

## PERTEMUAN

Nama : Teuku Ardhi Firmansyah Al Ghozali

NPM : 51421472

Kelas : 4IA28

### PERTEMUAN 5

---

#### OUTPUT TAMBAH DATA

Menu:

1. Tampilkan semua mahasiswa
2. Tambah mahasiswa baru
3. Cek koneksi database
4. Keluar

Pilih opsi: 2

Masukkan NPM : 51421472

Masukkan Nama : Teuku

Masukkan Semester : 7

Masukkan IPK : 4

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 : 51421231

Masukkan Nama : Raihan

Masukkan Semester : 9

Masukkan IPK : 4

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 : 5121123

Masukkan Nama : Adit

Masukkan Semester : 7

Masukkan IPK : 4

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 mm1\_0.id,mm1\_0.ipk,mm1\_0.nama,mm1\_0.npm,mm1\_0.semester from mahasiswa mm1\_0

Mahasiswa{id=1, nama='Teuku', nama='51421472', nama='7', jurusan='4.0'}

Mahasiswa{id=2, nama='Raihan', nama='51421231', nama='9', jurusan='4.0'}

Mahasiswa{id=3, nama='Adit', nama='5121123', nama='7', jurusan='4.0'}

## Pertemuan5SpringBootApplication

```
1 package com.mahasiswa;
2 import com.mahasiswa.controller.MahasiswaController;
3 import org.springframework.beans.factory.annotation.Autowired;
4 import org.springframework.boot.CommandLineRunner;
5 import org.springframework.boot.SpringApplication;
6 import org.springframework.boot.autoconfigure.SpringBootApplication;
7
8 /**
9  *
10  * @author ASUS
11  */
12 @SpringBootApplication
13 public class Pertemuan5SpringBootApplication implements CommandLineRunner{
14
15     @Autowired
16     private MahasiswaController mhsController;
17     public static void main(String[] args) {
18         SpringApplication.run(Pertemuan5SpringBootApplication.class, args);
19     }
20
21     @Override
22     public void run(String... args) throws Exception {
23         mhsController.tampilkanMenu();
24     }
25
26 }
```

## MahasiswaController

```
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;
                default:
                    System.out.println("Ops! 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

```

  /**
   * 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;
  import jakarta.persistence.*;

  /**
   *
   * @author Teuku
   */
  @Entity
  @Table(name = "mahasiswa")
  public class ModelMahasiswa {
      @Id
      @GeneratedValue(strategy = GenerationType.IDENTITY)
      @Column(name = "id")
      private int id;

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

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

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

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

      public ModelMahasiswa() {
      }
  }

```



```

    }

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

```

## PERTEMUAN

Nama : Teuku Ardhi Firmansyah Al Ghozali

NPM : 51421472

Kelas : 4IA28

## PERTEMUAN 6

---

### OUTPUT 3 DATA

NPM

51421213

Nama

Raihan

Sem...

7

IPK

3.8

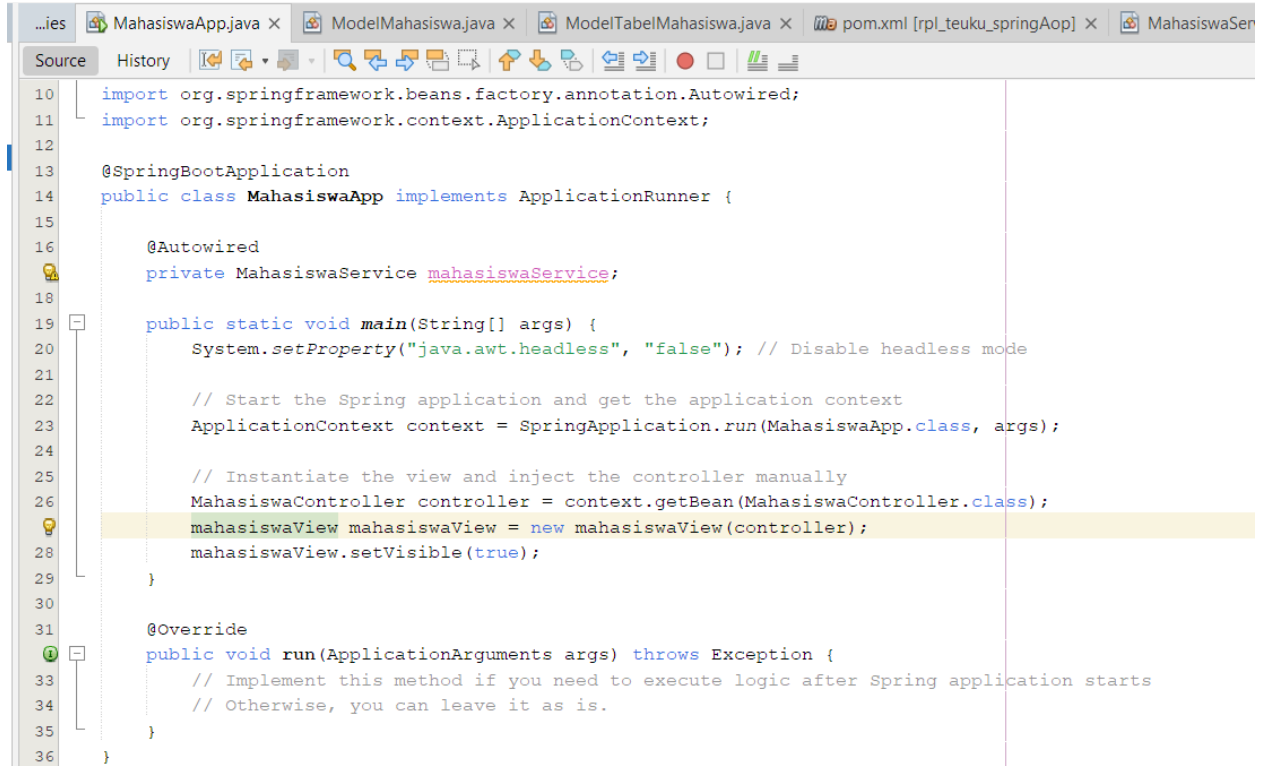
Simpan

Buang

ID	NPM	Nama	Semester	IPK
1	Teuku	51421472	7	4.0
2	Alif	51421412	7	3.9
3	Raihan	51421213	7	3.8

## CODE

### MahasiswaApp



```
10 import org.springframework.beans.factory.annotation.Autowired;
11 import org.springframework.context.ApplicationContext;
12
13 @SpringBootApplication
14 public class MahasiswaApp implements ApplicationRunner {
15
16     @Autowired
17     private MahasiswaService mahasiswaService;
18
19     public static void main(String[] args) {
20         System.setProperty("java.awt.headless", "false"); // Disable headless mode
21
22         // Start the Spring application and get the application context
23         ApplicationContext context = SpringApplication.run(MahasiswaApp.class, args);
24
25         // Instantiate the view and inject the controller manually
26         MahasiswaController controller = context.getBean(MahasiswaController.class);
27         mahasiswaView mahasiswaView = new mahasiswaView(controller);
28         mahasiswaView.setVisible(true);
29     }
30
31     @Override
32     public void run(ApplicationArguments args) throws Exception {
33         // Implement this method if you need to execute logic after Spring application starts
34         // Otherwise, you can leave it as is.
35     }
36 }
```

k,mm1\_0.nama,mm1\_0.npm,mm1\_0.semester from mahasiswa mm1\_0

### MahasiswaController



```
...va ModelTabelMahasiswa.java X pom.xml [rpl_teuku_springAop] X MahasiswaService.java X MahasiswaRe

Source History
1 package com.mahasiswa.controller;
2
3 import org.springframework.beans.factory.annotation.Autowired;
4 import org.springframework.web.bind.annotation.*;
5 import com.mahasiswa.model.ModelMahasiswa;
6 import com.mahasiswa.service.MahasiswaService;
7
8 import java.util.List;
9 import org.springframework.stereotype.Controller;
10
11
12 @Controller
13 public class MahasiswaController {
14
15     @Autowired
16     private MahasiswaService mahasiswaService;
17
18     // Add new Mahasiswa
19     public String addMahasiswa(@RequestBody ModelMahasiswa mhs) {
20         mahasiswaService.addMhs(mhs);
21         return "Mahasiswa added successfully";
22     }
23
24     // Get Mahasiswa by ID
25     public ModelMahasiswa getMahasiswa(@PathVariable int id) {
26         return mahasiswaService.getMhs(id);
27     }
28
29     // Update Mahasiswa
30     public String updateMahasiswa(@RequestBody ModelMahasiswa mhs) {
31         mahasiswaService.updateMhs(mhs);
32         return "Mahasiswa updated successfully";
33     }
34
35     // Delete Mahasiswa by ID
36     public String deleteMahasiswa(@PathVariable int id) {
```

## ModelMahasiswa

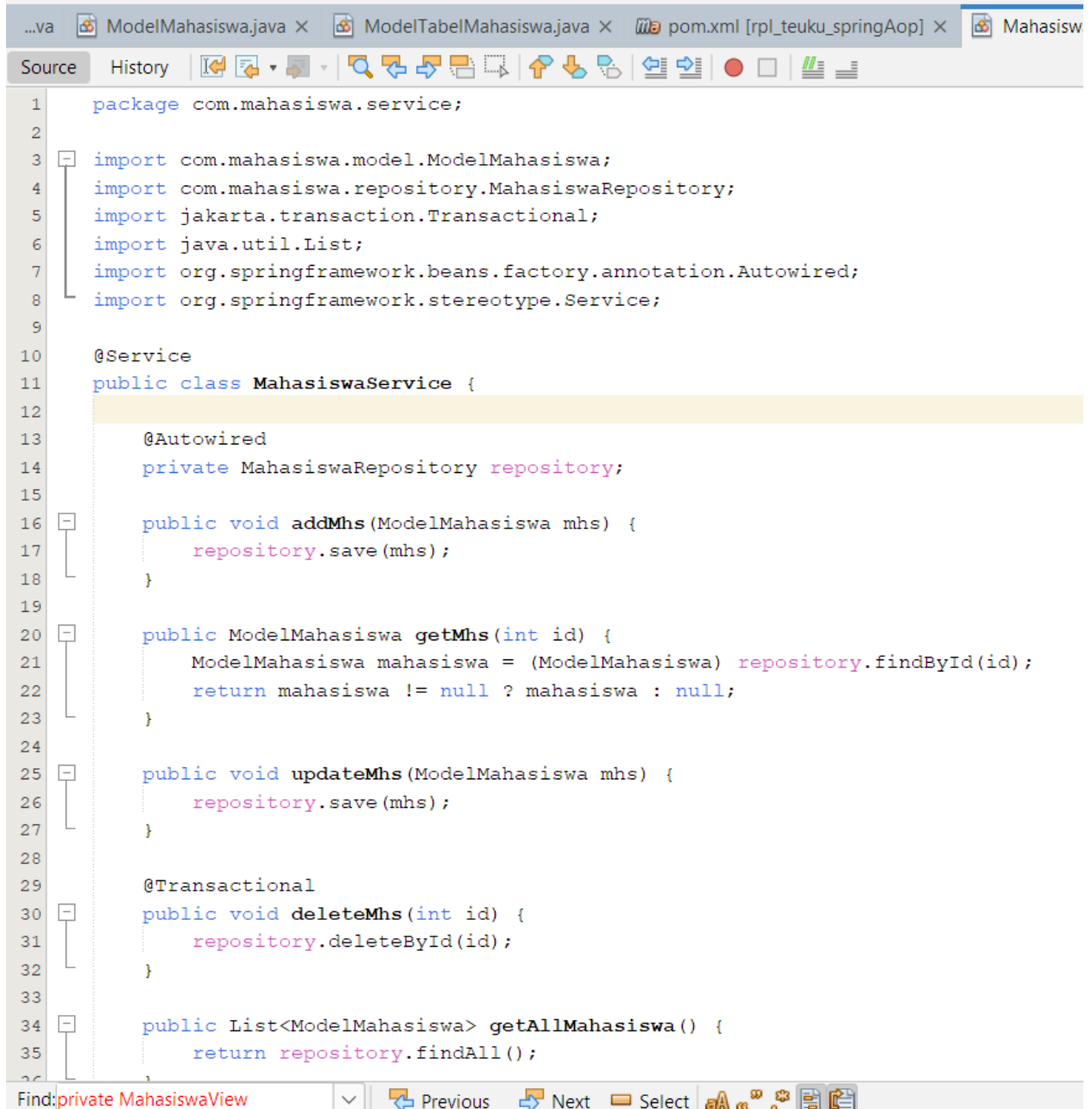
```
1  /*
2  * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
3  * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this template
4  */
5  package com.mahasiswa.model;
6
7  import jakarta.persistence.Entity;
8  import jakarta.persistence.*;
9
10 /**
11  *
12  * @author Teuku
13  */
14
15 @Entity
16 @Table(name = "mahasiswa")
17 public class ModelMahasiswa {
18     @Id
19     @GeneratedValue(strategy = GenerationType.IDENTITY)
20     @Column(name = "id")
21     private int id;
22
23     @Column(name = "nama", nullable = false, length = 50)
24     private String nama;
25
26     @Column(name = "npm", nullable = false, length = 8)
27     private String npm;
28
29     @Column(name = "semester")
30     private int semester;
31
32     @Column(name = "ipk")
33     private float ipk;
34
35     public ModelMahasiswa() {
36
37     }
38 }
```

## ModelTabelMahasiswa

```
1  package com.mahasiswa.model;
2  import javax.swing.table.AbstractTableModel;
3  import java.util.List;
4
5  public class ModelTabelMahasiswa extends AbstractTableModel {
6      private List<ModelMahasiswa> mahasiswaList;
7      private String[] columnNames = {"ID", "NPM", "Nama", "Semester", "IPK"};
8
9      public ModelTabelMahasiswa(List<ModelMahasiswa> mahasiswaList) {
10         this.mahasiswaList = mahasiswaList;
11     }
12
13     @Override
14     public int getRowCount() {
15         return mahasiswaList.size(); // Jumlah baris sesuai dengan jumlah data mahasiswa
16     }
17
18     @Override
19     public int getColumnCount() {
20         return columnNames.length; // Jumlah kolom sesuai dengan jumlah elemen dalam columnNames
21     }
22
23     @Override
24     public Object getValueAt(int rowIndex, int columnIndex) {
25         ModelMahasiswa mahasiswa = mahasiswaList.get(rowIndex);
26         switch (columnIndex) {
27             case 0:
28                 return mahasiswa.getId();
29             case 1:
30                 return mahasiswa.getNpm();
31             case 2:
32                 return mahasiswa.getNama();
33             case 3:
34                 return mahasiswa.getSemester();
35             case 4:
36                 return mahasiswa.getIpk();
37         }
38     }
39 }
```

find: private MahasiswaView

## MahasiswaService



```
1 package com.mahasiswa.service;
2
3 import com.mahasiswa.model.ModelMahasiswa;
4 import com.mahasiswa.repository.MahasiswaRepository;
5 import jakarta.transaction.Transactional;
6 import java.util.List;
7 import org.springframework.beans.factory.annotation.Autowired;
8 import org.springframework.stereotype.Service;
9
10 @Service
11 public class MahasiswaService {
12
13     @Autowired
14     private MahasiswaRepository repository;
15
16     public void addMhs(ModelMahasiswa mhs) {
17         repository.save(mhs);
18     }
19
20     public ModelMahasiswa getMhs(int id) {
21         ModelMahasiswa mahasiswa = (ModelMahasiswa) repository.findById(id);
22         return mahasiswa != null ? mahasiswa : null;
23     }
24
25     public void updateMhs(ModelMahasiswa mhs) {
26         repository.save(mhs);
27     }
28
29     @Transactional
30     public void deleteMhs(int id) {
31         repository.deleteById(id);
32     }
33
34     public List<ModelMahasiswa> getAllMahasiswa() {
35         return repository.findAll();
36     }
37 }
```

Find: private MahasiswaView

## MahasiswaView



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

The screenshot displays a form design interface. On the left, a vertical panel shows a tree structure with a root node selected. A dashed line connects this root node to a form layout in the center. The form layout consists of a table with four columns labeled "Title 1", "Title 2", "Title 3", and "Title 4". Below the table, there is a large empty rectangular area. To the right of the table, there is a vertical gray bar with a wavy pattern. The form layout includes four input fields labeled "NPM", "Nama", "Seme...", and "IPK". To the right of these fields are two buttons labeled "Simpan" and "Buang". The "Buang" button is highlighted with a yellow border. The entire form is set against a light gray background.