

Nama:alghany sihombing

Kelas:4IA28

Npm:50421118

---

Act5

### 1. Pom.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/4.0.0
https://maven.apache.org/xsd/maven-4.0.0.xsd">
  <modelVersion>4.0.0</modelVersion>
  <parent>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-parent</artifactId>
    <version>3.3.5</version>
    <relativePath/> <!-- lookup parent from repository -->
  </parent>
  <groupId>com.mahasiswa</groupId>
  <artifactId>Pert5</artifactId>
  <version>0.0.1-SNAPSHOT</version>
  <name>Pert5</name>
  <description>Pert5</description>
  <url/>
  <licenses>
    <license/>
  </licenses>
  <developers>
    <developer/>
  </developers>
  <scm>
    <connection/>
    <developerConnection/>
    <tag/>
    <url/>
  </scm>
  <properties>
    <java.version>17</java.version>
  </properties>
  <dependencies>
    <dependency>
      <groupId>org.springframework.boot</groupId>
      <artifactId>spring-boot-starter</artifactId>
    </dependency>

    <dependency>
      <groupId>com.mysql</groupId>
      <artifactId>mysql-connector-j</artifactId>
      <scope>runtime</scope>
    </dependency>
    <dependency>
      <groupId>org.springframework.boot</groupId>
      <artifactId>spring-boot-starter-test</artifactId>
      <scope>test</scope>
    </dependency>
    <dependency>
      <groupId>org.springframework.boot</groupId>
      <artifactId>spring-boot-starter-data-jpa</artifactId>
    </dependency>
  </dependencies>

  <build>
```

```

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

```

## 2. MahasiswaController.java

```

package com.mahasiswa.pert5.controller;

import com.mahasiswa.pert5.Repository.MahasiswaRepository;
import com.mahasiswa.pert5.model.ModelMahasiswa;
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("Menambahkan Menu");
            System.out.println("1 - Tampilkan semua Mahasiswa");
            System.out.println("2 - Tambahkan mahasiswa baru");
            System.out.println("3 - cek koneksi database");
            System.out.println("4 - keluar");
            System.out.println("pilih opsi");
            opsi = scanner.nextInt();
            scanner.nextLine();
            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");
            }

        } while (opsi != 4);
    }

    private void cekKoneksi() {
        try {
            mahasiswaRepository.findAll();
        }
    }
}

```

```

        System.out.println("Koneksi ke database berhasil");
    } catch (Exception e) {
        System.out.println("Gagal terhubung ke database");
    }
}

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

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

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

```

### 3. ModelMahasiswa.java

```

package com.mahasiswa.pert5.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;

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

    public ModelMahasiswa(String nama, float ipk, int semester, String npm, int id) {
        this.nama = nama;
        this.ipk = ipk;
        this.semester = semester;
        this.npm = npm;
        this.id = id;
    }
}

```

#### 4. MahasiswaRepository.java

```

package com.mahasiswa.pert5.Repository;

import com.mahasiswa.pert5.model.ModelMahasiswa;
import org.springframework.data.jpa.repository.JpaRepository;
import org.springframework.stereotype.Repository;

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

```

## 5. Pert5Application.java

```
package com.mahasiswa.pert5;

import com.mahasiswa.pert5.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 Pert5Application implements CommandLineRunner {

    @Autowired
    private MahasiswaController mahasiswaController;

    public static void main(String[] args) {
        SpringApplication.run(Pert5Application.class, args);
    }

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

## 6. Application.properties

```
spring.application.name=Pert5
spring.datasource.url=jdbc:mysql://localhost:3306/selalu_gagal_db?useSSL=false&serverTime
zone=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

### I. Opsi 1

```
Run Pert5Application x
Console
2024-11-02T20:42:41.745+07:00 INFO 2900 --- [Pert5] [ main] com.mahasiswa.pert5.Pert5Application : Started Pert5Application in 4.758 seconds (process running for 5.755)
Menambahkan Menu
1 - Tampilkan semua Mahasiswa
2 - Tambahkan 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
ModelMahasiswa(id=1, npm='SD421118', nama='Ani', semester=4, ipk=3.0)
ModelMahasiswa(id=2, npm='SD111416', nama='andi', semester=5, ipk=4.0)
ModelMahasiswa(id=3, npm='SD421118', nama='alghany', semester=5, ipk=3.0)
ModelMahasiswa(id=4, npm='SD421010', nama='ani', semester=4, ipk=4.0)
Menambahkan Menu
1 - Tampilkan semua Mahasiswa
2 - Tambahkan mahasiswa baru
3 - cek koneksi database
4 - keluar
pilih opsi
1
```

### II. Opsi 2

```
Run Pert5Application x
Console
Insert into mahasiswa (ipk,nama,npm,semester) values (2,?,?,?)
Menambahkan Menu
1 - Tampilkan semua Mahasiswa
2 - Tambahkan mahasiswa baru
3 - cek koneksi database
4 - keluar
pilih opsi
2
Masukkan NPM:
SD431115
Masukkan Nama:
adi
Masukkan Semester:
6
Masukkan IPK:
4
Hibernate: insert into mahasiswa (ipk,nama,npm,semester) values (2,?,?,?)
Mahasiswa berhasil ditambahkan
Menambahkan Menu
1 - Tampilkan semua Mahasiswa
2 - Tambahkan mahasiswa baru
3 - cek koneksi database
4 - keluar
pilih opsi
1
```

### III. Opsi 3

```
Run Pert5Application x
Console
Masukkan Nama:
adi
Masukkan Semester:
6
Masukkan IPK:
4
Hibernate: insert into mahasiswa (ipk,nama,npn,semester) values (?,?,?,?)
Mahasiswa berhasil ditambahkan
Menambahkan Menu
1 - Tampilkan semua Mahasiswa
2 - Tambahkan mahasiswa baru
3 - cek koneksi database
4 - keluar
pilih opsi
3
Hibernate: select mm1_0.id,mm1_0.ipk,mm1_0.nama,mm1_0.npn,mm1_0.semester from mahasiswa mm1_0
Koneksi ke database berhasil
Menambahkan Menu
1 - Tampilkan semua Mahasiswa
2 - Tambahkan mahasiswa baru
3 - cek koneksi database
4 - keluar
pilih opsi
```

#### IV. Opsi 4

```
Koneksi ke database berhasil
Menambahkan Menu
1 - Tampilkan semua Mahasiswa
2 - Tambahkan mahasiswa baru
3 - cek koneksi database
4 - keluar
pilih opsi
4
keluar dari program
2024-11-02T20:48:46.255+07:00 INFO 2900 --- [Pert5] [ionShutdownHook] j.LocalContainerEntityManagerFactoryBean : Closing JPA EntityManagerFactory for persistence unit 'default'
2024-11-02T20:48:46.257+07:00 INFO 2900 --- [Pert5] [ionShutdownHook] com.zaxxer.hikari.HikariDataSource : HikariPool-1 - Shutdown initiated...
2024-11-02T20:48:46.266+07:00 INFO 2900 --- [Pert5] [ionShutdownHook] com.zaxxer.hikari.HikariDataSource : HikariPool-1 - Shutdown completed.
Process finished with exit code 0
```

#### V. Opsi default

```
Menambahkan Menu
1 - Tampilkan semua Mahasiswa
2 - Tambahkan mahasiswa baru
3 - cek koneksi database
4 - keluar
pilih opsi
6
opsi tidak valid
Menambahkan Menu
1 - Tampilkan semua Mahasiswa
2 - Tambahkan mahasiswa baru
3 - cek koneksi database
4 - keluar
pilih opsi
```

Pert6

Code

\*application properties

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

Delete.java

```
TextField idField = new TextField(5);

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

// Menampilkan dialog box dengan TextField, 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();
```



## MahasiswaApp.java

```
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.
    }
}
```

## MahasiswaCotroller.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();
    }
}

```

MahasiswaRepository

```
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>
{

}

}
```

#### MahasiswaService

```
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();
    }

```

#### Mahasiswa View

```

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

    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);
    }

```

#### ModelTabelMahasiswa.java

```

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
}

```

```
}  
}
```

Pom.xml

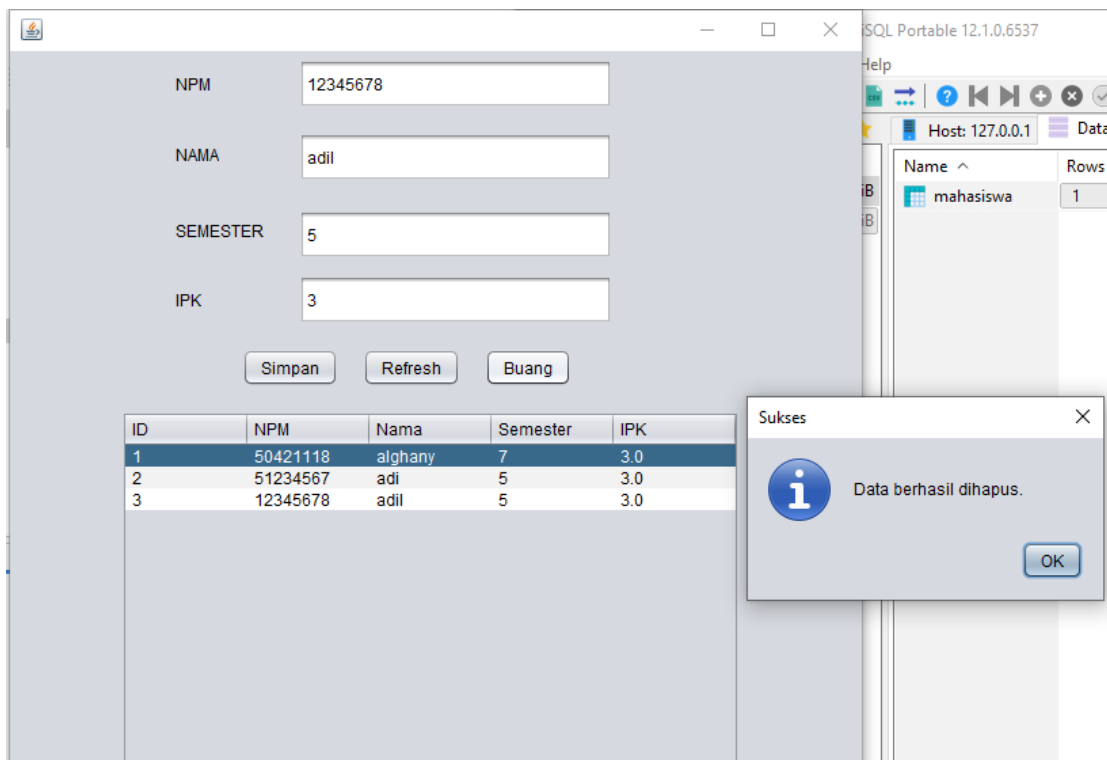
```
<parent>  
  <groupId>org.springframework.boot</groupId>  
  <artifactId>spring-boot-starter-parent</artifactId>  
  <version>3.3.3</version>  
  <relativePath/>  
</parent>  
  
<dependencies>  
  <!-- Hibernate + Spring Data JPA -->  
  <dependency>  
    <groupId>org.springframework.boot</groupId>  
    <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>  
    <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>
```

Simpan.java

```
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();
```

## Output



The screenshot shows a Java Swing application window titled "Simpan". It contains a form with four input fields: "NPM" (value: 12345678), "NAMA" (value: adil), "SEMESTER" (value: 5), and "IPK" (value: 3). Below the form are three buttons: "Simpan", "Refresh", and "Buang". At the bottom of the window is a table with the following data:

ID	NPM	Nama	Semester	IPK
1	50421118	alghany	7	3.0
2	51234567	adi	5	3.0
3	12345678	adil	5	3.0

A "Sukses" dialog box is open in the foreground, displaying an information icon and the message "Data berhasil dihapus." (Data successfully deleted). The dialog has an "OK" button.

NPM

12345678

NAMA

adil

SEMESTER

5

IPK

3

Simpan

Refresh

Buang

ID	NPM	Nama	Semester	IPK
2	51234567	adi	5	3.0
3	12345678	adil	5	3.0