

Activity 5

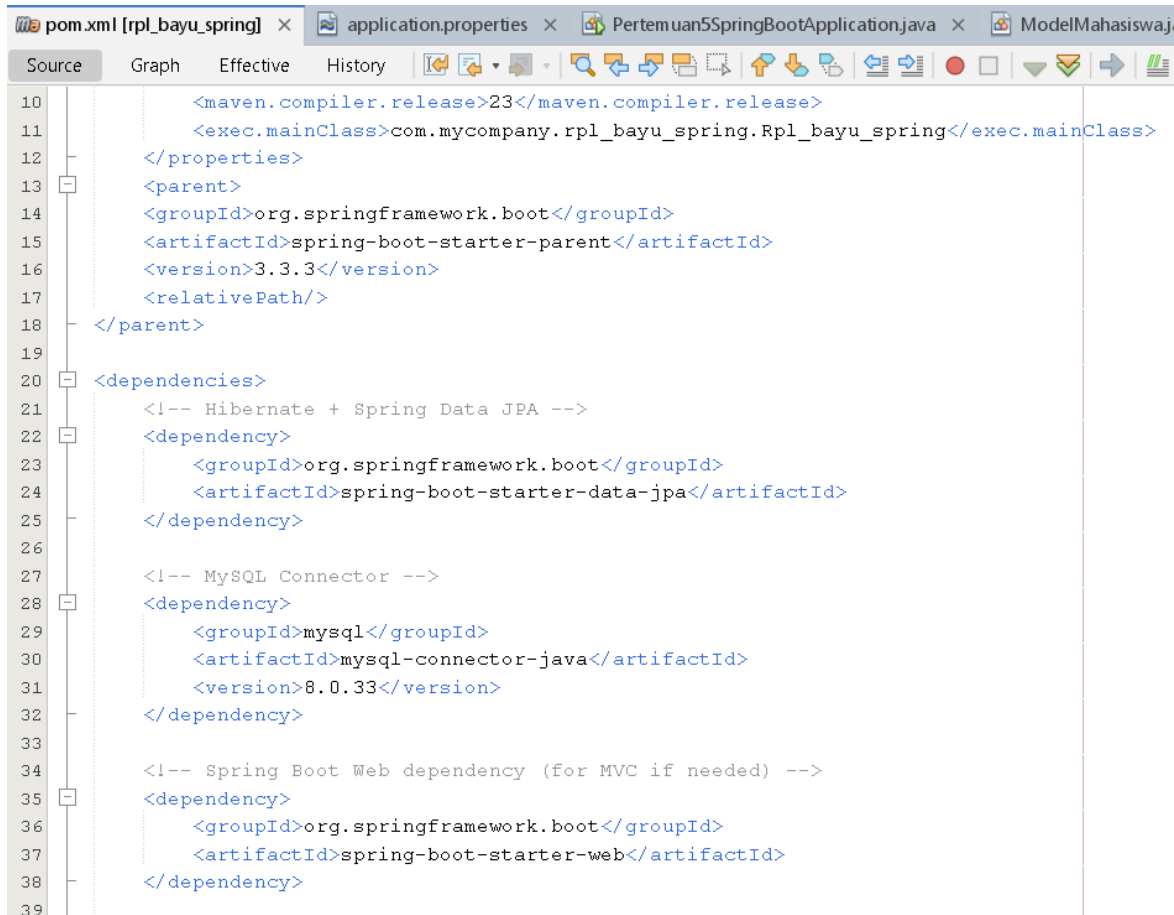
Nama : Bayu Tirta Ardi

Kelas : 4IA28

NPM : 50421268

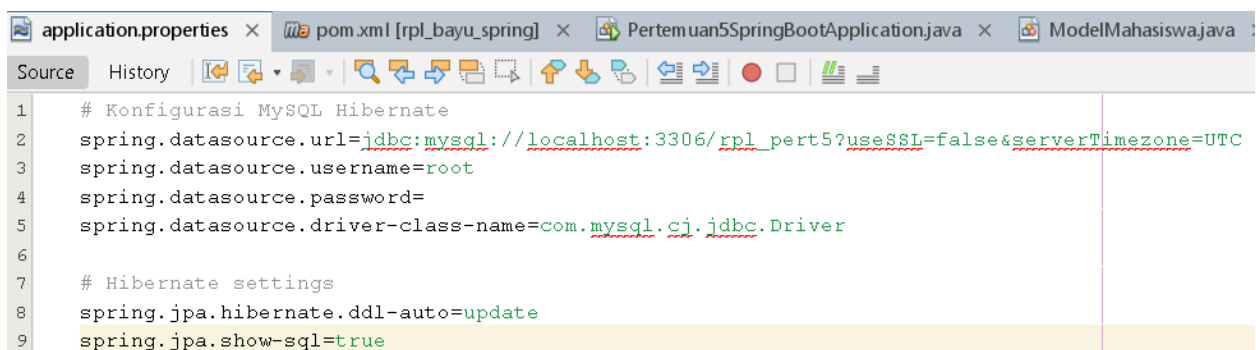
Mata Praktikum : Rekayasa Perangkat Lunak 2

pom.xml



```
10      <maven.compiler.release>23</maven.compiler.release>
11      <exec.mainClass>com.mycompany.rpl_bayu_spring.Rpl_bayu_spring</exec.mainClass>
12    </properties>
13    <parent>
14      <groupId>org.springframework.boot</groupId>
15      <artifactId>spring-boot-starter-parent</artifactId>
16      <version>3.3.3</version>
17      <relativePath/>
18    </parent>
19
20    <dependencies>
21      <!-- Hibernate + Spring Data JPA -->
22      <dependency>
23        <groupId>org.springframework.boot</groupId>
24        <artifactId>spring-boot-starter-data-jpa</artifactId>
25      </dependency>
26
27      <!-- MySQL Connector -->
28      <dependency>
29        <groupId>mysql</groupId>
30        <artifactId>mysql-connector-java</artifactId>
31        <version>8.0.33</version>
32      </dependency>
33
34      <!-- Spring Boot Web dependency (for MVC if needed) -->
35      <dependency>
36        <groupId>org.springframework.boot</groupId>
37        <artifactId>spring-boot-starter-web</artifactId>
38      </dependency>
39    </dependencies>
```

application.properties



```
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
```

Pertemuan5SpringBootApplication

```
application.properties x pom.xml [rpl_bayu_spring] x Pertemuan5SpringBootApplication.java x M
Source History
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 }
```

ModelMahasiswa.java

```
application.properties x pom.xml [rpl_bayu_spring] x Pertemuan5SpringBootApplication.java x ModelMahasiswa.java
Source History
16 @GeneratedValue(strategy = GenerationType.IDENTITY)
17
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() {
```

MahasiswaController.java

```
es MahasiswaController.java x pom.xml [rpl_bayu_spring] x Pertemuan5SpringBootApplication.java x
source History

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);
}
```

MahasiswaRepository.java

```
..va MahasiswaRepository.java x pom.xml [rpl_bayu_spring] x Pertemuan5SpringBootApplication.java x Mode
source History

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 }
```

Output

```
Output - Run (Pertemuan5SpringBootApplication) x
2024-11-02T10:31:33.250+07:00 INFO 2360 --- [main] com.zaxxer.hikari.pool.HikariPo
2024-11-02T10:31:33.252+07:00 INFO 2360 --- [main] com.zaxxer.hikari.HikariDataSou
2024-11-02T10:31:33.290+07:00 WARN 2360 --- [main] org.hibernate.dialect.Dialect
2024-11-02T10:31:33.803+07:00 INFO 2360 --- [main] o.h.e.t.j.p.i.JtaPlatformInitia
2024-11-02T10:31:33.824+07:00 INFO 2360 --- [main] j.LocalContainerEntityManagerFa
2024-11-02T10:31:34.033+07:00 WARN 2360 --- [main] JpaBaseConfiguration$JpaWebConf
2024-11-02T10:31:34.306+07:00 INFO 2360 --- [main] o.s.b.w.embedded.tomcat.TomcatW
2024-11-02T10:31:34.321+07:00 INFO 2360 --- [main] c.m.Pertemuan5SpringBootApplica

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, npm='50421268', nama='Bayu Tirta Ardi', semester='7', ipk='3.72')
Mahasiswa(id=2, npm='51428237', nama='Sandiaz Radjaoloan Gultom', semester='7', ipk='3.8')
Mahasiswa(id=3, npm='5141239', nama='Muhhamad Faiz Ramadhan', semester='7', ipk='3.52')
```

Activity 6

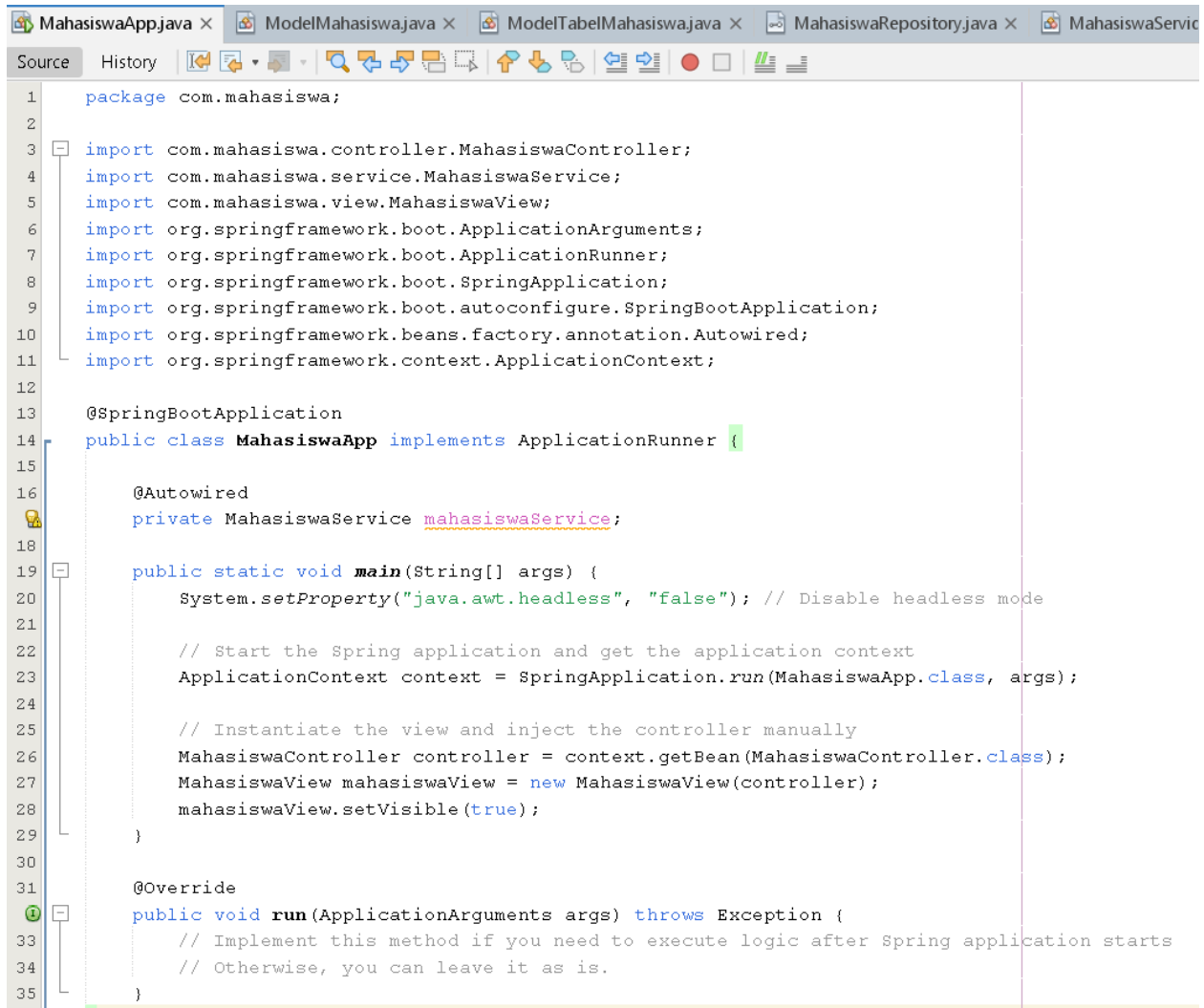
Nama : Bayu Tirta Ardi

Kelas : 4IA28

NPM : 50421268

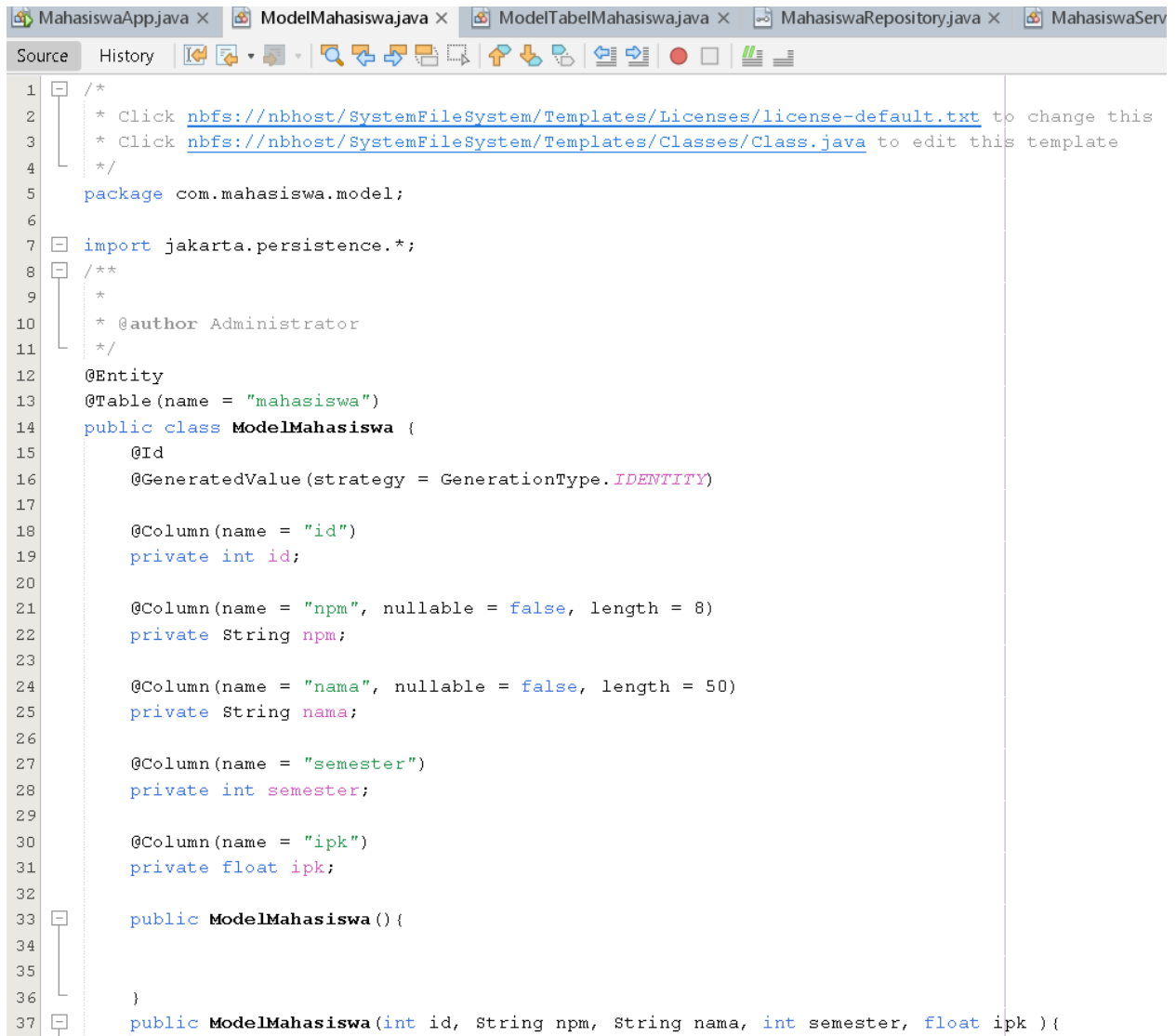
Mata Praktikum : Rekayasa Perangkat Lunak 2

MahasiswaApp.java



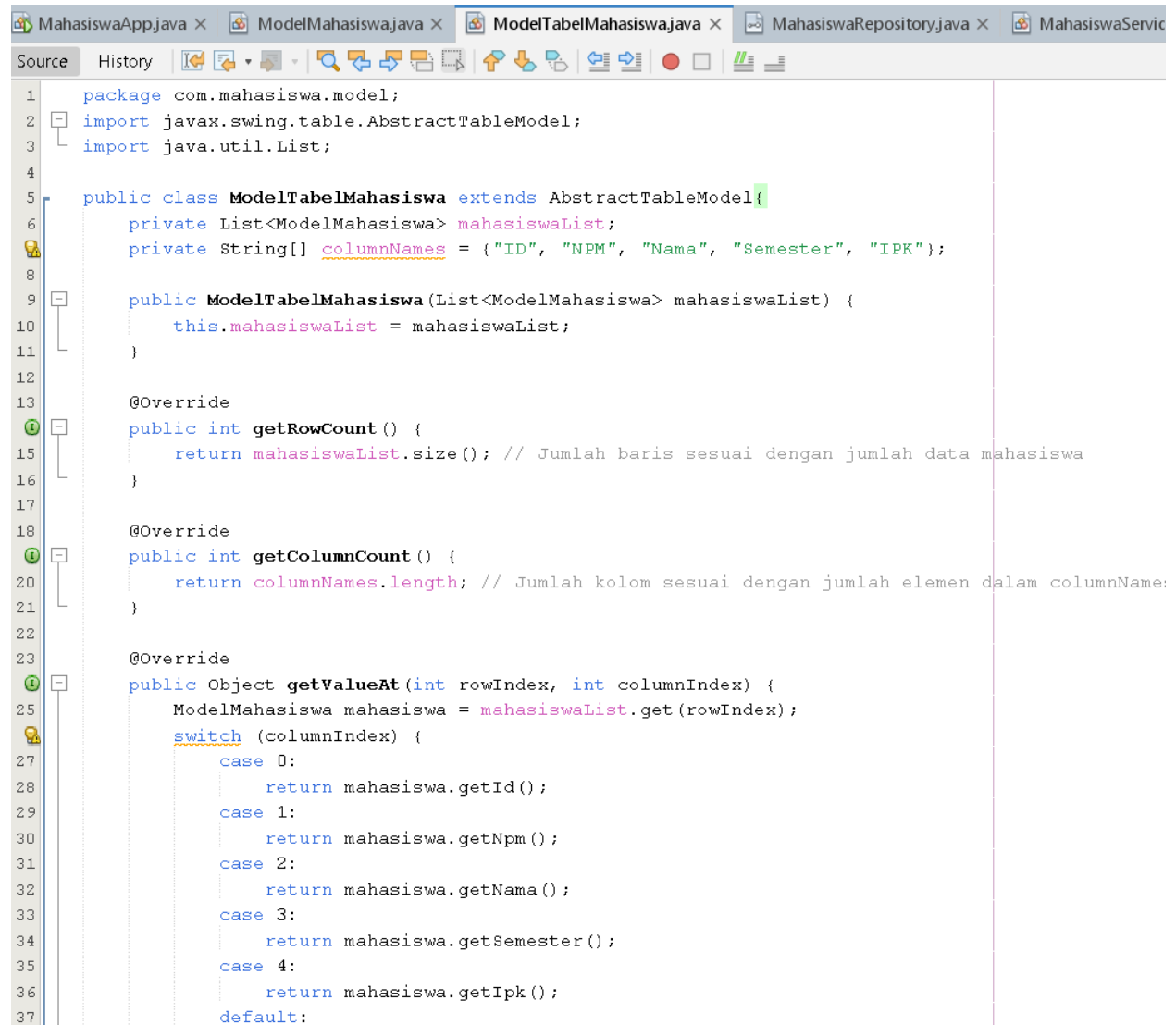
```
1 package com.mahasiswa;
2
3 import com.mahasiswa.controller.MahasiswaController;
4 import com.mahasiswa.service.MahasiswaService;
5 import com.mahasiswa.view.MahasiswaView;
6 import org.springframework.boot.ApplicationArguments;
7 import org.springframework.boot.ApplicationRunner;
8 import org.springframework.boot.SpringApplication;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

ModelMahasiswa.java



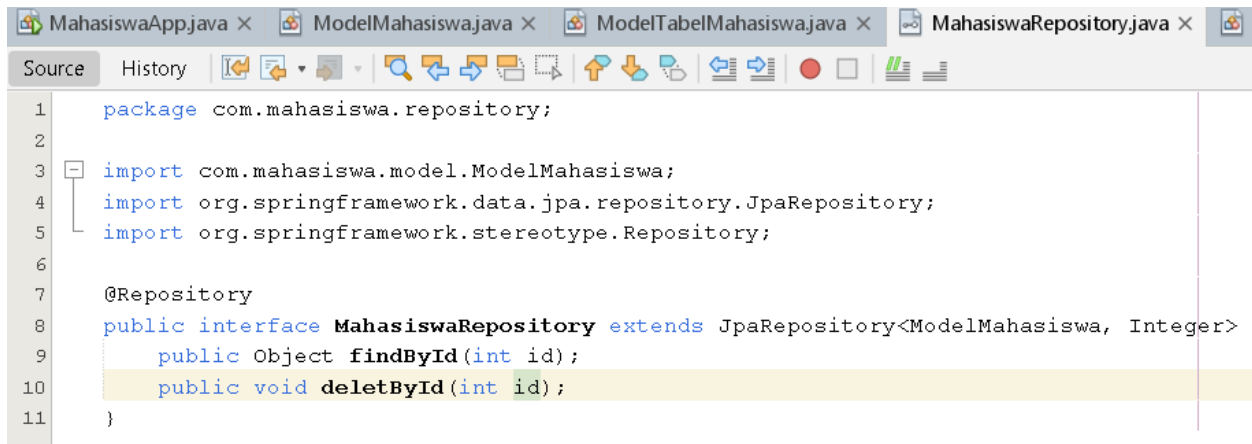
```
1  /**
2   * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this
3   * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this template
4   */
5   package com.mahasiswa.model;
6
7   import jakarta.persistence.*;
8   /**
9    *
10   * @author Administrator
11   */
12   @Entity
13   @Table(name = "mahasiswa")
14   public class ModelMahasiswa {
15       @Id
16       @GeneratedValue(strategy = GenerationType.IDENTITY)
17
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) {
```

ModelTableMahasiswa.java



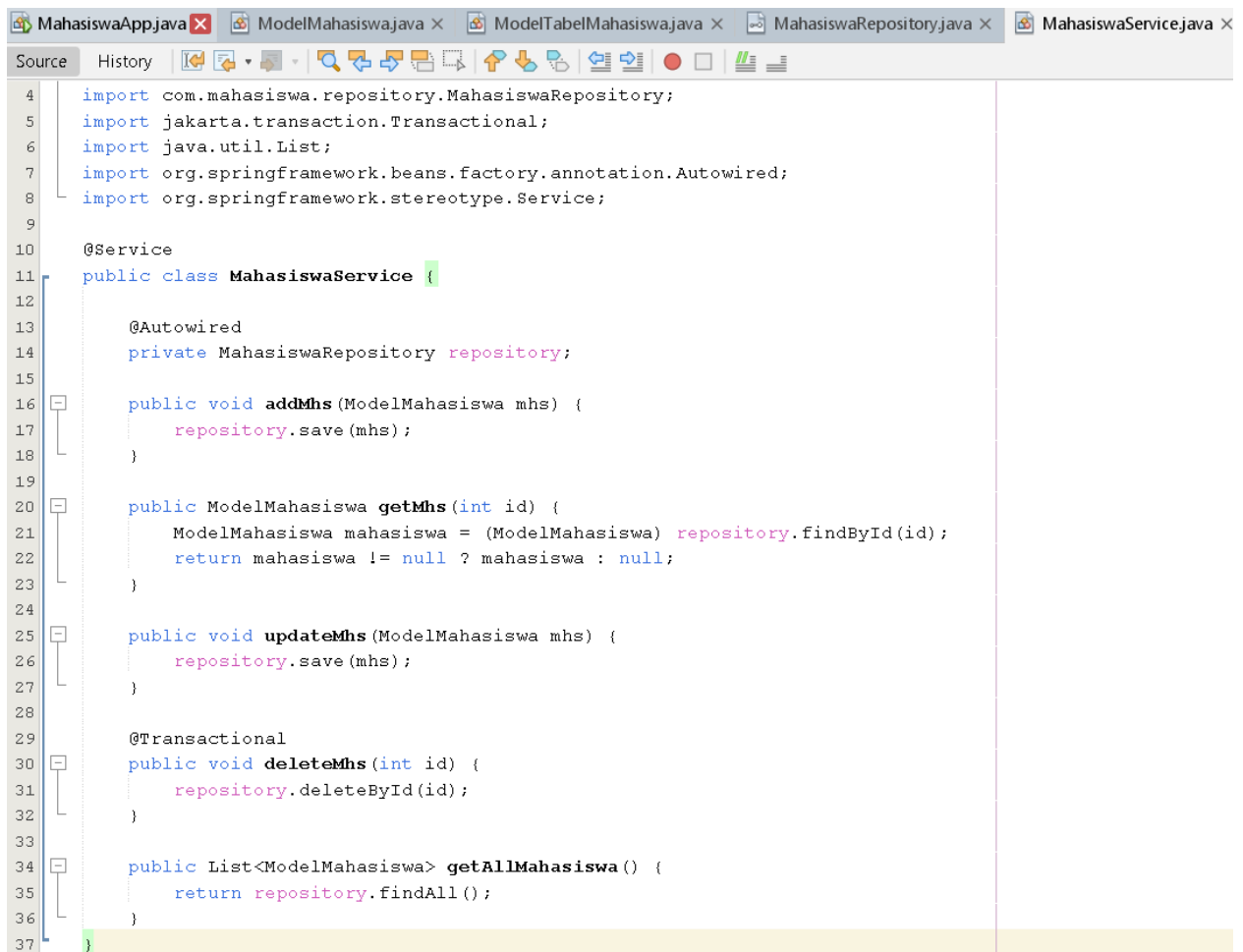
```
1 package com.mahasiswa.model;
2 import javax.swing.table.AbstractTableModel;
3 import java.util.List;
4
5 public class ModelTableMahasiswa extends AbstractTableModel {
6     private List<ModelMahasiswa> mahasiswaList;
7     private String[] columnNames = {"ID", "NPM", "Nama", "Semester", "IPK"};
8
9     public ModelTableMahasiswa(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 columnName
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



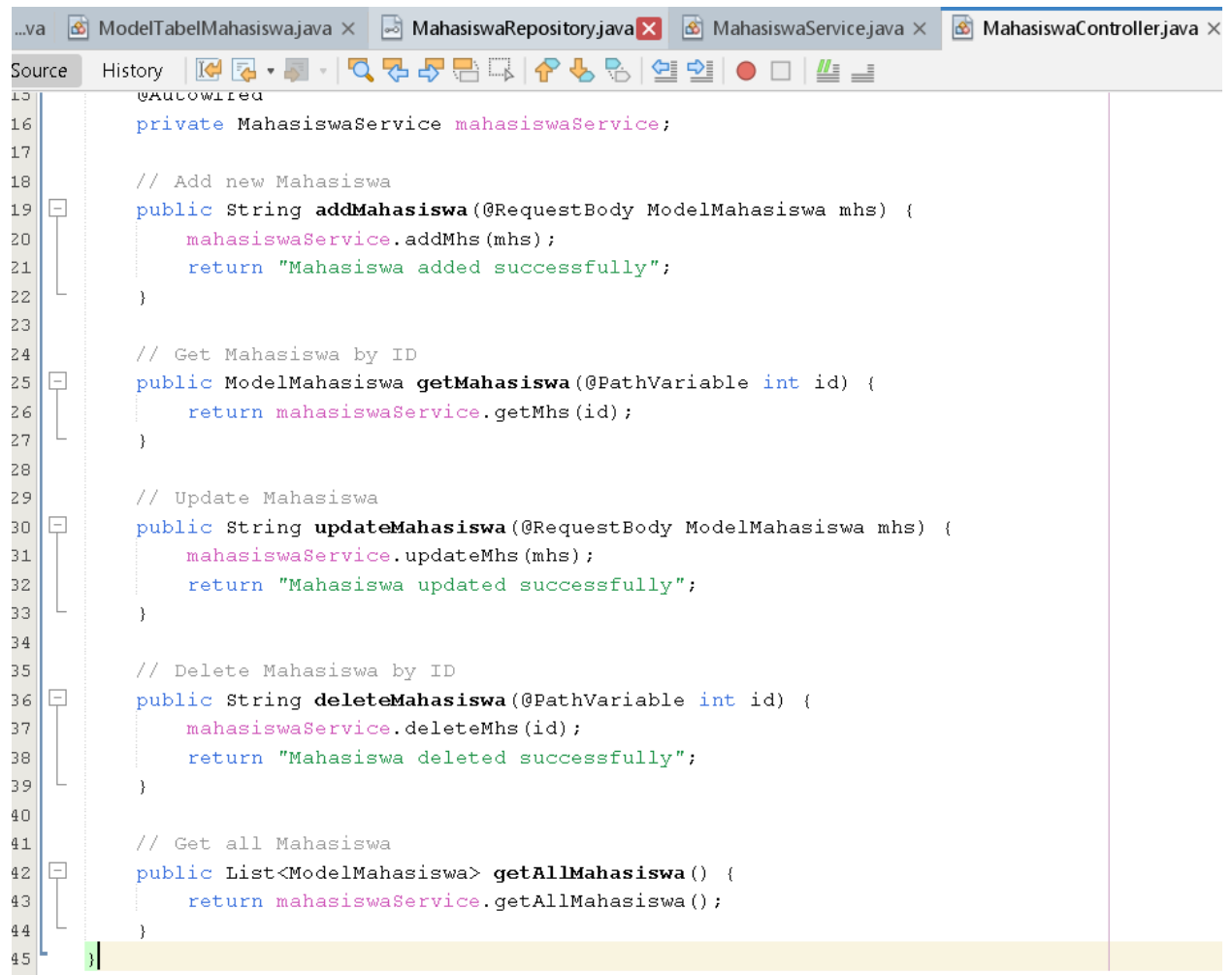
```
1 package com.mahasiswa.repository;
2
3 import com.mahasiswa.model.ModelMahasiswa;
4 import org.springframework.data.jpa.repository.JpaRepository;
5 import org.springframework.stereotype.Repository;
6
7 @Repository
8 public interface MahasiswaRepository extends JpaRepository<ModelMahasiswa, Integer>
9     public Object findById(int id);
10     public void deleteById(int id);
11 }
```

MahasiswaService.java



```
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 }
```


MahasiswaController.java



```
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);
38      return "Mahasiswa deleted successfully";
39  }
40
41  // Get all Mahasiswa
42  public List<ModelMahasiswa> getAllMahasiswa() {
43      return mahasiswaService.getAllMahasiswa();
44  }
45  }
```

MahasiswaView.java

```
...va ModelTabelMahasiswa.java x MahasiswaRepository.java x MahasiswaService.java x MahasiswaController.java x MahasiswaView.java x
Source Design History
144
145 public void setSemesterField(JLabel semesterField) {
146     this.semesterField = semesterField;
147 }
148
149 private void buangButtonActionPerformed(java.awt.event.ActionEvent evt) {
150     JTextField idField = new JTextField(5);
151
152     // Membuat panel untuk menampung JTextField
153     JPanel panel = new JPanel();
154     panel.add(new JLabel("Masukkan ID yang ingin dihapus:"));
155     panel.add(idField);
156
157     // Menampilkan dialog box dengan JTextField, tombol OK, dan Cancel
158     int result = JOptionPane.showConfirmDialog(null, panel,
159         "Hapus Mahasiswa", JOptionPane.OK_CANCEL_OPTION, JOptionPane.PLAIN_MESSAGE);
160
161     // Jika tombol OK ditekan
162     if (result == JOptionPane.OK_OPTION) {
163         try {
164             // Mengambil input ID dan memanggil metode deleteMhs
165             int id = Integer.parseInt(idField.getText());
166             controller.deleteMahasiswa(id);
167             JOptionPane.showMessageDialog(null, "Data berhasil dihapus.", "Sukses", JOptionPane.INFORMATION_MESSAGE);
168         } catch (NumberFormatException e) {
169             // Menangani error jika ID yang dimasukkan bukan angka
170             JOptionPane.showMessageDialog(null, "ID harus berupa angka.", "Error", JOptionPane.ERROR_MESSAGE);
171         }
172     }
173     loadMahasiswaTable();
174 }
175
176 /**
177  * @param args the command line arguments
178  */
179 public static void main(String args[]) {
180     /* Set the Nimbus look and feel */
```

Output Sebelum Hapus

The screenshot shows a Java Swing window titled "MahasiswaView". It contains four text input fields labeled "NPM", "Nama", "Semester", and "IPK". The "NPM" field contains "5043856", "Nama" contains "Majella", "Semester" contains "3", and "IPK" contains "4.0". To the right of the "NPM" field is a "Simpan" button, and to the right of the "Semester" field is a "Buang" button. Below the input fields is a table with 5 columns: ID, NPM, Nama, Semester, and IPK. The table contains 6 rows of data.

ID	NPM	Nama	Semester	IPK
8	51443543	Bayu	7	3.72
9	5043234	Tirta	5	3.5
10	5043487	Ardi	5	3.3
11	5043823	Gerard	1	4.0
12	5043856	Majella	3	4.0

Pop Up Hapus

The screenshot shows a student management application window. At the top, there are input fields for NPM (5043856), Nama (Majella), Semester (3), and IPK (4.0), along with 'Simpan' and 'Buang' buttons. Below these is a table of student data. A 'Hapus Mahasiswa' dialog box is open, prompting the user to enter the ID of the student to be deleted. The ID '10' is entered in the input field.

ID	NPM	Nama	Semester	IPK
8	51443543	Bayu	7	3.72
9	5043234	Tirta	5	3.5
10	5043487	Ardi	5	3.3
11	5043823	Gerard	1	4.0
12	5043856	Majella	3	4.0

Hapus Mahasiswa

Masukkan ID yang ingin dihapus: 10

OK Cancel

Output Sesudah di Hapus

The screenshot shows the same student management application window after deleting student ID 10. The 'Hapus Mahasiswa' dialog box is no longer present. The table of student data now only contains four rows, with ID 10 removed.

ID	NPM	Nama	Semester	IPK
8	51443543	Bayu	7	3.72
9	5043234	Tirta	5	3.5
11	5043823	Gerard	1	4.0
12	5043856	Majella	3	4.0