Nama : Muhammad Sahrul Farhan

Kelas : 4IA28

NPM : 51421076

ACTIVITY 5

Listing Program

Pertemuan5SpringBootApplication.java

```java
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 Administrator
 */
@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();
    }

}
```

MahasiswaRepository.java

```java
package com.mahasiswa.repository;

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

/**
 *
 * @author Administrator
 */
@Repository
public interface MahasiswaRepository extends JpaRepository<ModelMahasiswa, Long> {

}
```

MahasiswaController.java

```java
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 Administrator
 */
@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));
        }
    }
```

```
63
64      private void tambahMahasiswa(Scanner scanner) {
65          System.out.print("Masukkan NPM : ");
66          String npm = scanner.nextLine();
67          System.out.print("Masukkan Nama : ");
68          String nama = scanner.nextLine();
69          System.out.print("Masukkan Semester : ");
70          int semester = scanner.nextInt();
71          System.out.print("Masukkan IPK : ");
72          float ipk = scanner.nextFloat();
73
74          ModelMahasiswa mahasiswa = new ModelMahasiswa(0, npm, nama, semester, ipk);
75          mahasiswaRepository.save(mahasiswa);
76          System.out.println("Mahasiswa berhasil ditambahkan.");
77      }
78
79      private void cekKoneksi() {
80          try {
81              mahasiswaRepository.findAll();
82              System.out.println("Koneksi ke database berhasil.");
83          } catch (Exception e) {
84              System.out.println("Gagal terhubung ke database.");
85          }
86      }
87  }
```

ModelMahasiswa.java

```
1   package com.mahasiswa.model;
2
3   import jakarta.persistence.*;
4
5   /**
6    *
7    * @author aditrhamid
8    */
9   @Entity
10  @Table(name = "mahasiswa")
11
12  public class ModelMahasiswa {
13      @Id
14      @GeneratedValue(strategy = GenerationType.IDENTITY)
15
16      @Column(name = "id")
17      private int id;
18
19      @Column(name = "semester")
20      private int semester;
21
22      @Column(name = "npm", nullable = false, length = 8)
23      private String npm;
24
25      @Column(name = "nama", nullable = false, length = 50)
26      private String nama;
27
28      @Column(name = "ipk")
29      private float ipk;
30
31      public ModelMahasiswa() {
```

```
32
33      }
34
35      public ModelMahasiswa(int id, String npm, String nama, int semester, float ipk) {
36          this.id = id;
37          this.npm = npm;
38          this.nama = nama;
39          this.semester = semester;
40          this.ipk = ipk;
41      }
42
43      /**
44       * @return the id
45       */
46      public int getId() {
47          return id;
48      }
49
50      /**
51       * @param id the id to set
52       */
53      public void setId(int id) {
54          this.id = id;
55      }
56
57      /**
58       * @return the semester
59       */
60      public int getSemester() {
61          return semester;
62      }
```

```
63
64      /**
65       * @param semester the semester to set
66       */
67      public void setSemester(int semester) {
68          this.semester = semester;
69      }
70
71      /**
72       * @return the npm
73       */
74      public String getNpm() {
75          return npm;
76      }
77
78      /**
79       * @param npm the npm to set
80       */
81      public void setNpm(String npm) {
82          this.npm = npm;
83      }
84
85      /**
86       * @return the nama
87       */
88      public String getNama() {
89          return nama;
90      }
91
92      /**
93       * @param nama the nama to set
```

Output

Menampilkan data yang telah ditambahkan

```
Mahasiswa berhasil ditambahkan.

Menu:
1. Tampilkan semua mahasiswa
2. Tambah mahasiswa baru
3. Cek koneksi database
4. Keluar
Pilih opsi: 1
Hibernate: select mml_0.id,mml_0.ipk,mml_0.nama,mml_0.npm,mml_0.semester from mahasiswa mml_0
Mahasiswa{id=1, nama='51421076', nama='Muhammad', nama='7', jurusan='3.0'}
Mahasiswa{id=2, nama='51421077', nama='Sahrul', nama='6', jurusan='3.0'}
Mahasiswa{id=3, nama='51421078', nama='Farhan', nama='5', jurusan='3.0'}
```

Cek Koneksi database

```
Menu:
1. Tampilkan semua mahasiswa
2. Tambah mahasiswa baru
3. Cek koneksi database
4. Keluar
Pilih opsi: 3
Hibernate: select mml_0.id,mml_0.ipk,mml_0.nama,mml_0.npm,mml_0.semester from mahasiswa mml_0
Koneksi ke database berhasil.
```

Nama : Muhammad Sahrul Farhan

Kelas : 4IA28

NPM : 51421076

ACTIVITY 6

Listing Program

MahasiswaApp.java

```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;

/**
 *
 * @author Administrator
 */
@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

```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();
    }
}
```

ModelTabelMahasiswa.java

```java
package com.mahasiswa.model;
import javax.swing.table.AbstractTableModel;
import java.util.List;

/**
 *
 * @author Administrator
 */
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
    }
}
```

ModelMahasiswa.java

```java
package com.mahasiswa.model;

import jakarta.persistence.*;

/**
 *
 * @author aditrhamid
 */
@Entity
@Table(name = "mahasiswa")

public class ModelMahasiswa {
    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)

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

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

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

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

    @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;
    }

    /**
     * @return the id
     */
    public int getId() {
        return id;
    }

    /**
     * @param id the id to set
     */
    public void setId(int id) {
        this.id = id;
    }

    /**
     * @return the semester
     */
    public int getSemester() {
        return semester;
    }
```

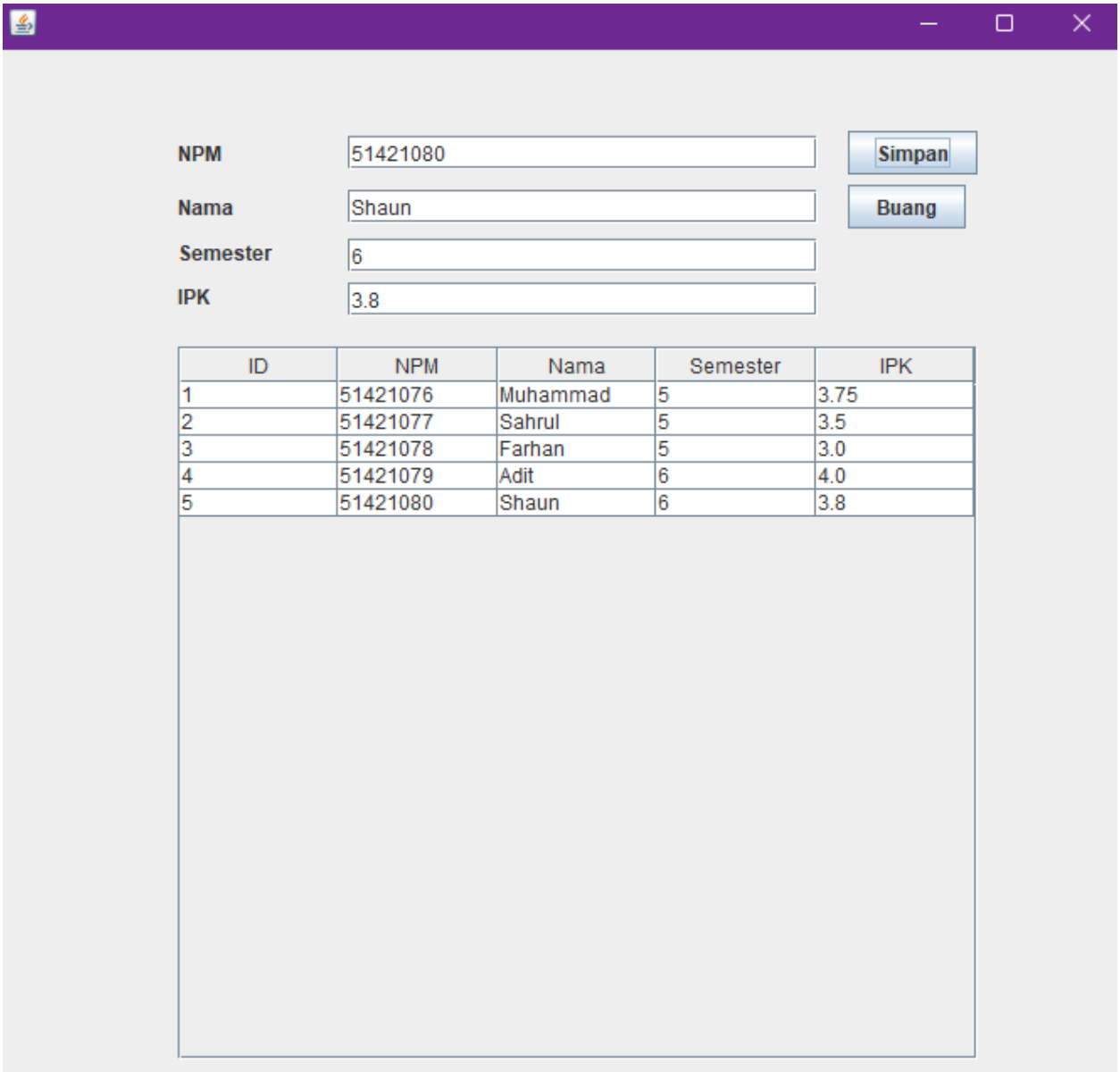
```
63
64      /**
65       * @param semester the semester to set
66       */
67      public void setSemester(int semester) {
68          this.semester = semester;
69      }
70
71      /**
72       * @return the npm
73       */
74      public String getNpm() {
75          return npm;
76      }
77
78      /**
79       * @param npm the npm to set
80       */
81      public void setNpm(String npm) {
82          this.npm = npm;
83      }
84
85      /**
86       * @return the nama
87       */
88      public String getNama() {
89          return nama;
90      }
91
92      /**
93       * @param nama the nama to set
94       */
95      public void setNama(String nama) {
96          this.nama = nama;
97      }
98
99      /**
100      * @return the ipk
101      */
102     public float getIpk() {
103         return ipk;
104     }
105
106     /**
107      * @param ipk the ipk to set
108      */
109     public void setIpk(float ipk) {
110         this.ipk = ipk;
111     }
112
113 }
114
```

MahasiswaService.java

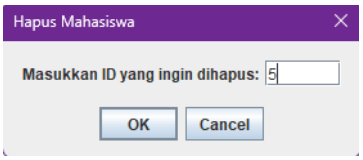
```java
package com.mahasiswa.service;

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;

/**
 *
 * @author Administrator
 */
@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();
    }
}
```

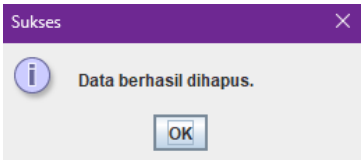
Output

Database dengan data yang ditampilkan



Opsi ketika menghapus data





Tampilan ketika ada data yang telah dihapus