

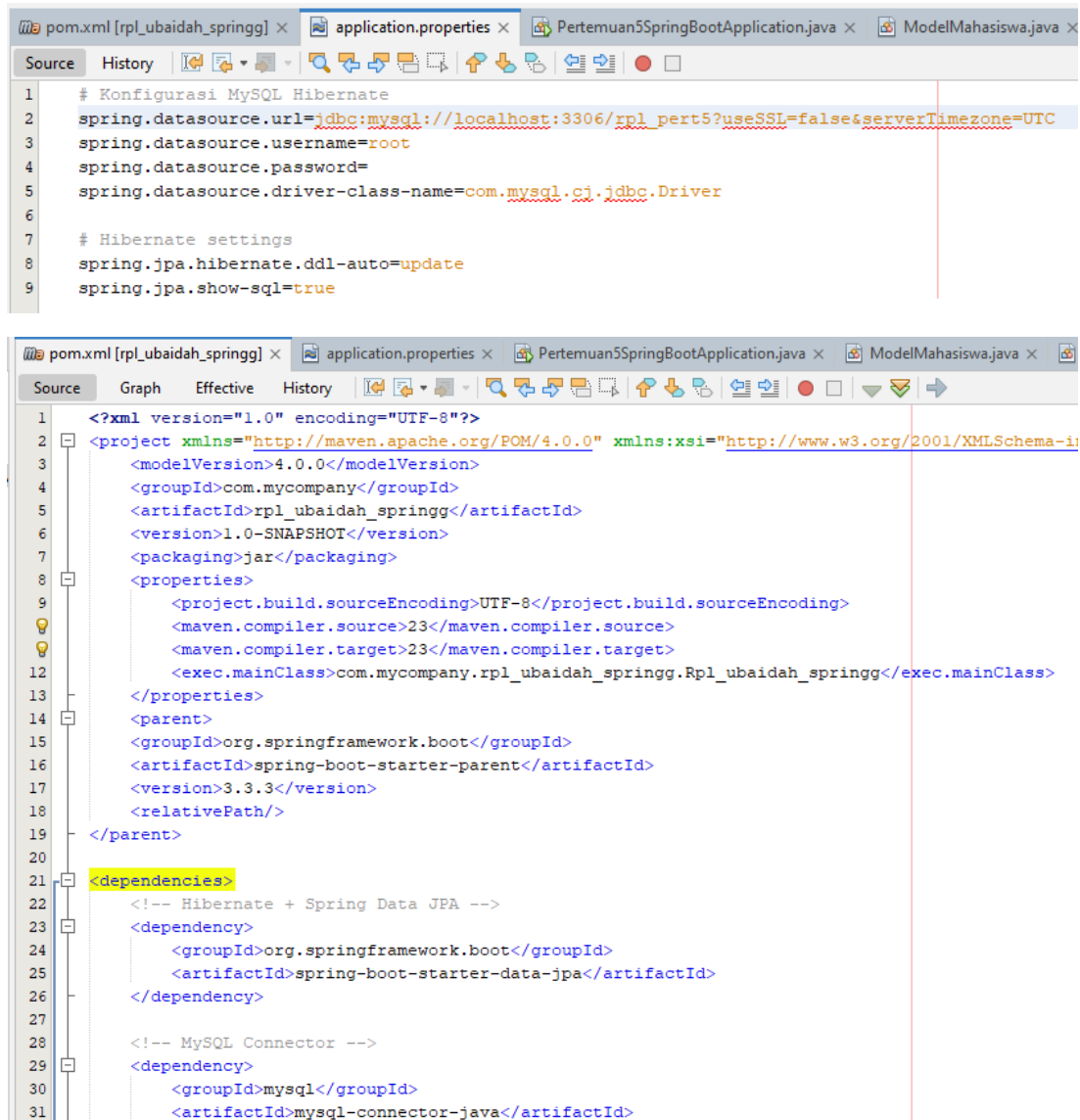
ACTIVITY PERTEMUAN 5

Nama : Ubaidah Luthfiyah Zain

NPM : 51421491

Kelas : 4IA28

Source Code



```
1 # Konfigurasi MySQL Hibernate
2 spring.datasource.url=jdbc:mysql://localhost:3306/rpl_pert5?useSSL=false&serverTimezone=UTC
3 spring.datasource.username=root
4 spring.datasource.password=
5 spring.datasource.driver-class-name=com.mysql.cj.jdbc.Driver
6
7 # Hibernate settings
8 spring.jpa.hibernate.ddl-auto=update
9 spring.jpa.show-sql=true


1 <?xml version="1.0" encoding="UTF-8"?>
2 <project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-i
3   <modelVersion>4.0.0</modelVersion>
4   <groupId>com.mycompany</groupId>
5   <artifactId>rpl_ubaidah_springg</artifactId>
6   <version>1.0-SNAPSHOT</version>
7   <packaging>jar</packaging>
8   <properties>
9     <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
10    <maven.compiler.source>23</maven.compiler.source>
11    <maven.compiler.target>23</maven.compiler.target>
12    <exec.mainClass>com.mycompany.rpl_ubaidah_springg.Rpl_ubaidah_springg</exec.mainClass>
13  </properties>
14  <parent>
15    <groupId>org.springframework.boot</groupId>
16    <artifactId>spring-boot-starter-parent</artifactId>
17    <version>3.3.3</version>
18    <relativePath/>
19  </parent>
20
21  <dependencies>
22    <!-- Hibernate + Spring Data JPA -->
23    <dependency>
24      <groupId>org.springframework.boot</groupId>
25      <artifactId>spring-boot-starter-data-jpa</artifactId>
26    </dependency>
27
28    <!-- MySQL Connector -->
29    <dependency>
30      <groupId>mysql</groupId>
31      <artifactId>mysql-connector-java</artifactId>
```

```

32 |         <version>8.0.33</version>
33 |     </dependency>
34 |
35 |     <!-- Spring Boot Web dependency (for MVC if needed) -->
36 |     <dependency>
37 |         <groupId>org.springframework.boot</groupId>
38 |         <artifactId>spring-boot-starter-web</artifactId>
39 |     </dependency>
40 |
41 |     <!-- Testing dependencies -->
42 |     <dependency>
43 |         <groupId>org.springframework.boot</groupId>
44 |         <artifactId>spring-boot-starter-test</artifactId>
45 |         <scope>test</scope>
46 |     </dependency>
47 | </dependencies>
48 |
49 | <build>
50 |     <plugins>
51 |         <plugin>
52 |             <groupId>org.springframework.boot</groupId>
53 |             <artifactId>spring-boot-maven-plugin</artifactId>
54 |         </plugin>
55 |     </plugins>
56 | </build>
57 | </project>

```

...es | Pertemuan5SpringBootApplication.java x | ModelMahasiswa.java x | MahasiswaController.java x | MahasiswaRep

Source | History | 

```

4 |  */
5 |  package com.mahasiswa;
6 |
7 |  /**
8 |   *
9 |   * @author ACER
10 |  */
11 |
12 |  import com.mahasiswa.controller.MahasiswaController;
13 |  import org.springframework.beans.factory.annotation.Autowired;
14 |  import org.springframework.boot.CommandLineRunner;
15 |  import org.springframework.boot.SpringApplication;
16 |  import org.springframework.boot.autoconfigure.SpringBootApplication;
17 |
18 |  @SpringBootApplication
19 |  public class Pertemuan5SpringBootApplication implements CommandLineRunner{
20 |
21 |      @Autowired
22 |      private MahasiswaController mhsController;
23 |      public static void main(String[] args) {
24 |          SpringApplication.run(primarySource: Pertemuan5SpringBootApplication.class, args);
25 |      }
26 |
27 |      @Override
28 |      public void run(String... args) throws Exception {
29 |          mhsController.tampilkanMenu();
30 |      }
31 |
32 |  }

```

```
...es Pertemuan5SpringBootApplication.java x ModelMahasiswa.java x MahasiswaController.java x Maha
Source History
5 package com.mahasiswa.controller;
6
7 /**
8  *
9  * @author ACER
10 */
11 import com.mahasiswa.model.ModelMahasiswa;
12 import com.mahasiswa.repository.MahasiswaRepository;
13 import org.springframework.beans.factory.annotation.Autowired;
14 import org.springframework.stereotype.Controller;
15
16 import java.util.List;
17 import java.util.Scanner;
18
19 @Controller
20 public class MahasiswaController {
21
22     @Autowired
23     private MahasiswaRepository mahasiswaRepository;
24
25     public void tampilkanMenu() {
26         Scanner scanner = new Scanner(System.in);
27         int opsi;
28
29         do {
30             System.out.println("\nMenu:");
31             System.out.println("1. Tampilkan semua mahasiswa");
32             System.out.println("2. Tambah mahasiswa baru");
33             System.out.println("3. Cek koneksi database");
34             System.out.println("4. Keluar");
35             System.out.print("Pilih opsi: ");
36
37             opsi = scanner.nextInt();
38             scanner.nextLine(); // menangkap newline
39
40             switch (opsi) {
41                 case 1:
42                     tampilkanSemuaMahasiswa();
43                     break;
44                 case 2:
45                     tambahMahasiswa(scanner);
46                     break;
47                 case 3:
48                     cekKoneksi();
49                     break;
50                 case 4:
51                     System.out.println("Keluar dari program.");
52                     break;
53                 default:
54                     System.out.println("Ops! tidak valid, coba lagi.");
55             }
56         } while (opsi != 4);
57     }
58
59     private void tampilkanSemuaMahasiswa() {
60         List<ModelMahasiswa> mahasiswaList = mahasiswaRepository.findAll();
61         if (mahasiswaList.isEmpty()) {
62             System.out.println("Tidak ada data mahasiswa.");
63         } else {
64             mahasiswaList.forEach(mahasiswa -> System.out.println(mahasiswa));
65         }
66     }
67 }
```

```

68 private void tambahMahasiswa(Scanner scanner) {
69     System.out.print(s: "Masukkan NPM : ");
70     String npm = scanner.nextLine();
71     System.out.print(s: "Masukkan Nama : ");
72     String nama = scanner.nextLine();
73     System.out.print(s: "Masukkan Semester : ");
74     int semester = scanner.nextInt();
75     System.out.print(s: "Masukkan IPK : ");
76     float ipk = scanner.nextFloat();
77
78     ModelMahasiswa mahasiswa = new ModelMahasiswa(id: 0, npm, nama, semester, ipk);
79     mahasiswaRepository.save(entity: mahasiswa);
80     System.out.println(s: "Mahasiswa berhasil ditambahkan.");
81 }
82
83 private void cekKoneksi() {
84     try {
85         mahasiswaRepository.findAll();
86         System.out.println(s: "Koneksi ke database berhasil.");
87     } catch (Exception e) {
88         System.out.println(s: "Gagal terhubung ke database.");
89     }
90 }
91
92

```

...es Pertemuan5SpringBootApplication.java x ModelMahasiswa.java x MahasiswaController.java x Ma

Source History

```

4
5 package com.mahasiswa.model;
6
7 import jakarta.persistence.*;
8
9 /**
10  *
11  * @author ACER
12  */
13 @Entity
14 @Table(name = "mahasiswa")
15
16 public class ModelMahasiswa {
17     @Id
18     @GeneratedValue(strategy = GenerationType.IDENTITY)
19     @Column(name = "id")
20     private int id;
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 = "semester")
29     private int semester;
30
31     @Column(name = "ipk")
32     private float ipk;
33
34     public ModelMahasiswa() {
35

```

```

38 public ModelMahasiswa(int id, String npm, String nama, int semester, float ipk){
39     this.id = id;
40     this.npm = npm;
41     this.nama = nama;
42     this.semester = semester;
43     this.ipk = ipk;
44 }
45
46 public int getId() {
47     return id;
48 }
49
50 public void setId(int id) {
51     this.id = id;
52 }
53
54 public String getNpm() {
55     return npm;
56 }
57
58 public void setNpm(String npm) {
59     this.npm = npm;
60 }
61
62 public String getName() {
63     return nama;
64 }
65
66 public void setName(String nama) {
67     this.nama = nama;
68 }

```

```

69
70 public int getSemester() {
71     return semester;
72 }
73
74 public void setSemester(int semester) {
75     this.semester = semester;
76 }
77
78 public float getIpk() {
79     return ipk;
80 }
81
82 public void setIpk(float ipk) {
83     this.ipk = ipk;
84 }
85
86 @Override
87 public String toString() {
88     return "Mahasiswa{" +
89         "id=" + id +
90         ", npm='" + npm + '\'' +
91         ", nama='" + nama + '\'' +
92         ", semester='" + semester + '\'' +
93         ", ipk='" + ipk + '\'' +
94         '}';
95 }
96 }
97

```

```
...es Pertemuan5SpringBootApplication.java x ModelMahasiswa.java x MahasiswaController.java x MahasiswaRepository.java x
Source History
2  /* Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
3  * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Interface.java to edit this template
4  */
5  package com.mahasiswa.repository;
6
7  /**
8   *
9   * @author ACER
10  */
11  import com.mahasiswa.model.ModelMahasiswa;
12  import org.springframework.data.jpa.repository.JpaRepository;
13  import org.springframework.stereotype.Repository;
14
15  @Repository
16  public interface MahasiswaRepository extends JpaRepository<ModelMahasiswa, Long> {
17
18  }
19
```

Output

```
Output - Run (Pertemuan5SpringBootApplication) x
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.

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, npm='51421491', nama='Ubaidah', semester='7', ipk='3.8'}
Mahasiswa{id=2, npm='50421244', nama='Ayu', semester='7', ipk='3.75'}
Mahasiswa{id=3, npm='4', nama='2', semester='2', ipk='2.0'}

Menu:
1. Tampilkan semua mahasiswa
2. Tambah mahasiswa baru
3. Cek koneksi database
4. Keluar
Pilih opsi: 2
Masukkan NPM : 50421176
Masukkan Nama : Annisa
Masukkan Semester : 7
Masukkan IPK : 3,7
```



```
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 mml_0.id,mml_0.ipk,mml_0.nama,mml_0.npm,mml_0.semester from mahasiswa mml_0
```

```
Mahasiswa{id=1, npm='51421491', nama='Ubaidah', semester='7', ipk='3.8'}
```

```
Mahasiswa{id=2, npm='50421244', nama='Ayu', semester='7', ipk='3.75'}
```

```
Mahasiswa{id=3, npm='4', nama='2', semester='2', ipk='2.0'}
```

```
Mahasiswa{id=4, npm='50421176', nama='Annisa', semester='7', ipk='3.7'}
```

Menu:

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

Pilih opsi: 4

Keluar dari program.

ACTIVITY PERTEMUAN 6

Nama : Ubaidah Luthfiyah Zain

NPM : 51421491

Kelas : 4IA28

Source Code

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-ins
3     <modelVersion>4.0.0</modelVersion>
4     <groupId>com.mycompany</groupId>
5     <artifactId>rp_UBaidah_springAop</artifactId>
6     <version>1.0-SNAPSHOT</version>
7     <packaging>jar</packaging>
8     <properties>
9         <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
10        <maven.compiler.source>23</maven.compiler.source>
11        <maven.compiler.target>23</maven.compiler.target>
12        <exec.mainClass>com.mycompany.rp_UBaidah_springAop.Rp_UBaidah_springAop</exec.mainClass>
13    </properties>
14    <parent>
15        <groupId>org.springframework.boot</groupId>
16        <artifactId>spring-boot-starter-parent</artifactId>
17        <version>3.3.3</version>
18        <relativePath/>
19    </parent>
20
21    <dependencies>
22        <!-- Hibernate + Spring Data JPA -->
23        <dependency>
24            <groupId>org.springframework.boot</groupId>
25            <artifactId>spring-boot-starter-data-jpa</artifactId>
26        </dependency>
27
28        <!-- MySQL Connector -->
29        <dependency>
30            <groupId>mysql</groupId>
31            <artifactId>mysql-connector-java</artifactId>
32
33            <version>8.0.33</version>
34        </dependency>
35
36        <!-- Spring Boot Web dependency (for MVC if needed) -->
37        <dependency>
38            <groupId>org.springframework.boot</groupId>
39            <artifactId>spring-boot-starter-web</artifactId>
40        </dependency>
41
42        <!-- Testing dependencies -->
43        <dependency>
44            <groupId>org.springframework.boot</groupId>
45            <artifactId>spring-boot-starter-test</artifactId>
46            <scope>test</scope>
47        </dependency>
48    </dependencies>
49
50    <build>
51        <plugins>
52            <plugin>
53                <groupId>org.springframework.boot</groupId>
54                <artifactId>spring-boot-maven-plugin</artifactId>
55            </plugin>
56        </plugins>
57    </build>
58 </project>
```



```
MahasiswaController.java x MahasiswaView.java x application.properties x MahasiswaApp.java x
Source History
1 # Konfigurasi MySQL Hibernate
2 spring.datasource.url=jdbc:mysql://localhost:3306/rpl_pert6?useSSL=false&serverTimezone=UTC
3 spring.datasource.username=root
4 spring.datasource.password=
5 spring.datasource.driver-class-name=com.mysql.cj.jdbc.Driver
6
7 # Hibernate settings
8 spring.jpa.hibernate.ddl-auto=update
9 spring.jpa.show-sql=true
```

```
pom.xml [rpl_ubaidah_springAop] x application.properties x ModelMahasiswa.java x ModelTabelMah
Source History
5 package com.mahasiswa.model;
6
7 import jakarta.persistence.*;
8
9 /**
10  *
11  * @author ACER
12  */
13 @Entity
14 @Table(name = "mahasiswa")
15
16 public class ModelMahasiswa {
17     @Id
18     @GeneratedValue(strategy = GenerationType.IDENTITY)
19     @Column(name = "id")
20     private int id;
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 = "semester")
29     private int semester;
30
31     @Column(name = "ipk")
32     private float ipk;
33
34     public ModelMahasiswa() {
35
```

```

36     }
37
38     public ModelMahasiswa(int id, String npm, String nama, int semester, float ipk){
39         this.id = id;
40         this.npm = npm;
41         this.nama = nama;
42         this.semester = semester;
43         this.ipk = ipk;
44     }
45
46     public int getId() {
47         return id;
48     }
49
50     public void setId(int id) {
51         this.id = id;
52     }
53
54     public String getNpm() {
55         return npm;
56     }
57
58     public void setNpm(String npm) {
59         this.npm = npm;
60     }
61
62     public String getName() {
63         return nama;
64     }
65
66     public void setName(String nama) {
67         this.nama = nama;
68     }
69
70     public int getSemester() {
71         return semester;
72     }
73
74     public void setSemester(int semester) {
75         this.semester = semester;
76     }
77
78     public float getIpk() {
79         return ipk;
80     }
81
82     public void setIpk(float ipk) {
83         this.ipk = ipk;
84     }
85 }

```

```
pom.xml [rp_lubaidah_springAop] x MahasiswaRepository.java x application.properties x ModelMahasiswa.java
Source History
2  /*
3  * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change the license
4  * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Interface.java to edit this template
5  */
6
7  package com.mahasiswa.repository;
8
9  /**
10   *
11   * @author ACER
12   */
13
14  import com.mahasiswa.model.ModelMahasiswa;
15  import org.springframework.data.jpa.repository.JpaRepository;
16  import org.springframework.stereotype.Repository;
17
18  @Repository
19  public interface MahasiswaRepository extends JpaRepository<ModelMahasiswa, Integer> {
20      public Object findById(int id);
21
22      public void deleteById(int id);
23  }
```

```
MahasiswaRepository.java x ModelMahasiswa.java x ModelTabelMahasiswa.java x MahasiswaService.java
Source History
5  package com.mahasiswa.service;
6
7  /**
8   *
9   * @author ACER
10   */
11
12  import com.mahasiswa.model.ModelMahasiswa;
13  import com.mahasiswa.repository.MahasiswaRepository;
14  import jakarta.transaction.Transactional;
15  import java.util.List;
16  import org.springframework.beans.factory.annotation.Autowired;
17  import org.springframework.stereotype.Service;
18
19  @Service
20  public class MahasiswaService {
21
22      @Autowired
23      private MahasiswaRepository repository;
24
25      public void addMhs(ModelMahasiswa mhs) {
26          repository.save(entity: mhs);
27      }
28
29      public ModelMahasiswa getMhs(int id) {
30          ModelMahasiswa mahasiswa = (ModelMahasiswa) repository.findById(id);
31          return mahasiswa != null ? mahasiswa : null;
32      }
33
34      public void updateMhs(ModelMahasiswa mhs) {
35          repository.save(entity: mhs);
36      }
37
38      @Transactional
39      public void deleteMhs(int id) {
40          repository.deleteById(id);
41      }
42
43      public List<ModelMahasiswa> getAllMahasiswa() {
44          return repository.findAll();
45      }
46  }
```

```
ModelTabelMahasiswa.java x MahasiswaController.java x
Source History

package com.mahasiswa.model;

/**
 * @author ACER
 */
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(index: rowIndex);
        switch (columnIndex) {

            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
    }
}
```

```
MahasiswaController.java x MahasiswaView.java x
Source History
5 package com.mahasiswa.controller;
6
7 /**
8  *
9  * @author ACER
10 */
11 import org.springframework.beans.factory.annotation.Autowired;
12 import org.springframework.web.bind.annotation.*;
13 import com.mahasiswa.model.ModelMahasiswa;
14 import com.mahasiswa.service.MahasiswaService;
15
16 import java.util.List;
17 import org.springframework.stereotype.Controller;
18
19 @Controller
20 public class MahasiswaController {
21
22     @Autowired
23     private MahasiswaService mahasiswaService;
24
25     // Add new Mahasiswa
26     public String addMahasiswa(@RequestBody ModelMahasiswa mhs) {
27         mahasiswaService.addMhs(mhs);
28         return "Mahasiswa added successfully";
29     }
30
31     // Get Mahasiswa by ID
32     public ModelMahasiswa getMahasiswa(@PathVariable int id) {
33         return mahasiswaService.getMhs(id);
34     }
35
36     // Update Mahasiswa
37     public String updateMahasiswa(@RequestBody ModelMahasiswa mhs) {
38         mahasiswaService.updateMhs(mhs);
39         return "Mahasiswa updated successfully";
40     }
41
42     // Delete Mahasiswa by ID
43     public String deleteMahasiswa(@PathVariable int id) {
44         mahasiswaService.deleteMhs(id);
45         return "Mahasiswa deleted successfully";
46     }
47
48     // Get all Mahasiswa
49     public List<ModelMahasiswa> getAllMahasiswa() {
50         return mahasiswaService.getAllMahasiswa();
51     }
52 }
53
54
```

```
MahasiswaController.java x MahasiswaView.java x MahasiswaApp.java x application.properties x
Source History
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(key:"java.awt.headless", value: "false"); // Disable headless mode

        // Start the Spring application and get the application context
        ApplicationContext context = SpringApplication.run(primarySource: MahasiswaApp.class, args);

        // Instantiate the view and inject the controller manually
        MahasiswaController controller = context.getBean(requiredType: MahasiswaController.class);
        MahasiswaView mahasiswaView = new MahasiswaView(controller);
        mahasiswaView.setVisible(b: 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.
    }
}
```

```
MahasiswaView.java x application.properties x MahasiswaApp.java x
Source Design History
private void simpanButtonActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    String npm = getNpmField().getText();
    String nama = getNamaField().getText();
    int semester = Integer.parseInt(s: getSemesterField().getText());
    float ipk = Float.parseFloat(s: getIpkField().getText());
    ModelMahasiswa mahasiswa = new ModelMahasiswa(id: 0, npm, nama, semester, ipk);
    System.out.println(x: mahasiswa.getId());
    System.out.println(x: mahasiswa.getNama());
    System.out.println(x: mahasiswa.getSemester());
    System.out.println(x: mahasiswa.getNpm());

    controller.addMahasiswa(mhs:mahasiswa);
    loadMahasiswaTable();
}

private void buangButtonActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    JTextField idField = new JTextField(columns:5);

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

    // Menampilkan dialog box dengan JTextField, tombol OK, dan Cancel
    int result = JOptionPane.showConfirmDialog(parentComponent:null, message:panel,
    title: "Hapus Mahasiswa", optionType: JOptionPane.OK_CANCEL_OPTION, messageType:JOptionPane.PLAIN_MESSAGE);
}
```

```

201 // Jika tombol OK ditekan
202 if (result == JOptionPane.OK_OPTION) {
203     try {
204         // Mengambil input ID dan memanggil metode deleteMhs
205         int id = Integer.parseInt(idField.getText());
206         controller.deleteMahasiswa(id);
207         JOptionPane.showMessageDialog(parentComponent:null, message:"Data berhasil dihapus.", title:"Sukses",
208     } catch (NumberFormatException e) {
209         // Menangani error jika ID yang dimasukkan bukan angka
210         JOptionPane.showMessageDialog(parentComponent:null, message:"ID harus berupa angka.", title:"Error",
211     }
212 }
213 loadMahasiswaTable();
214
215 }

```

MahasiswaController.java x MahasiswaView.java x application.properties x

Source Design History

The Tools>Palette>Swing/AWT Components menu item allows you to modify the content of the Palette.

NPM

Nama

Semester

IPK

Title 1	Title 2	Title 3	Title 4

Output

NPM

51421491

Nama

Ubaidah

Semester

7

IPK

3.80

Simpan

Buang

ID	NPM	Nama	Semester	IPK
----	-----	------	----------	-----

NPM

51421411

Nama

Annisa

Semester

7

IPK

3.7

Simpan

Buang

ID	NPM	Nama	Semester	IPK
1	51421491	Ubaidah	7	3.8
2	50421244	Ayu	7	3.75
3	51421411	Annisa	7	3.7

Formulir input data mahasiswa:

NPM:

Nama:

Semester:

IPK:

Modal dialog: Hapus Mahasiswa

Masukkan ID yang ingin dihapus:

ID	NPM	IPK
1	51421491	
2	50421244	
3	51421411	

Formulir input data mahasiswa:

NPM:

Nama:

Semester:

IPK:

ID	NPM	Nama	Semester	IPK
1	51421491	Ubaidah	7	3.8
2	50421244	Ayu	7	3.75
3	51421411	Annisa	7	3.7

Modal dialog: Sukses

Data berhasil dihapus.

NPM

51421411

Simpan

Nama

Annisa

Buang

Semester

7

IPK

3.7

ID	NPM	Nama	Semester	IPK
1	51421491	Ubaidah	7	3.8
2	50421244	Ayu	7	3.75