

## ACTIVITY 5

Nama : Alif Fathur Rahman

Kelas : 4IA28

Npm : 51421577

---

### Pertemuan5SpringBoot.java

```
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;
6  import com.mahasiswa.controller.MahasiswaController;
7  import org.springframework.beans.factory.annotation.Autowired;
8  import org.springframework.boot.CommandLineRunner;
9  import org.springframework.boot.SpringApplication;
10 import org.springframework.boot.autoconfigure.SpringBootApplication;
11
12 /**
13 *
14 * @author ASUS
15 */
16 @SpringBootApplication
17 public class Pertemuan5SpringBootApplication implements CommandLineRunner{
18
19     @Autowired
20     private MahasiswaController mhsController;
21     public static void main(String[] args) {
22         SpringApplication.run(Pertemuan5SpringBootApplication.class, args);
23     }
24
25     @Override
26     public void run(String... args) throws Exception {
27         mhsController.tampilkanMenu();
28     }
29
30 }
```

## MahasiswaController.java

```
1 package com.mahasiswa.controller;
2
3 import com.mahasiswa.model.ModelMahasiswa;
4 import com.mahasiswa.repository.MahasiswaRepository;
5 import org.springframework.beans.factory.annotation.Autowired;
6 import org.springframework.stereotype.Controller;
7
8 import java.util.List;
9 import java.util.Scanner;
10
11 @Controller
12 public class MahasiswaController {
13
14     @Autowired
15     private MahasiswaRepository mahasiswaRepository;
16
17     public void tampilkanMenu() {
18         Scanner scanner = new Scanner(System.in);
19         int opsi;
20
21         do {
22             System.out.println("\nMenu:");
23             System.out.println("1. Tampilkan semua mahasiswa");
24             System.out.println("2. Tambah mahasiswa baru");
25             System.out.println("3. Cek koneksi database");
26             System.out.println("4. Keluar");
27             System.out.print("Pilih opsi: ");
28             opsi = scanner.nextInt();
29             scanner.nextLine(); // menangkap newline
30
31             switch (opsi) {
32                 case 1:
33                     tampilkanSemuaMahasiswa();
34                     break;
35                 case 2:
36                     tambahMahasiswa(scanner);
37                     break;
38                 case 3:
39                     cekKoneksi();
40                     break;
41                 case 4:
42                     System.out.println("Keluar dari program.");
43                     break;
44                 default:
45                     System.out.println("Ops! tidak valid, coba lagi.");
46             }
47         } while (opsi != 4);
48     }
49
50     private void tampilkanSemuaMahasiswa() {
51         List<ModelMahasiswa> mahasiswaList = mahasiswaRepository.findAll();
52         if (mahasiswaList.isEmpty()) {
53             System.out.println("Tidak ada data mahasiswa.");
54         } else {
55             mahasiswaList.forEach(mahasiswa -> System.out.println(mahasiswa));
56         }
57     }
58
59     private void tambahMahasiswa(Scanner scanner) {
60         System.out.print("Masukkan NPM : ");
61         String npm = scanner.nextLine();
62         System.out.print("Masukkan Nama : ");
63         String nama = scanner.nextLine();
64     }
65 }
```

```

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

```

## ModelMahasiswa.java

```

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.*;
8
9   /**
10    *
11    * @author alif
12    */
13   @Entity
14   @Table(name = "mahasiswa")
15   public class ModelMahasiswa {
16       @Id
17       @GeneratedValue(strategy = GenerationType.IDENTITY)
18       @Column(name = "id")
19       private int id;
20
21       @Column(name = "npm", nullable = false, length = 8)
22       private String npm;
23
24       @Column(name = "nama", nullable = false, length = 50)
25       private String nama;
26
27       @Column(name = "semester")
28       private int semester;
29
30       @Column(name = "ipk")
31       private float ipk;
32
33       public ModelMahasiswa() {

```

```

34 |
35 |     }
36 |
37 |     public ModelMahasiswa(int id, String npm, String nama, int semester, float ipk){
38 |         this.id = id;
39 |         this.npm = npm;
40 |         this.nama = nama;
41 |         this.semester = semester;
42 |         this.ipk = ipk;
43 |     }
44 |
45 |     public int getId() {
46 |         return id;
47 |     }
48 |
49 |     public void setId(int id) {
50 |         this.id = id;
51 |     }
52 |
53 |     public String getNpm() {
54 |         return npm;
55 |     }
56 |
57 |     public void setNpm(String npm) {
58 |         this.npm = npm;
59 |     }
60 |
61 |     public String getNama() {
62 |         return nama;
63 |     }
64 |
65 |     public void setNama(String nama) {
66 |         this.nama = nama;

```

```

67 |     }
68 |
69 |     public int getSemester() {
70 |         return semester;
71 |     }
72 |
73 |     public void setSemester(int semester) {
74 |         this.semester = semester;
75 |     }
76 |
77 |     public float getIpk() {
78 |         return ipk;
79 |     }
80 |
81 |     public void setIpk(float ipk) {
82 |         this.ipk = ipk;
83 |     }
84 |
85 |     @Override
86 |     public String toString() {
87 |         return "Mahasiswa{" +
88 |             "id=" + id +
89 |             ", nama='" + npm + '\'' +
90 |             ", nama='" + nama + '\'' +
91 |             ", nama='" + semester + '\'' +
92 |             ", jurusan='" + ipk + '\'' +
93 |             '}';
94 |     }
95 |
96 | }

```

## MahasiswaRepository.java

```
1 package com.mahasiswa.repository;
2
3 import com.mahasiswa.model.ModelMahasiswa;
4 import org.springframework.data.jpa.repository.JpaRepository;
5 import org.springframework.stereotype.Repository;
6
7 @Repository
8 public interface MahasiswaRepository extends JpaRepository<ModelMahasiswa, Long> {
9
10 }
```

## Output:

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='51421577', nama='alif', nama='0', jurusan='3.9'}
Mahasiswa{id=2, nama='51425166', nama='rahman', nama='0', jurusan='3.9'}
Mahasiswa{id=3, nama='56738999', nama='fathur', nama='0', jurusan='3.9'}
Mahasiswa{id=4, nama='5432677', nama='ganteng', nama='0', jurusan='4.0'}
```

## ACTIVITY 6

Nama : Alif Fathur Rahman

Kelas : 4IA28

Npm : 51421577

---

### Input:

MahasiswaApp.java

```
1  package com.mahasiswa;
2
3  import com.mahasiswa.controller.MahasiswaController;
4  import com.mahasiswa.service.MahasiswaService;
5  import com.mahasiswa.view.MahasiswaView;
6  import org.springframework.boot.ApplicationArguments;
7  import org.springframework.boot.ApplicationRunner;
8  import org.springframework.boot.SpringApplication;
9  import org.springframework.boot.autoconfigure.SpringBootApplication;
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 }
```

## MahasiswaController.java

```
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) {
37          mahasiswaService.deleteMhs(id);
```

## ModelMahasiswa.java

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

## ModelTabelMahasiswa.java

```
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               default:
```



## MahasiswaRepository.java

```
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             default:
```

## MahasiswaService.java

```
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 }
```

## MahasiswaView.java

```
1  /*
2  * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
3  * Click nbfs://nbhost/SystemFileSystem/Templates/GuiForms/JFrame.java to edit this template
4  */
5  package com.mahasiswa.view;
6
7  import com.mahasiswa.controller.MahasiswaController;
8  import com.mahasiswa.model.ModelMahasiswa;
9  import com.mahasiswa.model.ModelTabelMahasiswa;
10 import java.util.List;
11 import javax.swing.JLabel;
12 import javax.swing.JOptionPane;
13 import javax.swing.JPanel;
14 import javax.swing.JTextField;
15
16 /**
17  *
18  * @author alif
19  */
20 public class MahasiswaView extends javax.swing.JFrame {
21     private MahasiswaController controller;
22
23     /**
24      * Creates new form MahasiswaView
25      */
26     public MahasiswaView(MahasiswaController controller) {
27         this.controller = controller;
28         initComponents();
29         loadMahasiswaTable();
30     }
31
32     private MahasiswaView() {
33         throw new UnsupportedOperationException("Not supported yet."); // Generated from nbfs://nbhost/SystemFileSystem/Te
34     }
35
36     public void loadMahasiswaTable() {
```

**Output:**

**Simpan**

NPM

51421544

Nama

fathur

Semester

8

IPK

3.7

Buang

Simpan

ID	NPM	Nama	Semester	IPK
1	51421577	alif	0	4.0
2	51421544	fathur	0	3.9
3	51421544	fathur	0	3.7

**Buang**

NPM

51421544

Nama

fathur

Semester

8

IPK

3.7

Buang

Simpan

ID	NPM	Nama	Semester	IPK
1	51421577	alif	0	4.0
2	51421544	fathur	0	3.9
3	51421544	fathur	0	3.7

Hapus Mahasiswa

Masukkan ID yang ingin dihapus: 3

OK

Cancel

NPM

51421544

Nama

fathur

Semester

8

IPK

3.7

Buang

Simpan

ID	NPM	Nama	Semester	IPK
1	51421577	alif	0	4.0
2	51421544	fathur	0	3.9