ACTIVITY PERTEMUAN 5 & 6

Nama : Maulana Hidayatulloh Mujanah

NPM : 50421797

Kelas : 4IA28

Praktikum RPL 2

Pertemuan 5

Code :

pom.xml

```xml
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://maven.apache.org/POM
    <modelVersion>4.0.0</modelVersion>
    <groupId>com.mycompany</groupId>
    <artifactId>RPL_MaulanaHidayatulloh_Spring</artifactId>
    <version>1.0-SNAPSHOT</version>
    <packaging>jar</packaging>
    <properties>
        <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
        <maven.compiler.source>21</maven.compiler.source>
        <maven.compiler.target>21</maven.compiler.target>
        <exec.mainClass>com.mycompany.rpl_maulanahidayatulloh_spring.RPL_MaulanaHidayatulloh_Spring</exec.mainClass>
    </properties>
    <parent>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-parent</artifactId>
    <version>3.3.3</version>
    <relativePath/>
</parent>

<dependencies>
    <!-- Hibernate + Spring Data JPA -->
    <dependency>
        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-starter-data-jpa</artifactId>
    </dependency>

    <!-- MySQL Connector -->
    <dependency>
        <groupId>mysql</groupId>
        <artifactId>mysql-connector-java</artifactId>
        <version>8.0.33</version>
    </dependency>

    <!-- Spring Boot Web dependency (for MVC if needed) -->
    <dependency>
```

```xml
        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-starter-web</artifactId>
    </dependency>

    <!-- Testing dependencies -->
    <dependency>
        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-starter-test</artifactId>
        <scope>test</scope>
    </dependency>
</dependencies>

<build>
    <plugins>
        <plugin>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-maven-plugin</artifactId>
        </plugin>
    </plugins>
</build>
</project>
```

Application.properties

```
# Konfigurasi MySQL Hibernate
spring.datasource.url=jdbc:mysql://localhost:3306/db_rpl2_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
```

Pertemuan5SpringBootApplication.java

```
...4 lines
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;

/**
 *
 * @author Maulana
 */

@SpringBootApplication
public class Pertemuan5SpringBootApplication implements CommandLineRunner{

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

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

}
```

ModelMahasiswa.java

```java
...4 lines
package com.mahasiswa.model;

import jakarta.persistence.*;

/**
 *
 * @author Maulana
 */

@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) {
```

```java
        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;
```

```java
    }

    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 + '\'' +
                '}';
    }
}
```

MahasiswaController

```java
...4 lines
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

/**
 *
 * @author Maulana
 */
public class MahasiswaController {

    @Autowired
    private MahasiswaRepository mahasiswaRepository;

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

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

            switch (opsi) {
```

```java
                    case 1:
                        tampilkanSemuaMahasiswa();
                        break;
                    case 2:
                        tambahMahasiswa(scanner);
                        break;
                    case 3:
                        cekKoneksi();
                        break;
                    case 4:
                        System.out.println(x: "Keluar dari program.");
                        break;
                    default:
                        System.out.println(x: "Opsi tidak valid, coba lagi.");
                }

            } while (opsi != 4);
        }

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

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

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

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

```

MahasiswaRepository.java

```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> {

}
```

Output :

Cek Koneksi Database

```
Menu:
1. Tampilkan semua mahasiswa
2. Tambah mahasiswa baru
3. Cek koneksi database
4. Keluar
Pilih opsi: 3
Hibernate: select mm1_0.id,mm1_0.ipk,mm1_0.nama,mm1_0.npm,mm1_0.semester from mahasiswa mm1_0
Koneksi ke database berhasil.
```

Tambah Data Mahasiswa

```
Menu:
1. Tampilkan semua mahasiswa
2. Tambah mahasiswa baru
3. Cek koneksi database
4. Keluar
Pilih opsi: 2
Masukkan NPM : 50421797
Masukkan Nama : Maulana
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: 2
Masukkan NPM : 12312312
Masukkan Nama : Panjul
Masukkan Semester : 15
Masukkan IPK : 2.5
Hibernate: insert into mahasiswa (ipk,nama,npm,semester) values (?,?,?,?)
Mahasiswa berhasil ditambahkan.
```

Tampilkan Data Mahasiswa

```
Menu:
1. Tampilkan semua mahasiswa
2. Tambah mahasiswa baru
3. Cek koneksi database
4. Keluar
Pilih opsi: 1
Hibernate: select mm1_0.id,mm1_0.ipk,mm1_0.nama,mm1_0.npm,mm1_0.semester from mahasiswa mm1_0
Mahasiswa{id =1, npm ='50421797', nama ='Maulana', semester ='7', ipk ='4.0'}
Mahasiswa{id =2, npm ='12312312', nama ='Panjul', semester ='15', ipk ='2.5'}
```

Pertemuan 6

Code :

pom.xml

```xml
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://maven.apache.org/POM
    <modelVersion>4.0.0</modelVersion>
    <groupId>com.mycompany</groupId>
    <artifactId>RPL_MaulanaHidayatulloh_SpringAOP</artifactId>
    <version>1.0-SNAPSHOT</version>
    <packaging>jar</packaging>
    <properties>
        <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
        <maven.compiler.source>21</maven.compiler.source>
        <maven.compiler.target>21</maven.compiler.target>
        <exec.mainClass>com.mycompany.rpl_maulanahidayatulloh_springaop.RPL_MaulanaHidayatulloh_SpringAOP</exec.mainClass>
    </properties>
    <parent>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-parent</artifactId>
    <version>3.3.3</version>
    <relativePath/>
</parent>

<dependencies>
    <!-- Hibernate + Spring Data JPA -->
    <dependency>
        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-starter-data-jpa</artifactId>
    </dependency>

    <!-- MySQL Connector -->
    <dependency>
        <groupId>mysql</groupId>
        <artifactId>mysql-connector-java</artifactId>
        <version>8.0.33</version>
    </dependency>

    <!-- Spring Boot Web dependency (for MVC if needed) -->
    <dependency>
```

```xml
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-starter-web</artifactId>
        </dependency>

        <!-- Testing dependencies -->
        <dependency>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-starter-test</artifactId>
            <scope>test</scope>
        </dependency>
    </dependencies>

    <build>
        <plugins>
            <plugin>
                <groupId>org.springframework.boot</groupId>
                <artifactId>spring-boot-maven-plugin</artifactId>
            </plugin>
        </plugins>
    </build>
    </project>
```

application.properties

```
# Konfigurasi MySQL Hibernate
spring.datasource.url=jdbc:mysql://localhost:3306/db_rpl2_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
```

MahasiswaApp.java

```
...4 lines
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;

/**
 *
 * @author Maulana
 */

@SpringBootApplication
public class MahasiswaApp implements ApplicationRunner {

    @Autowired
    private MahasiswaService mahasiswaService;

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

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

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

ModelMahasiswa.java

```java
...4 lines
package com.mahasiswa.model;

import jakarta.persistence.*;

/**
 *
 * @author Maulana
 */

@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) {
```

```java
        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;
```

```java
        }

        public float getIpk() {
            return ipk;
        }

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

```

ModelTabelMahasiswa.java

```java
...4 lines
package com.mahasiswa.model;

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

/**
 *
 * @author Maulana
 */



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(index: rowIndex);
        switch (columnIndex) {
            case 0:
                return mahasiswa.getId();
```

```java
            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
    }
}

```

MahasiswaRepository.java

```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

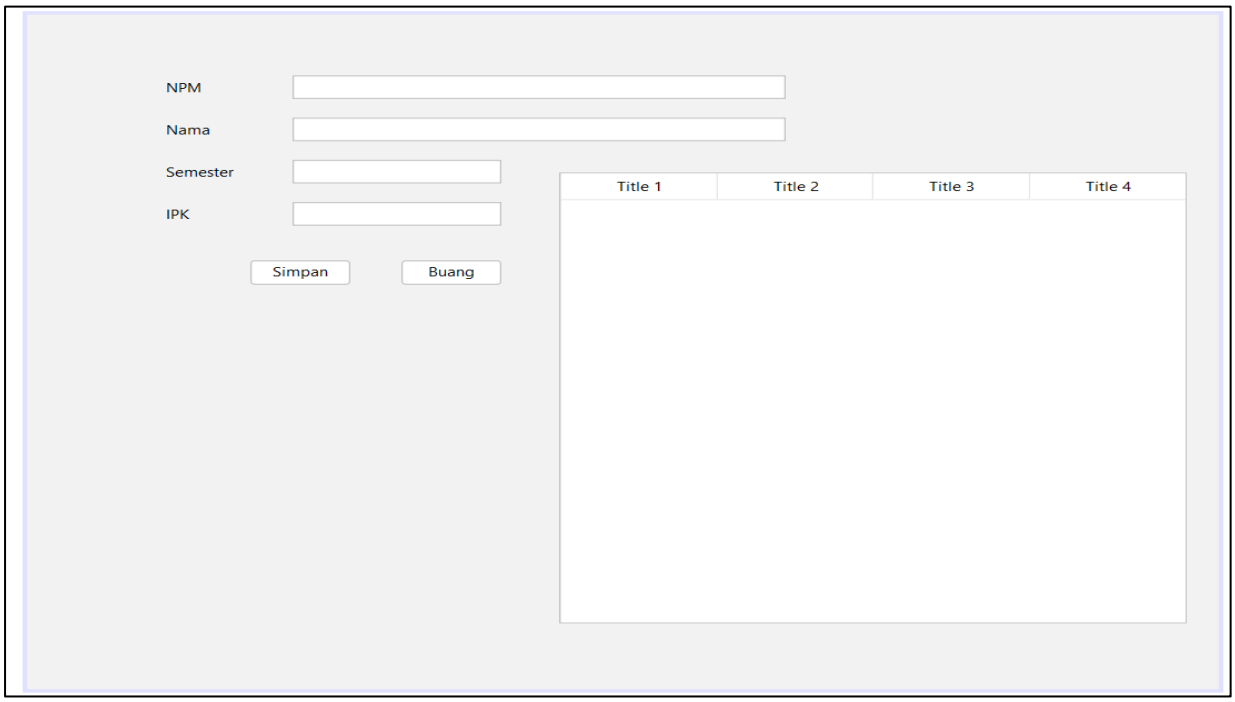
```java
package com.mahasiswa.service;

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(entity:mhs);
    }

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

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

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

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

MahasiswaController.java

```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);
```

```java
        return "Mahasiswa deleted successfully";
    }

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

MahasiswaView.java (Design)



MahasiswaView.java (Code)

```java
...4 lines
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/Ge
    }

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

```

```java
40          // Buat model tabel kustom dengan data mahasiswa
41          ModelTabelMahasiswa tableModel = new ModelTabelMahasiswa(mahasiswaList: listMahasiswa);
42
43          // Set model pada JTable
44          dataTable.setModel(dataModel:tableModel);
45      }
46
47          /**
48           * This method is called from within the constructor to initialize the form.
49           * WARNING: Do NOT modify this code. The content of this method is always
50           * regenerated by the Form Editor.
51           */
52          @SuppressWarnings("unchecked")
53          Generated Code
173
174         private void simpanButtonActionPerformed(java.awt.event.ActionEvent evt) {
175             String npm = getNpmField().getText();
176             String nama = getNamaField().getText();
177             int semester = Integer.parseInt(s: getSemesterField().getText());
178             float ipk = Float.parseFloat(s: getIpkField().getText());
179             ModelMahasiswa mahasiswa = new ModelMahasiswa(id: 0, npm, nama, semester, ipk);
180             System.out.println(x: mahasiswa.getIpk());
181             System.out.println(x: mahasiswa.getNama());
182             System.out.println(x: mahasiswa.getSemester());
183             System.out.println(x: mahasiswa.getNpm());
184
185             controller.addMahasiswa(mhs:mahasiswa);
186             loadMahasiswaTable();
187         }
188
189         private void buangButtonActionPerformed(java.awt.event.ActionEvent evt) {
190             JTextField idField = new JTextField(columns: 5);
191
192         // Membuat panel untuk menampung JTextField
193         JPanel panel = new JPanel();
194         panel.add(new JLabel(text: "Masukkan ID yang ingin dihapus:"));
```

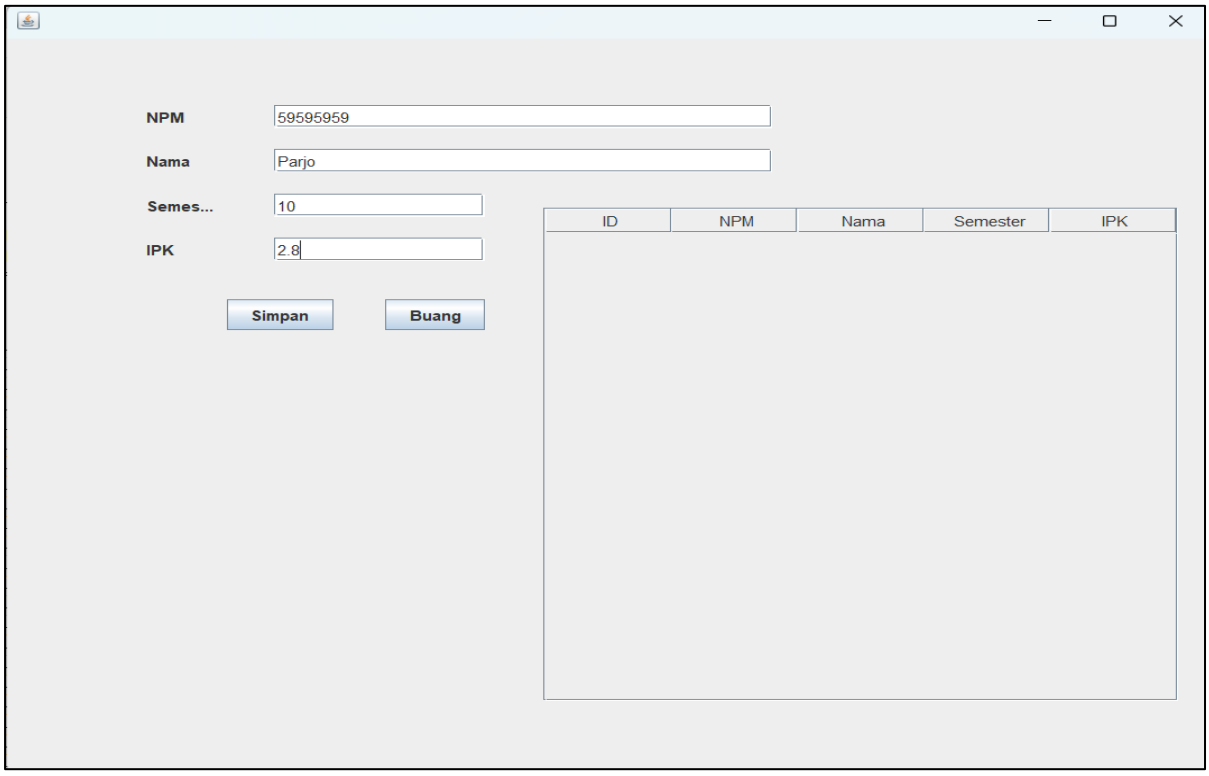
```java
195         panel.add(comp: idField);
196
197         // Menampilkan dialog box dengan JTextField, tombol OK, dan Cancel
198         int result = JOptionPane.showConfirmDialog(parentComponent: null, message: panel,
199             title: "Hapus Mahasiswa", optionType: JOptionPane.OK_CANCEL_OPTION, messageType: JOptionPane.PLAIN_MESSAGE);
200
201         // Jika tombol OK ditekan
202         if (result == JOptionPane.OK_OPTION) {
203             try {
204                 // Mengambil input ID dan memanggil metode deleteMhs
205                 int id = Integer.parseInt(s: idField.getText());
206                 controller.deleteMahasiswa(id);
207                 JOptionPane.showMessageDialog(parentComponent: null, message: "Data berhasil dihapus.", title: "Sukses", messageType: JOptionPane.INFORMATION_MESSAGE);
208             } catch (NumberFormatException e) {
209                 // Menangani error jika ID yang dimasukkan bukan angka
210                 JOptionPane.showMessageDialog(parentComponent: null, message: "ID harus berupa angka.", title: "Error", messageType: JOptionPane.ERROR_MESSAGE);
211             }
212         }
213         loadMahasiswaTable();
214         }
215
216         public JTextField getIpkField() {
217             return ipkField;
218         }
219
220         public void setIpkField(JTextField ipkField) {
221             this.ipkField = ipkField;
222         }
223
224         public JTextField getNamaField() {
225             return namaField;
226         }
227
228         public void setNamaField(JTextField namaField) {
229             this.namaField = namaField;
230         }
```

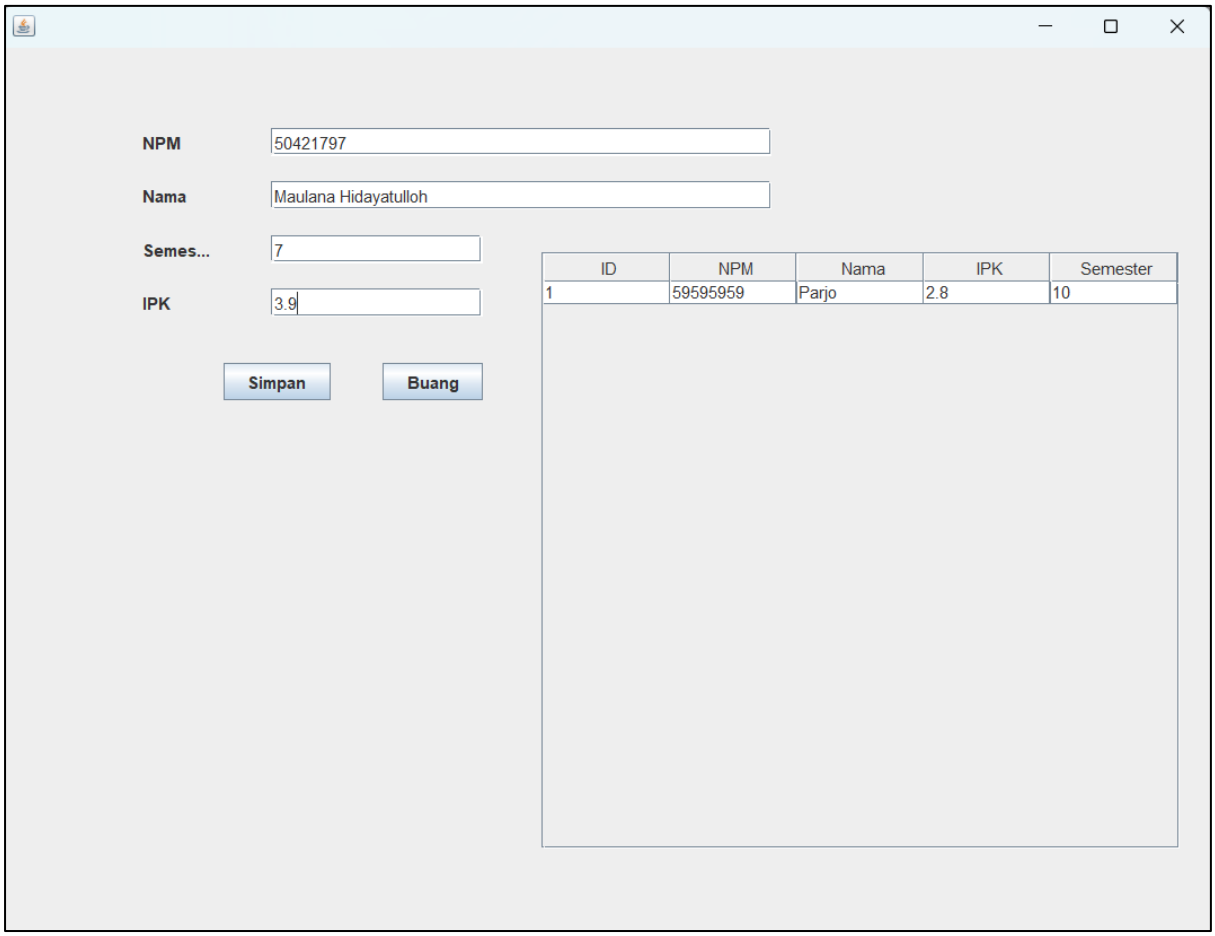
```java
    public JTextField getNpmField() {
        return npmField;
    }

    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 */
        Look and feel setting code (optional)

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

    // Variables declaration - do not modify
    private javax.swing.JButton buangButton;
    private javax.swing.JTable dataTable;
    private javax.swing.JTextField ipkField;
```

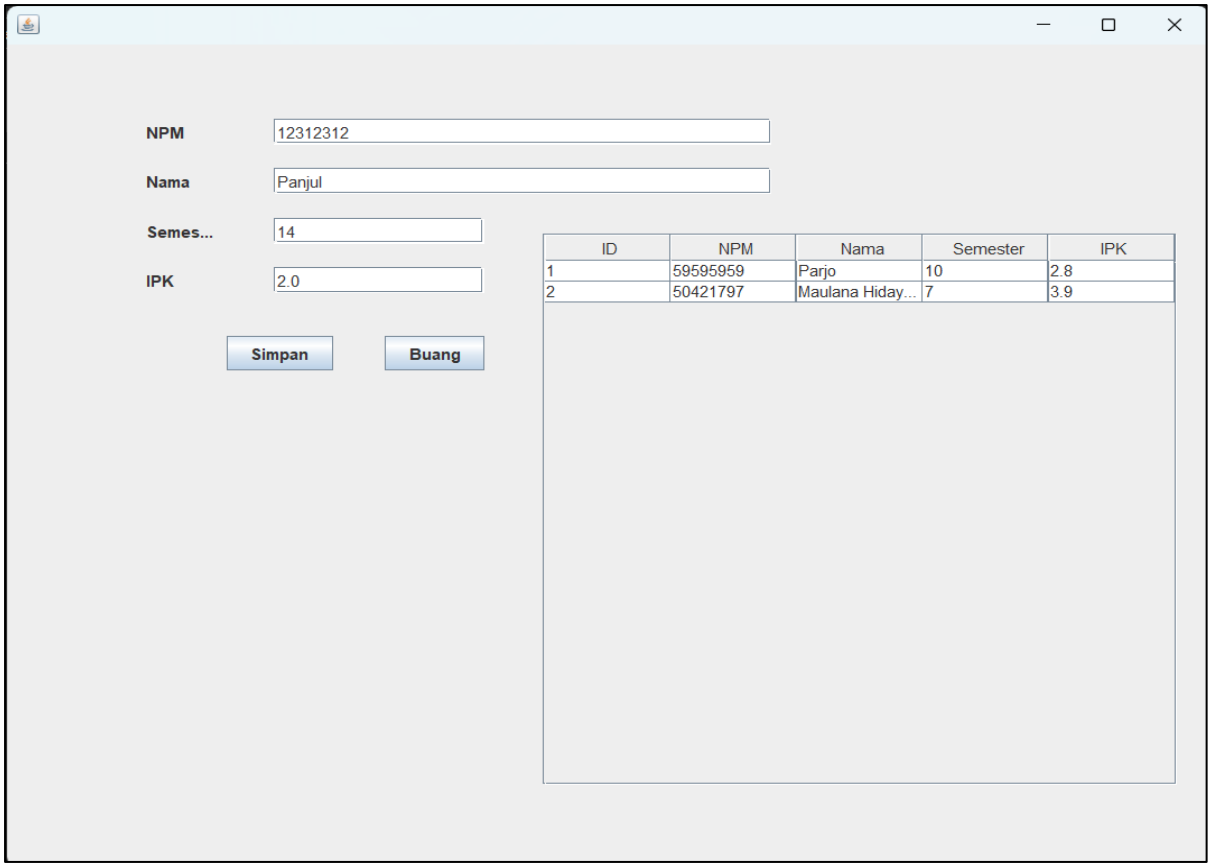
```java
    private javax.swing.JLabel jLabel1;
    private javax.swing.JLabel jLabel2;
    private javax.swing.JLabel jLabel3;
    private javax.swing.JLabel jLabel4;
    private javax.swing.JScrollPane jScrollPane1;
    private javax.swing.JTextField namaField;
    private javax.swing.JTextField npmField;
    private javax.swing.JTextField semesterField;
    private javax.swing.JButton simpanButton;
    // End of variables declaration
}
```

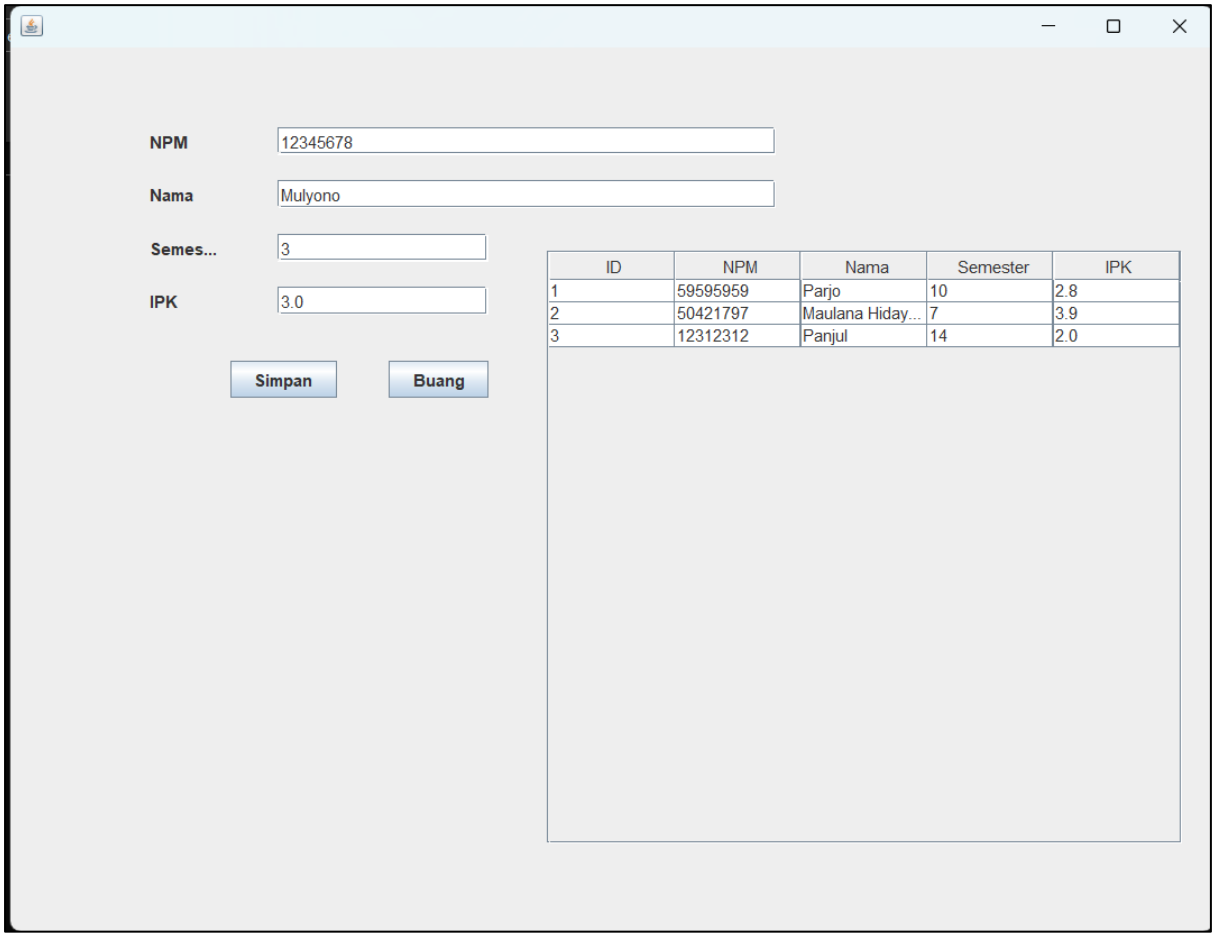
Output :

Tambah 5 Data Mahasiswa






NPM
12312312
Nama
Panjul
Semes...
14
IPK
2.0
Simpan
Buang
ID
NPM
Nama
Semester
IPK
1
59595959
Parjo
10
2.8
2
50421797
Maulana Hiday...
7
3.9


NPM
12345678
Nama
Mulyono
Semes...
3
IPK
3.0
Simpan
Buang
ID
NPM
Nama
Semester
IPK
1
59595959
Parjo
10
2.8
2
50421797
Maulana Hiday...
7
3.9
3
12312312
Panjul
14
2.0

NPM 89076555

Nama Naruto

Semes... 8

IPK 3.5

Simpan Buang

ID	NPM	Nama	Semester	IPK
1	59595959	Parjo	10	2.8
2	50421797	Maulana Hiday...	7	3.9
3	12312312	Panjul	14	2.0
4	12345678	Mulyono	3	3.0

NPM

Nama

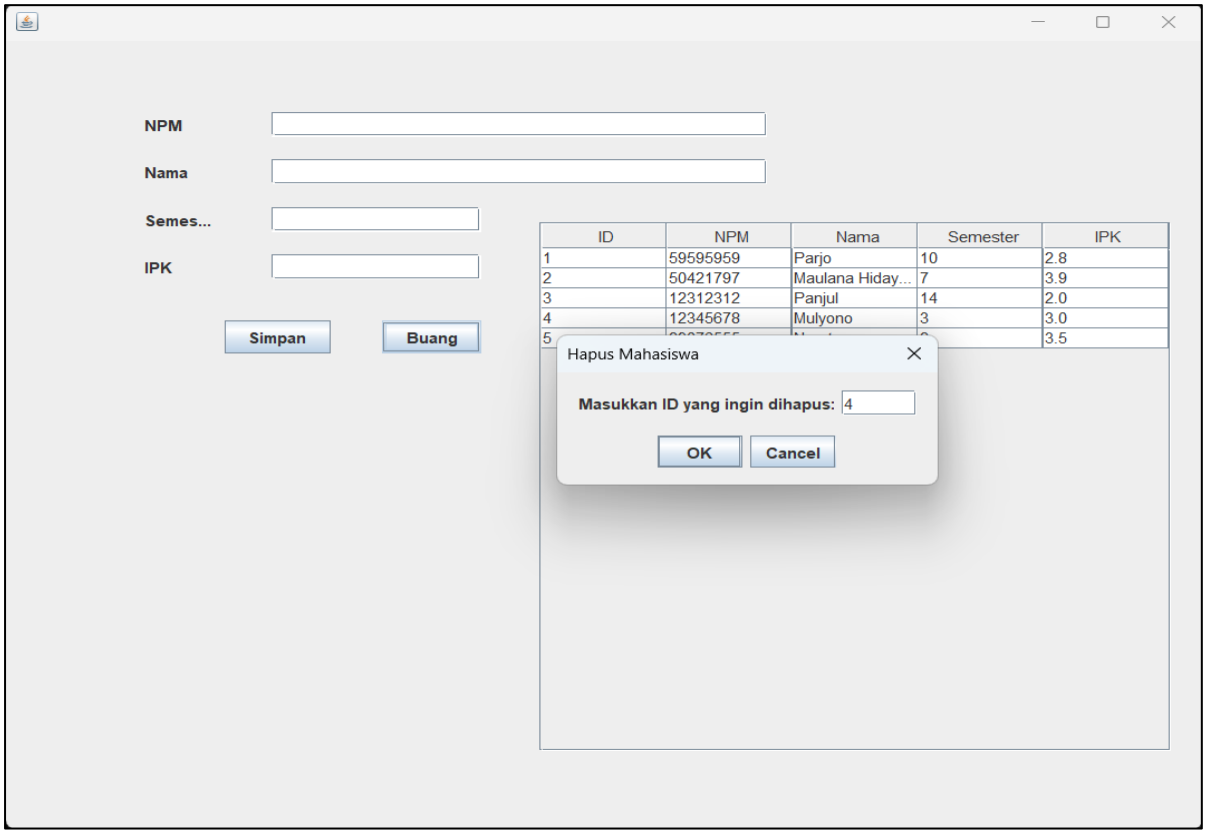
Semes...

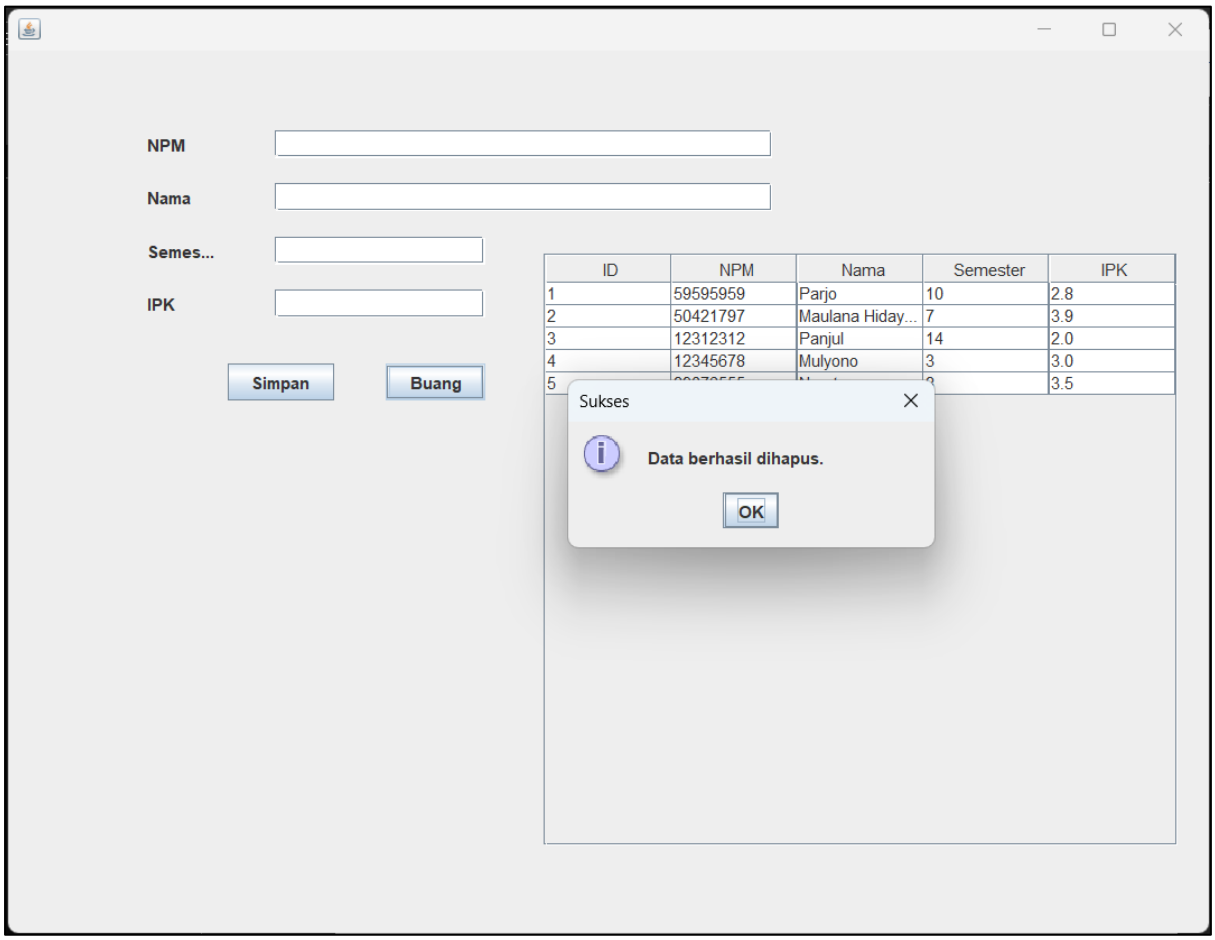
IPK

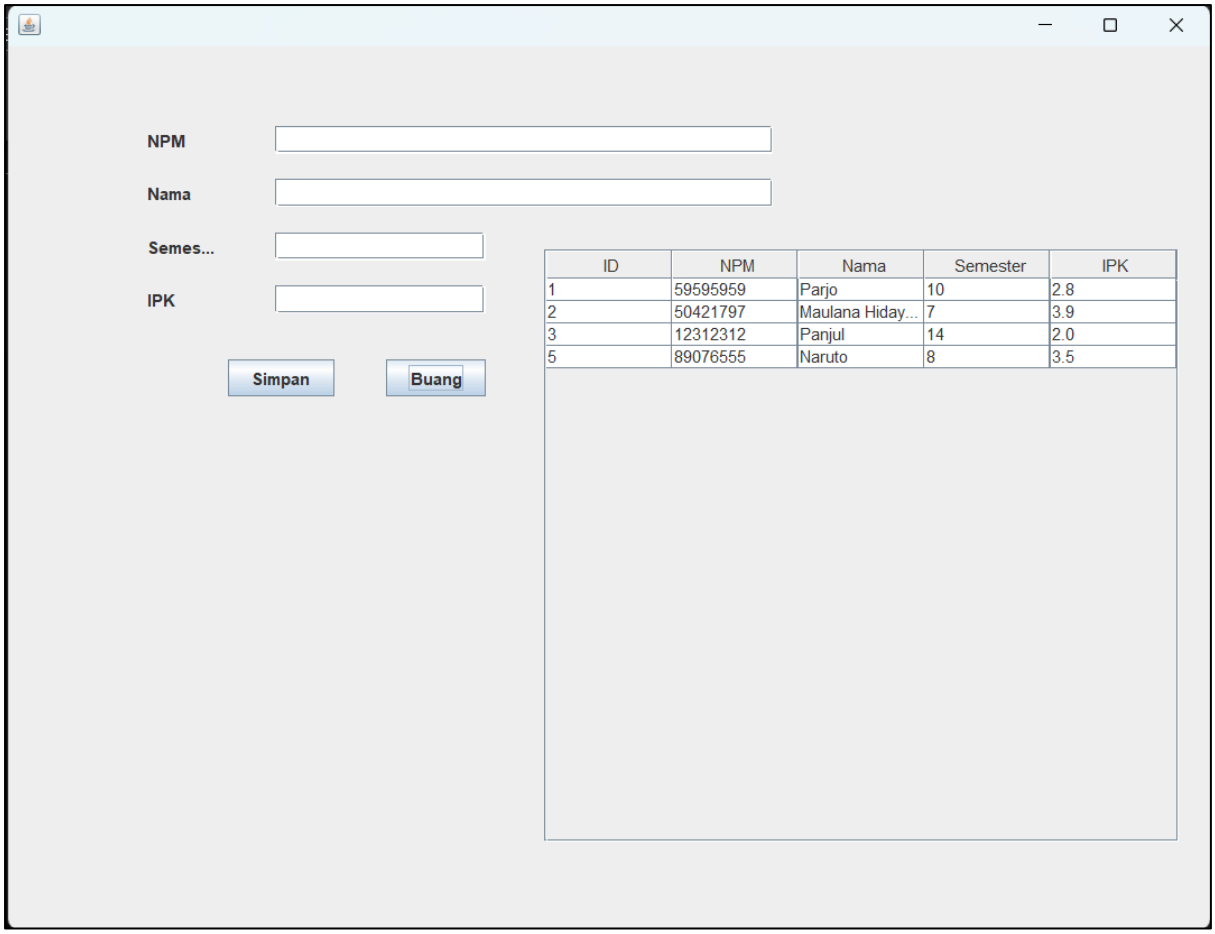
Simpan Buang

ID	NPM	Nama	Semester	IPK
1	59595959	Parjo	10	2.8
2	50421797	Maulana Hiday...	7	3.9
3	12312312	Panjul	14	2.0
4	12345678	Mulyono	3	3.0
5	89076555	Naruto	8	3.5

Buang Salah Satu Data Mahasiswa






NPM
Nama
Semes...
IPK
Simpan
Buang
ID
NPM
Nama
Semester
IPK
1
59595959
Parjo
10
2.8
2
50421797
Maulana Hiday...
7
3.9
3
12312312
Panjul
14
2.0
5
89076555
Naruto
8
3.5