mahasiswa berhasil ditambahkan.

Menu:
1. Tampilkan semua mahasiswa
2. Tambah mahasiswa baru
3. Cek koneksi database
4. Keluar
Pilih opsi: 1
Hibernate: select m1_0.id,m1_0.ipk,m1_0.nama,m1_0.npm,m1_0.semester from mahasiswa m1_0
Mahasiswa{id =1, npm ='51421607', nama ='Khoerul Akbar', semester ='7', ipk ='3.9'}
Mahasiswa{id =2, npm ='50421686', nama ='Johan Ginanjar', semester ='7', ipk ='4.0'}

Menu:
1. Tampilkan semua mahasiswa
2. Tambah mahasiswa baru
3. Cek koneksi database
4. Keluar
Pilih opsi: 4
Keluar dari program.
```

Activity Pertemuan 6

MahasiswaApp.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;

import com.mahasiswa.controller.MahasiswaController;
import com.mahasiswa.service.MahasiswaService;
import com.mahasiswa.view.MahasiswaView;
import org.springframework.boot.ApplicationArguments;
import org.springframework.boot.ApplicationRunner;
import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.context.ApplicationContext;


@SpringBootApplication
public class MahasiswaApp implements ApplicationRunner {

    @Autowired
    private MahasiswaService mahasiswaService;

    public static void main(String[] args) {
        System.setProperty("java.awt.headless", "false"); // Disable headless mode

        // Start the Spring application and get the application context
        ApplicationContext context = SpringApplication.run(MahasiswaApp.class, args);

        // Instantiate the view and inject the controller manually
        MahasiswaController controller = context.getBean(MahasiswaController.class);
        MahasiswaView mahasiswaView = new MahasiswaView(controller);
        mahasiswaView.setVisible(true);
    }

    @Override
    public void run(ApplicationArguments args) throws Exception {
        // Implement this method if you need to execute logic after Spring application starts
        // Otherwise, you can leave it as is.
    }
}
```

MahasiswaController.java

```
package com.mahasiswa.controller;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.web.bind.annotation.*;
import com.mahasiswa.model.ModelMahasiswa;
import com.mahasiswa.service.MahasiswaService;

import java.util.List;
import org.springframework.stereotype.Controller;


@Controller
public class MahasiswaController {

    @Autowired
    private MahasiswaService mahasiswaService;

    // Add new Mahasiswa
    public String addMahasiswa(@RequestBody ModelMahasiswa mhs) {
        mahasiswaService.addMhs(mhs);
        return "Mahasiswa added successfully";
    }

    // Get Mahasiswa by ID
    public ModelMahasiswa getMahasiswa(@PathVariable int id) {
        return mahasiswaService.getMhs(id);
    }

    // Update Mahasiswa
    public String updateMahasiswa(@RequestBody ModelMahasiswa mhs) {
        mahasiswaService.updateMhs(mhs);
        return "Mahasiswa updated successfully";
    }

    // Delete Mahasiswa by ID
    public String deleteMahasiswa(@PathVariable int id) {
        mahasiswaService.deleteMhs(id);
        return "Mahasiswa deleted successfully";
    }

    // Get all Mahasiswa
    public List<ModelMahasiswa> getAllMahasiswa() {
        return mahasiswaService.getAllMahasiswa();
    }
}
```

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.*;


@Entity
@Table(name = "mahasiswa")

public class ModelMahasiswa {

    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    @Column(name = "id")
    private int id;

    @Column(name = "npm", nullable = false, length = 8)
    private String npm;

    @Column(name = "nama", nullable = false, length = 50)
    private String nama;

    @Column(name = "semester")
    private int semester;

    @Column(name = "ipk")
    private float ipk;

    public ModelMahasiswa(){
    }

    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;
    }
}
```

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



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
        }
    }
```

MahasiswaPepository.java

```
package com.mahasiswa.repository;

import com.mahasiswa.model.ModelMahasiswa;
import org.springframework.data.jpa.repository.JpaRepository;
import org.springframework.stereotype.Repository;

@Repository
```

```
public interface MahasiswaRepository extends JpaRepository<ModelMahasiswa, Integer> {
   public Object findById(int id);

   public void deleteById(int id);

}
```

MahasiswaService.java

```
package com.mahasiswa.service;

import com.mahasiswa.model.ModelMahasiswa;
import com.mahasiswa.repository.MahasiswaRepository;
import jakarta.transaction.Transactional;
import java.util.List;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Service;

@Service
public class MahasiswaService {

   @Autowired
   private MahasiswaRepository repository;

   public void addMhs(ModelMahasiswa mhs) {
      repository.save(mhs);
   }

   public ModelMahasiswa getMhs(int id) {
      ModelMahasiswa mahasiswa = (ModelMahasiswa) repository.findById(id);
      return mahasiswa != null ? mahasiswa : null;
   }

   public void updateMhs(ModelMahasiswa mhs) {
      repository.save(mhs);
   }

   @Transactional
   public void deleteMhs(int id) {
      repository.deleteById(id);
   }

   public List<ModelMahasiswa> getAllMahasiswa() {
        return repository.findAll();
    }
}
```

application.resources

```
# Konfigurasi MySQL Hibernate
spring.datasource.url=jdbc:mysql://localhost:3306/rpl_pert6?useSSL=false&serverTimezone=UTC
spring.datasource.username=root
spring.datasource.password=
spring.datasource.driver-class-name=com.mysql.cj.jdbc.Driver

# Hibernate settings
spring.jpa.hibernate.ddl-auto=update
spring.jpa.show-sql=true
```

MahasiswaView.java

Source Code:

```
/*
 * 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.MahasiswaController;
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 Maulana
 */
```

```
public class MahasiswaView extends javax.swing.JFrame {
    private MahasiswaController controller;

    /**
     * Creates new form MahasiswaView
     */
    public MahasiswaView(MahasiswaController controller) {
        this.controller = controller;
        initComponents();
        loadMahasiswaTable();
    }

     private MahasiswaView() {
        throw new UnsupportedOperationException("Not supported yet."); // Generated from
nbfs://nbhost/SystemFileSystem/Templates/Classes/Code/GeneratedMethodBody
    }

     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() {

        jLabel1 = new javax.swing.JLabel();
        jLabel2 = new javax.swing.JLabel();
        jLabel3 = new javax.swing.JLabel();
        jLabel4 = new javax.swing.JLabel();
        npmField = new javax.swing.JTextField();
        namaField = new javax.swing.JTextField();
        semesterField = new javax.swing.JTextField();
```

```
ipkField = new javax.swing.JTextField();
simpanButton = new javax.swing.JButton();
buangButton = new javax.swing.JButton();
jScrollPane1 = new javax.swing.JScrollPane();
dataTable = new javax.swing.JTable();

setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);

jLabel1.setText("NPM");

jLabel2.setText("Nama");

jLabel3.setText("Semester");

jLabel4.setText("IPK");

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

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

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"
   }
));
jScrollPane1.setViewportView(dataTable);

javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
```

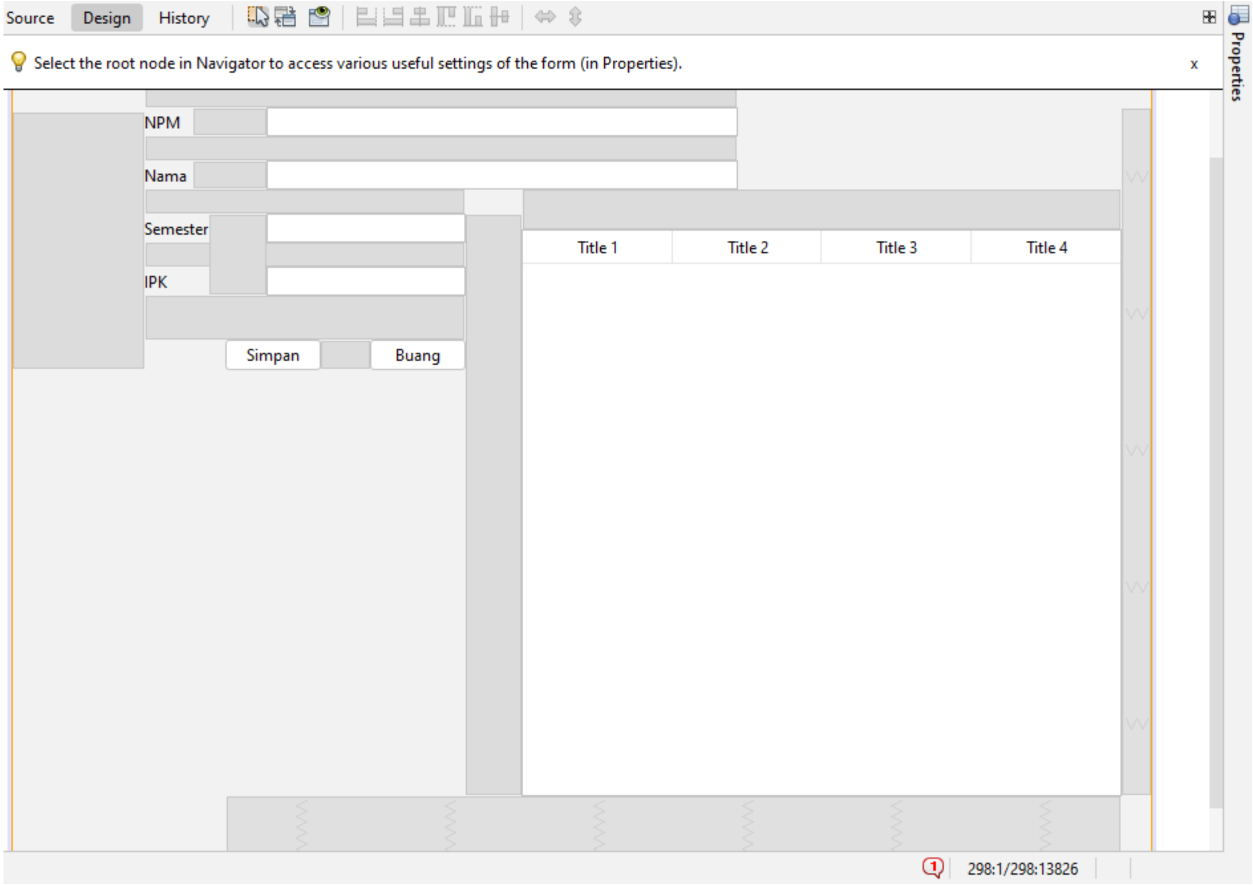
```
        getContentPane().setLayout(layout);
        layout.setHorizontalGroup(
            layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(layout.createSequentialGroup()
                .addGap(99, 99, 99)

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

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)
                        .addGroup(layout.createSequentialGroup()
                            .addComponent(jLabel1, javax.swing.GroupLayout.PREFERRED_SIZE, 37,
javax.swing.GroupLayout.PREFERRED_SIZE)
                            .addGap(55, 55, 55)
                            .addComponent(npmField, javax.swing.GroupLayout.PREFERRED_SIZE, 355,
javax.swing.GroupLayout.PREFERRED_SIZE))
                        .addGroup(layout.createSequentialGroup()
                            .addComponent(jLabel2, javax.swing.GroupLayout.PREFERRED_SIZE, 37,
javax.swing.GroupLayout.PREFERRED_SIZE)
                            .addGap(55, 55, 55)
                            .addComponent(namaField)))
                    .addGroup(layout.createSequentialGroup()

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

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                                    .addComponent(jLabel3, javax.swing.GroupLayout.PREFERRED_SIZE, 49,
javax.swing.GroupLayout.PREFERRED_SIZE)
                                    .addComponent(jLabel4, javax.swing.GroupLayout.PREFERRED_SIZE, 37,
javax.swing.GroupLayout.PREFERRED_SIZE))
                                .addGap(43, 43, 43)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)
                                    .addComponent(semesterField, javax.swing.GroupLayout.DEFAULT_SIZE,
150, Short.MAX_VALUE)
                                    .addComponent(ipkField)))
                            .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
layout.createSequentialGroup()
                                .addComponent(simpanButton)
                                .addGap(37, 37, 37)
                                .addComponent(buangButton)))
                        .addGap(42, 42, 42)
                        .addComponent(jScrollPane1, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)))
```

```
        .addContainerGap(22, Short.MAX_VALUE))
    );
    layout.setVerticalGroup(
      layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
      .addGroup(layout.createSequentialGroup()
        .addGap(57, 57, 57)

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

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
          .addComponent(jLabel2)
          .addComponent(namaField, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
          .addGroup(layout.createSequentialGroup()
            .addGap(18, 18, 18)

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

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
              .addComponent(jLabel4)
              .addComponent(ipkField, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
            .addGap(33, 33, 33)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
              .addComponent(simpanButton)
              .addComponent(buangButton)))
          .addGroup(layout.createSequentialGroup()
            .addGap(30, 30, 30)
            .addComponent(jScrollPane1, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)))
        .addContainerGap(62, Short.MAX_VALUE))
    );
```

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

    private void simpanButtonActionPerformed(java.awt.event.ActionEvent evt) {
     String npm = getNpmField().getText();
     String nama = getNamaField().getText();
     int semester = Integer.parseInt(getSemesterField().getText());
     float ipk = Float.parseFloat(getIpkField().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.addMahasiswa(mahasiswa);
     loadMahasiswaTable();
    }

    private void buangButtonActionPerformed(java.awt.event.ActionEvent evt) {
      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.deleteMahasiswa(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);
      }
    }
    loadMahasiswaTable();
    }

    public JTextField getIpkField() {
        return ipkField;
    }

    public void setIpkField(JTextField ipkField) {
        this.ipkField = ipkField;
    }

    public JTextField getNamaField() {
        return namaField;
    }

    public void setNamaField(JTextField namaField) {
        this.namaField = namaField;
    }

    public JTextField getNpmField() {
        return npmField;
    }

    public void setNpmField(JTextField npmField) {
        this.npmField = npmField;
    }

    public JTextField getSemesterField() {
        return semesterField;
    }

    public void setSemesterField(JTextField semesterField) {
        this.semesterField = semesterField;
    }

    /**
     * @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.Leve
l.SEVERE, null, ex);
        } catch (InstantiationException ex) {

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

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

java.util.logging.Logger.getLogger(MahasiswaView.class.getName()).log(java.util.logging.Leve
l.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.JButton buangButton;
    private javax.swing.JTable dataTable;
    private javax.swing.JTextField ipkField;
    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.JTextField namaField;
    private javax.swing.JTextField npmField;
    private javax.swing.JTextField semesterField;
    private javax.swing.JButton simpanButton;
    // End of variables declaration
}
```

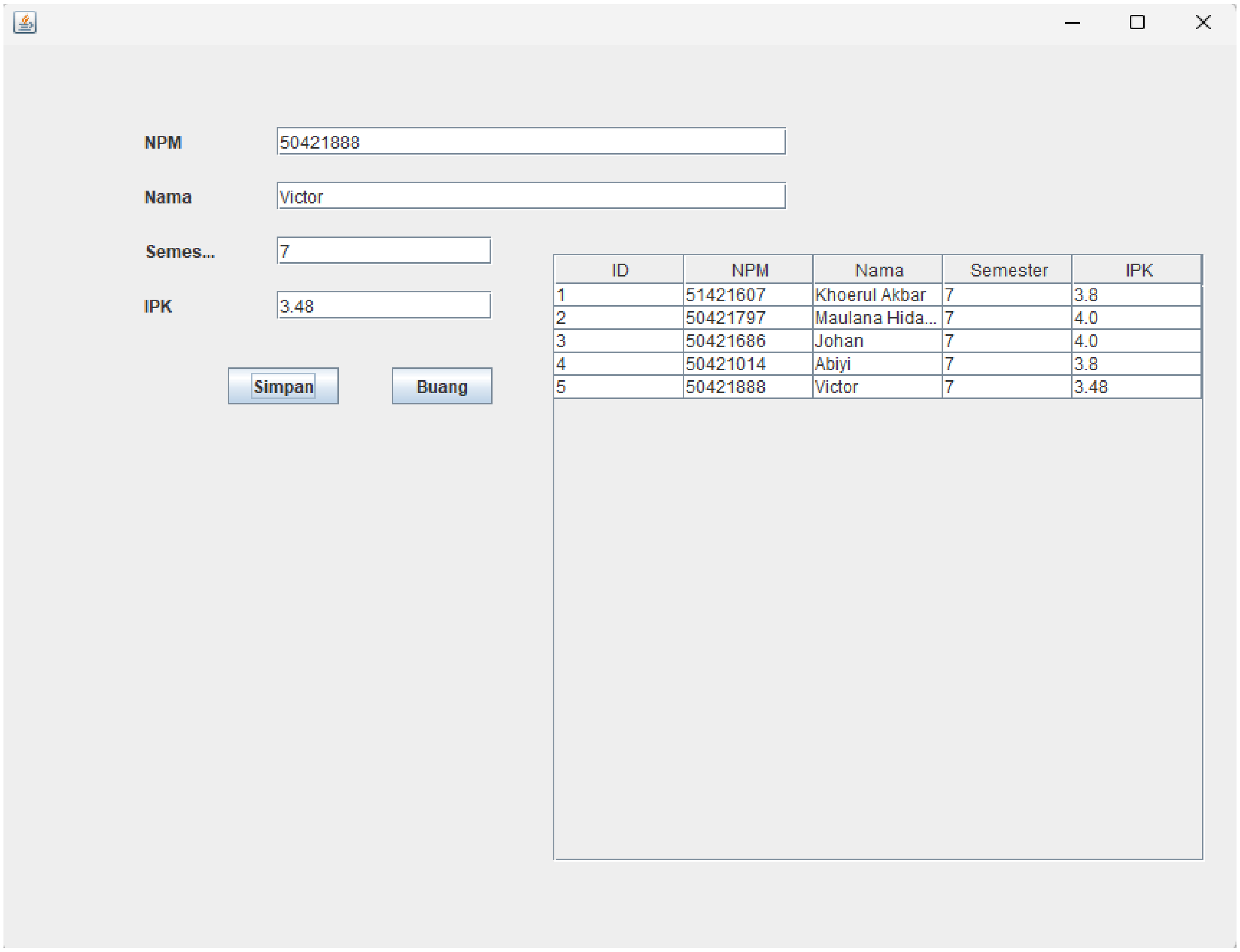
Design:



Select the root node in Navigator to access various useful settings of the form (in Properties).

Output:

Insert 5 data



NPM 50421888

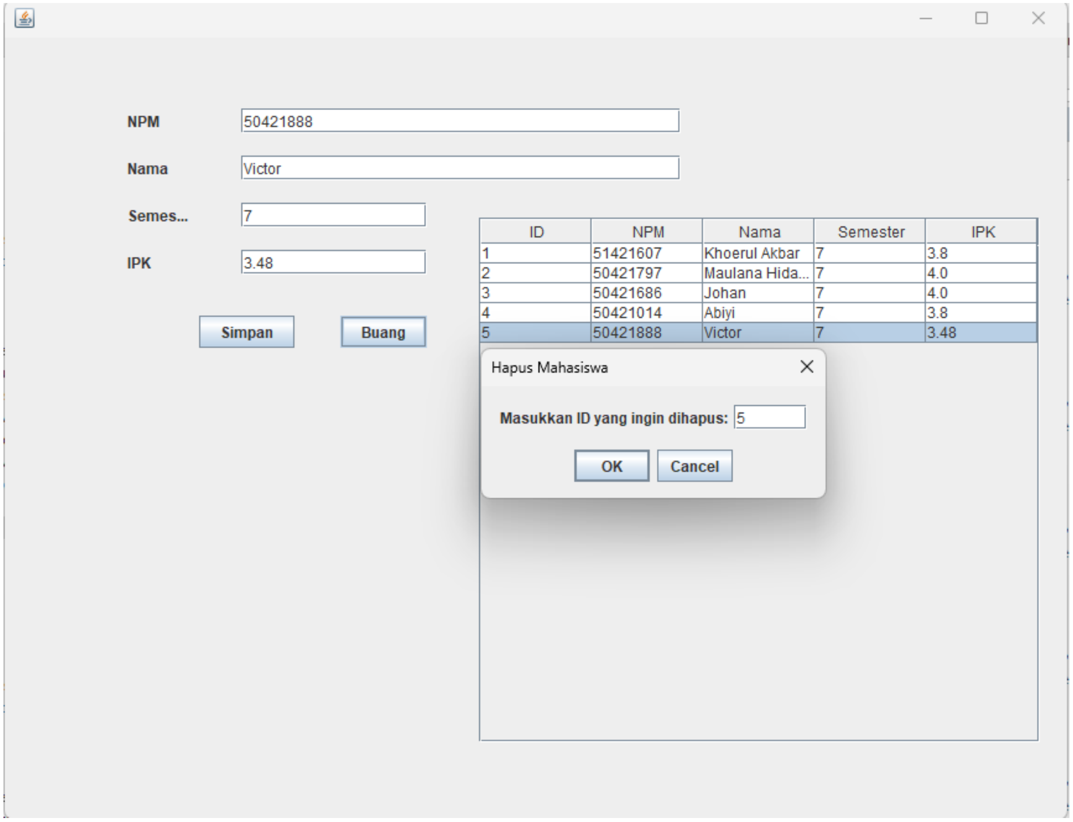
Nama Victor

Semes... 7

IPK 3.48

ID	NPM	Nama	Semester	IPK
1	51421607	Khoerul Akbar	7	3.8
2	50421797	Maulana Hida...	7	4.0
3	50421686	Johan	7	4.0
4	50421014	Abiyi	7	3.8
5	50421888	Victor	7	3.48

Hapus 1 data



NPM 50421888

Nama Victor

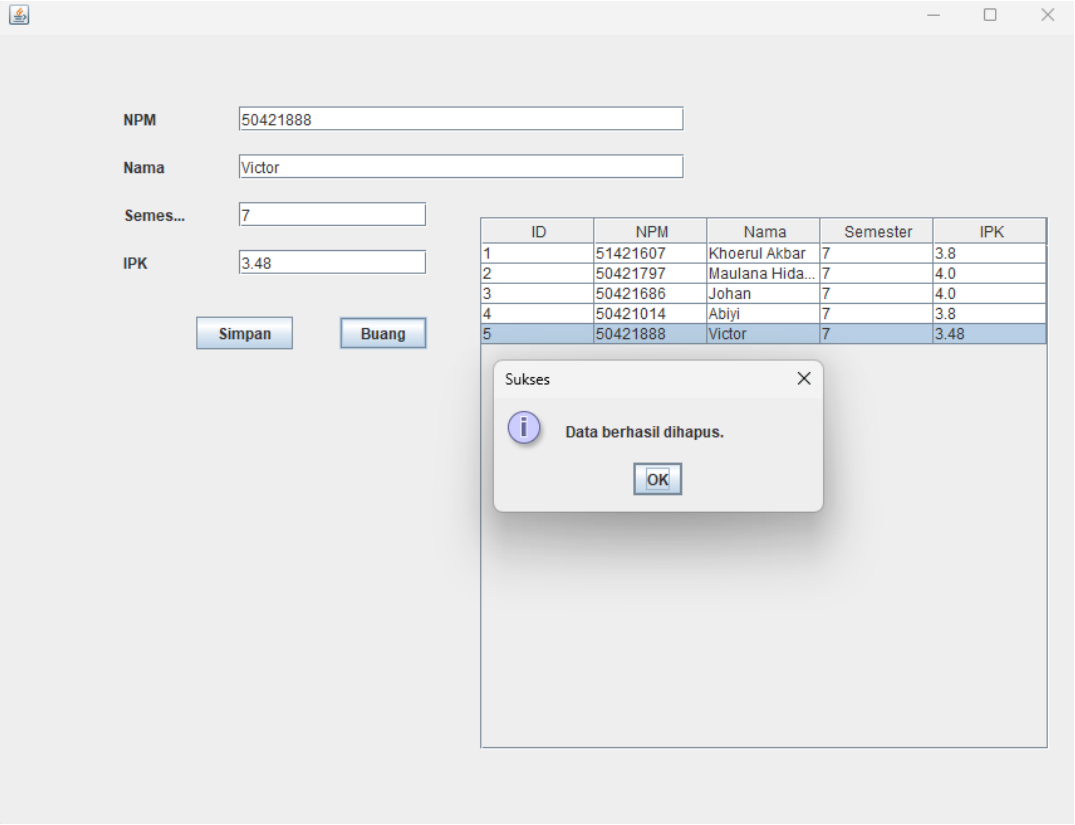
Semes... 7

IPK 3.48

ID	NPM	Nama	Semester	IPK
1	51421607	Khoerul Akbar	7	3.8
2	50421797	Maulana Hida...	7	4.0
3	50421686	Johan	7	4.0
4	50421014	Abiyi	7	3.8
5	50421888	Victor	7	3.48

Hapus Mahasiswa

Masukkan ID yang ingin dihapus: 5



NPM 50421888

Nama Victor

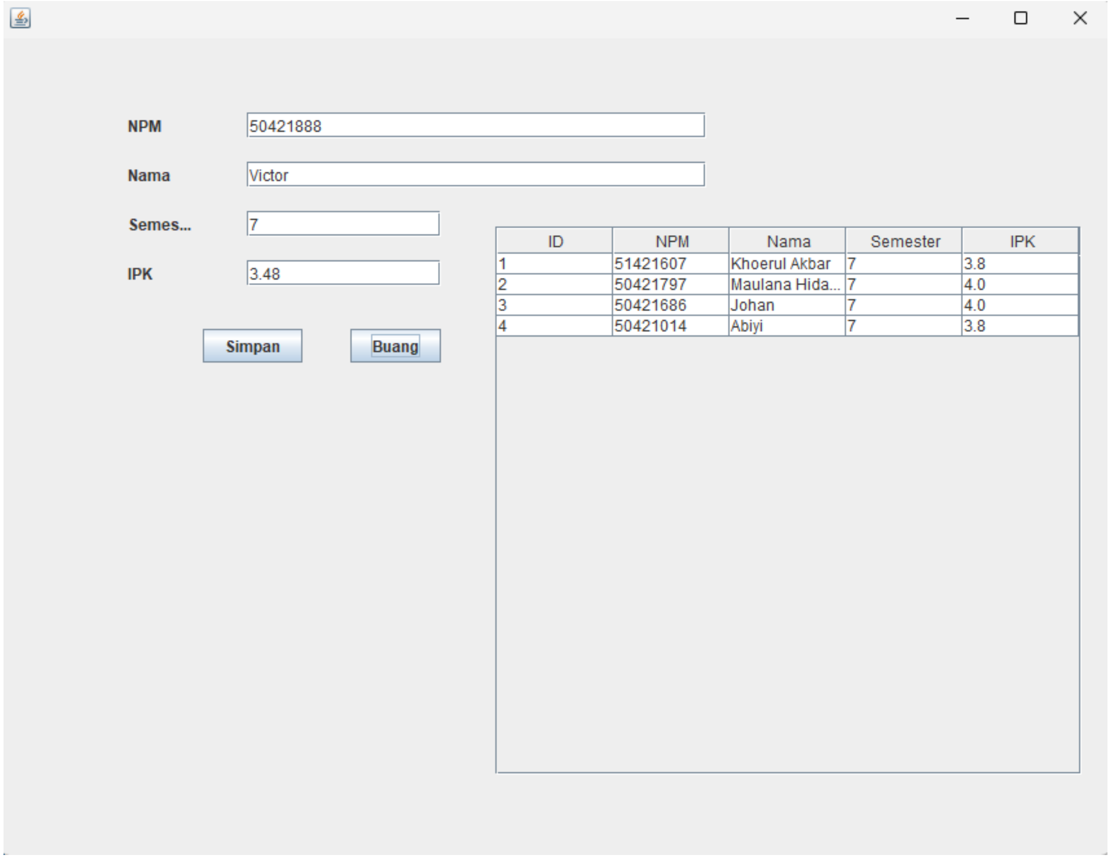
Semes... 7

IPK 3.48

ID	NPM	Nama	Semester	IPK
1	51421607	Khoerul Akbar	7	3.8
2	50421797	Maulana Hida...	7	4.0
3	50421686	Johan	7	4.0
4	50421014	Abiyi	7	3.8
5	50421888	Victor	7	3.48

Sukses

Data berhasil dihapus.



NPM: 50421888

Nama: Victor

Semes...: 7

IPK: 3.48

ID	NPM	Nama	Semester	IPK
1	51421607	Khoerul Akbar	7	3.8
2	50421797	Maulana Hida...	7	4.0
3	50421686	Johan	7	4.0
4	50421014	Abiyi	7	3.8