

Nama : DIKA DWIPATI

NPM : 50421378

Kelas : 4IA28

Source code

Spring boot

```
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 ASUS
 */
@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();
    }
}
```

Mahasiswa controller

```
/*
 * 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;
/**
 *
 * @author DIKA DWIPATI
 */
@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.");
    }
}
}

```

Model mahasiswa

```

/*
 * 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 DIKA DWIPATI
 */
@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 +
        ", nama='" + npm + '\'' +
        ", nama='" + nama + '\'' +
        ", nama='" + semester + '\'' +
        ", jurusan='" + ipk + '\'' +
        '}';
}
}

```

Mahasiswa repository

```

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> {  
}
```

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>
```

Output Pert 5

```
Menu:
1. Tampilkan semua mahasiswa
2. Tambah mahasiswa baru
3. Cek koneksi database
4. Keluar
Pilih opsi: 2
Masukkan NPM : 50421378
Masukkan Nama : dika
Masukkan Semester : 7
Masukkan IPK : 3
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 : 50421222
Masukkan Nama : rey
Masukkan Semester : 4
Masukkan IPK : 3
Hibernate: insert into mahasiswa (ipk,nama,npm,semester) values (?,?,?,?)
Mahasiswa berhasil ditambahkan.
```

10 - Run (Performance Spring Boot Application)

```
Menu:  
1. Tampilkan semua mahasiswa  
2. Tambah mahasiswa baru  
3. Cek koneksi database  
4. Keluar  
Pilih opsi: 2  
Masukkan NPM : 58642223  
Masukkan Nama : yopi  
Masukkan Semester : 8  
Masukkan IPK : 3  
Hibernate: insert into mahasiswa (ipk,nama,npn,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 mm1_0.id,mm1_0.ipk,mm1_0.nama,mm1_0.npm,mm1_0.semester from mahasiswa mm1_0  
Mahasiswa{id=1, nama='50421378', nama='dika', nama='6', jurusan='3.0'}  
Mahasiswa{id=2, nama='50421378', nama='dika', nama='7', jurusan='3.0'}  
Mahasiswa{id=3, nama='50421222', nama='rey', nama='4', jurusan='3.0'}  
Mahasiswa{id=4, nama='58642223', nama='yopi', nama='8', jurusan='3.0'}  
  
Menu:
```

Pert 6

Source code

Mahasiswa app

```
/*
@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.
    }
}
```

Mahasiswa 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";
    }
}
```

Model mahasiswa

```
@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;  
}
```

Model table mahasiswa

```
package com.manasiswa.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();
        }
    }
}
```

Mahasiswa repository

```
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> {
    public Object findById(int id);

    public void deleteById(int id);
}
```

Mahasiswa service

```
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;

/**
 *
 * @author DIKA DWIPATI
 */
@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);
    }
}
```

Mahasiswa view

```
/***
 *
 * @author DIKA DWIPATI
 */
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
    }
    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);
    }
}
```

Output

The screenshot shows a Windows application window titled "Output". The interface includes several input fields and two buttons:

- NPM:** An input field followed by a blue "Simpan" button.
- Nama:** An input field followed by a blue "Buang" button.
- Semester:** An input field.
- IPK:** An input field.

Below these fields is a table with five columns:

ID	NPM	Nama	Semester	IPK

Tambah data

NPM

Nama

Semester

IPK

ID	NPM	Nama	Semester	IPK
1	50421378	dika	7	3.0
2	5224455	rey	4	3.0
3	5224477	koy	4	3.0
4	5224499	oitt	4	3.0
5	55224499	auuuu	4	3.0

Buang data

NPM	<input type="text" value="55224499"/>	<input type="button" value="Simpan"/>
Nama	<input type="text" value="auuuu"/>	<input type="button" value="Buang"/>
Semester	<input type="text" value="4"/>	
IPK	<input type="text" value="3"/>	

ID	NPM	Nama	Semester	IPK
1	50421378	dika	7	3.0
3	5224477	koy	4	3.0
4	5224499	oitt	4	3.0
5	55224499	auuuu	4	3.0