

ACT PERTEMUAN 5

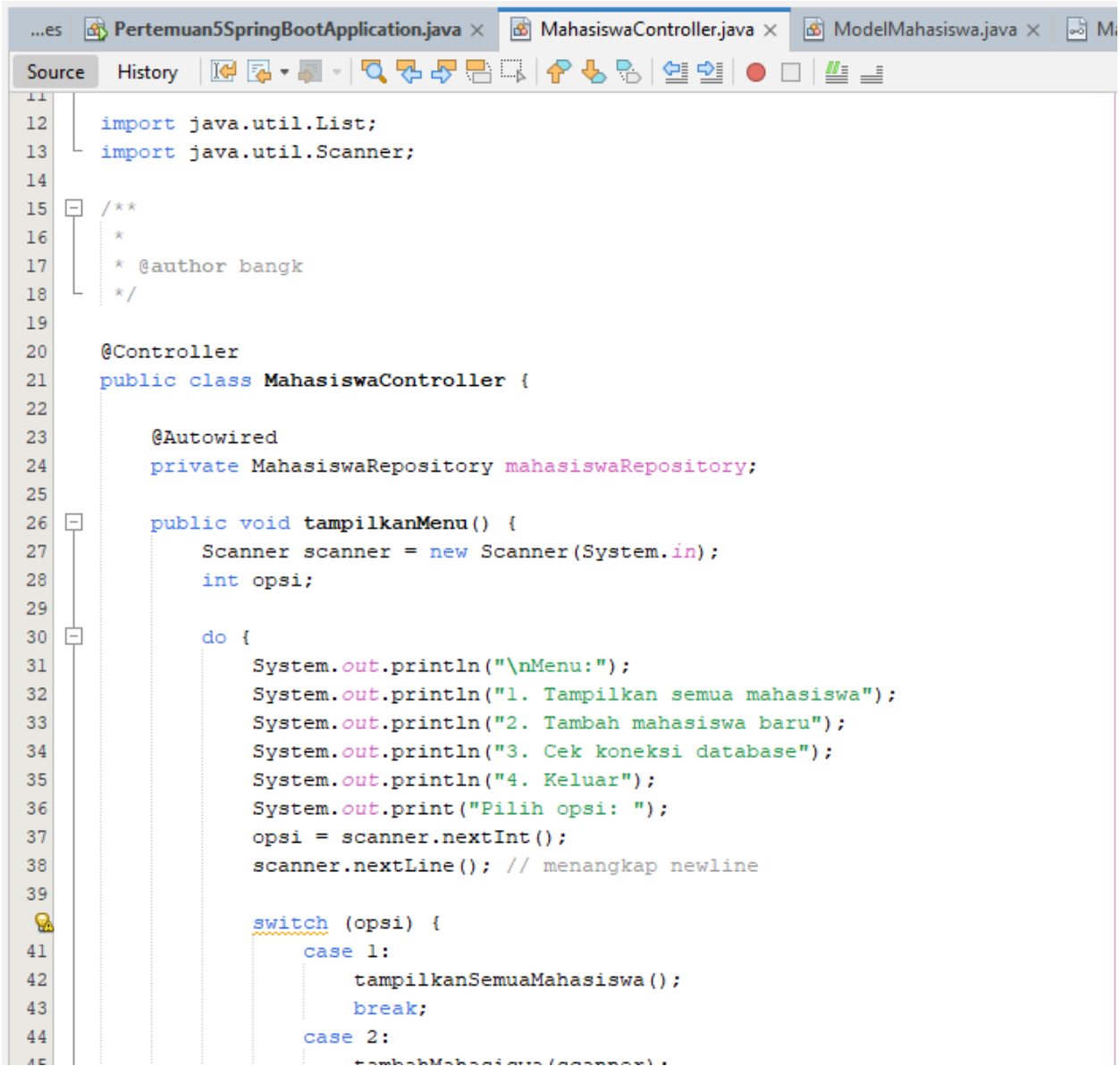
Nama : Muhamad Kukuh Fadhilah

NPM : 50421871

Kelas : 4IA28

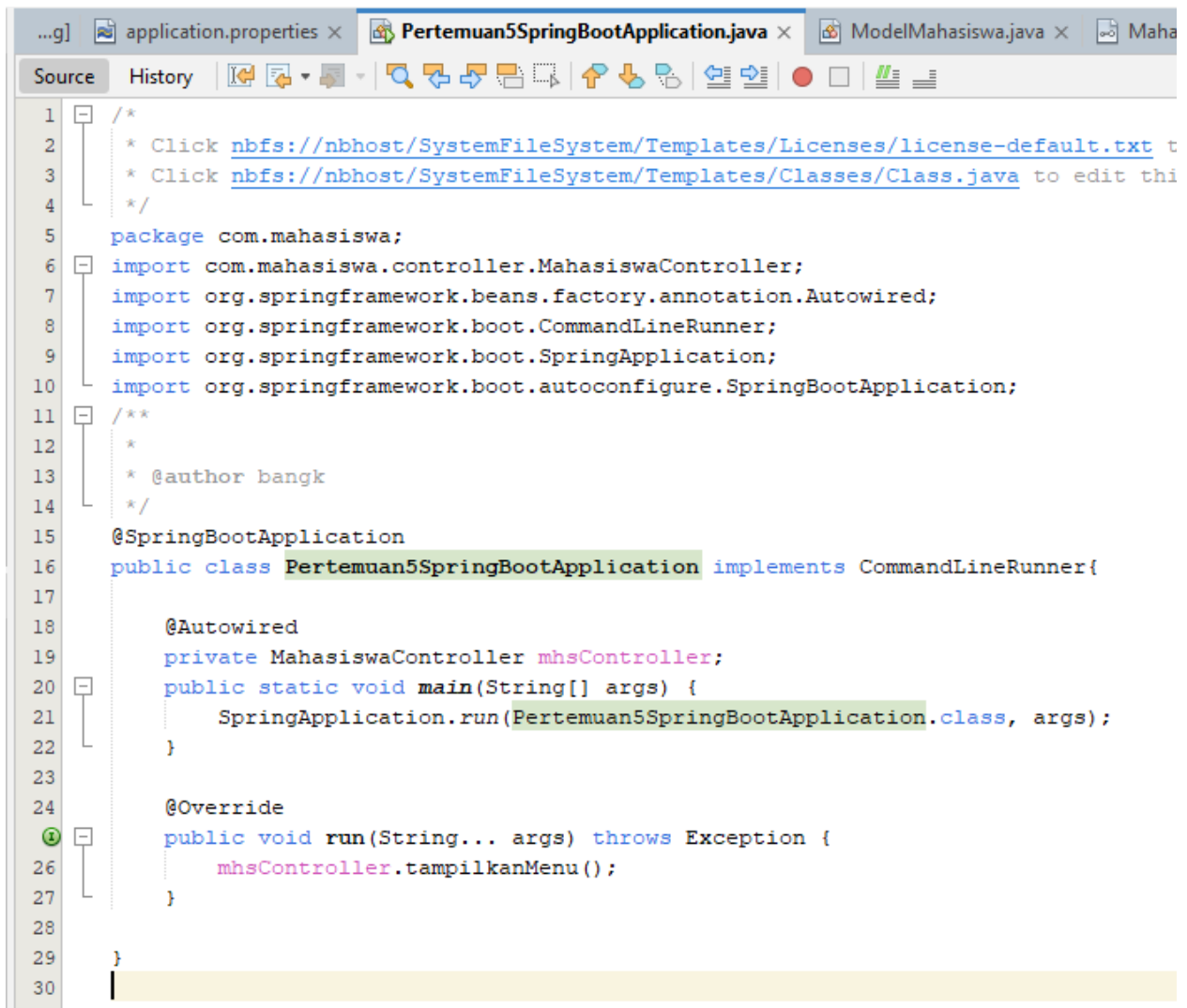
CODE

MahasiswaController



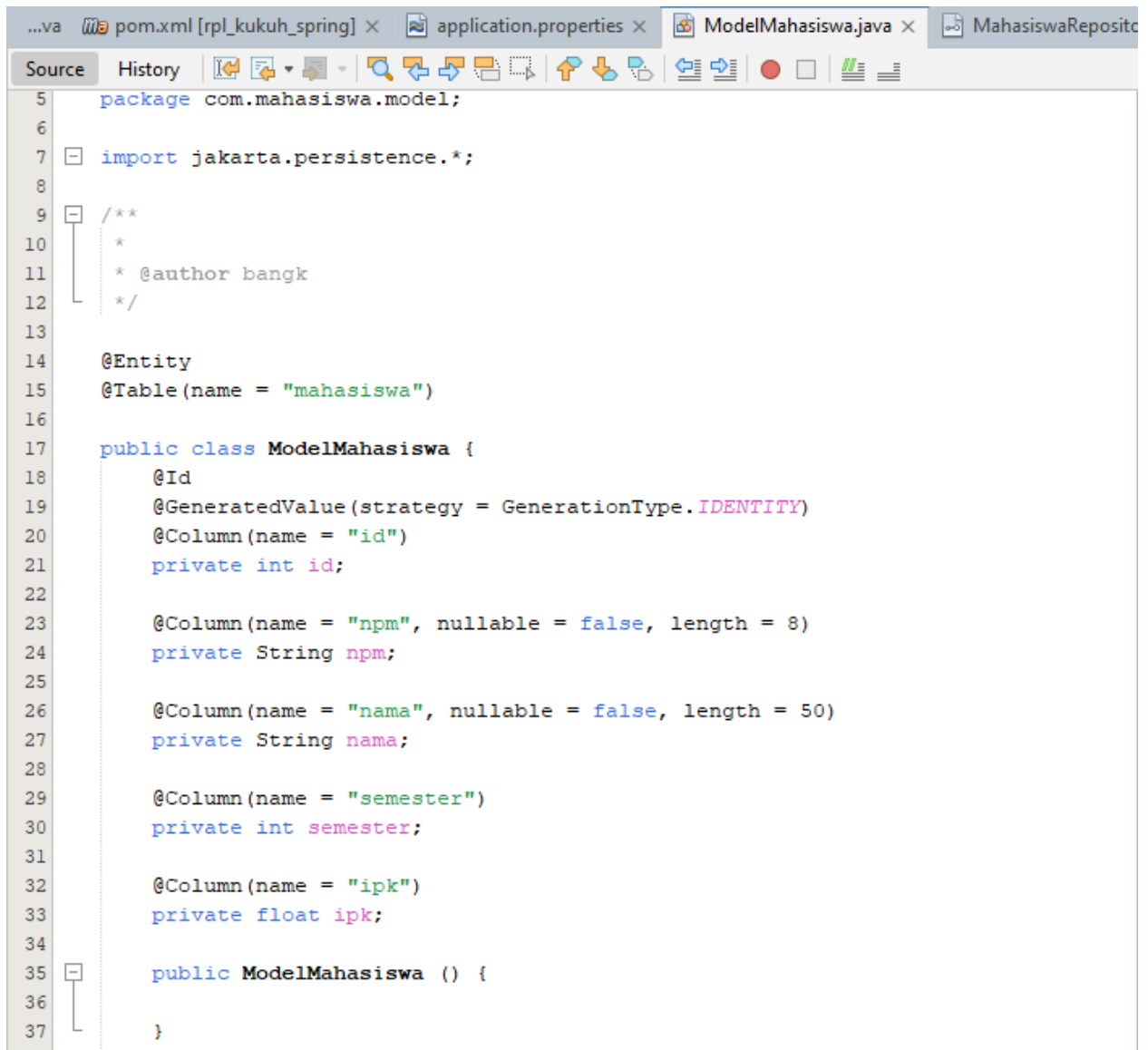
```
11
12 import java.util.List;
13 import java.util.Scanner;
14
15 /**
16  *
17  * @author bangk
18  */
19
20 @Controller
21 public class MahasiswaController {
22
23     @Autowired
24     private MahasiswaRepository mahasiswaRepository;
25
26     public void tampilanMenu() {
27         Scanner scanner = new Scanner(System.in);
28         int opsi;
29
30         do {
31             System.out.println("\nMenu:");
32             System.out.println("1. Tampilkan semua mahasiswa");
33             System.out.println("2. Tambah mahasiswa baru");
34             System.out.println("3. Cek koneksi database");
35             System.out.println("4. Keluar");
36             System.out.print("Pilih opsi: ");
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);
```

Pertemuan5SpringBootApplication



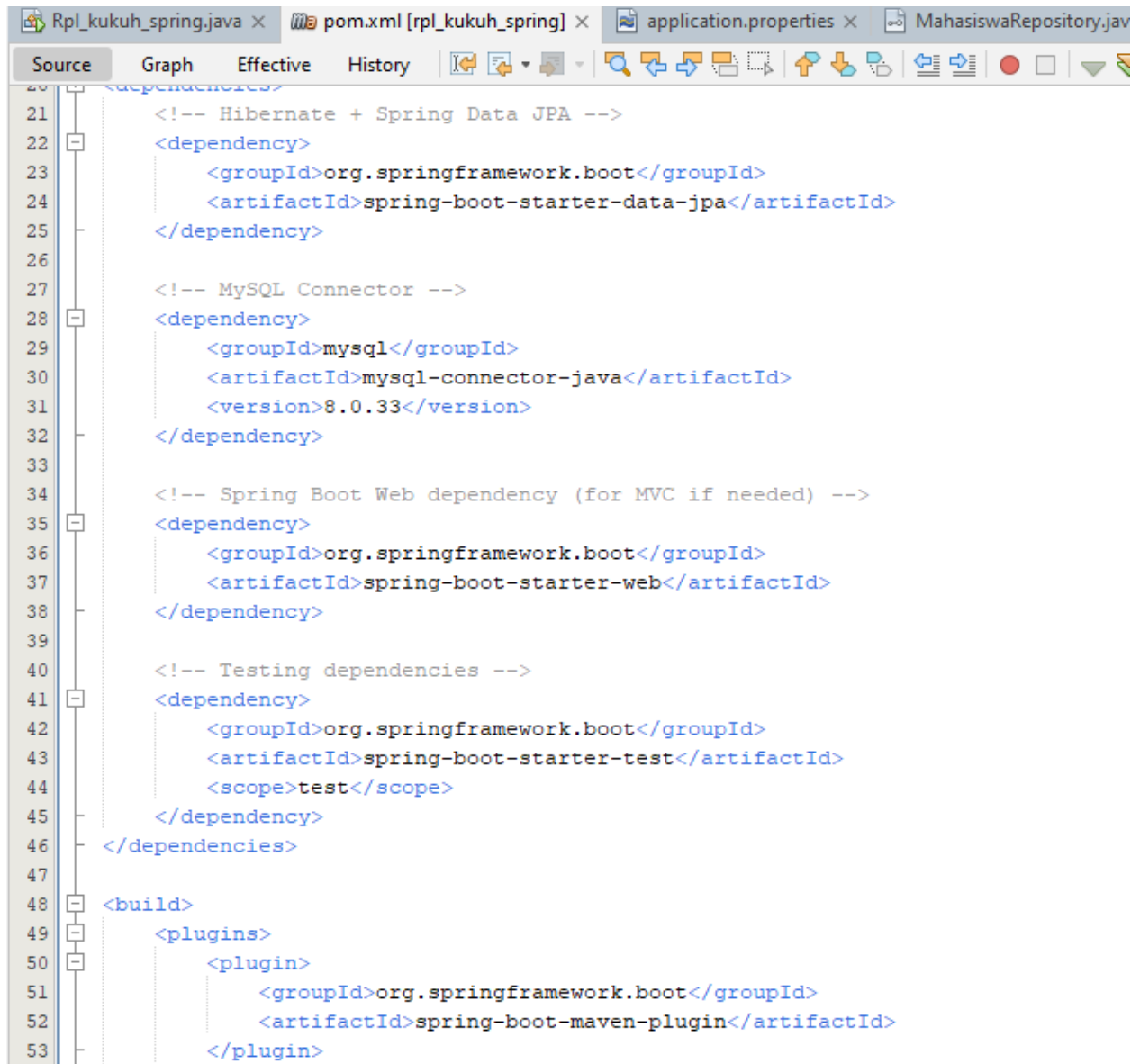
```
1  /*
2   * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to
3   * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this
4   */
5  package com.mahasiswa;
6  import com.mahasiswa.controller.MahasiswaController;
7  import org.springframework.beans.factory.annotation.Autowired;
8  import org.springframework.boot.CommandLineRunner;
9  import org.springframework.boot.SpringApplication;
10 import org.springframework.boot.autoconfigure.SpringBootApplication;
11 /**
12  *
13  * @author bangk
14  */
15 @SpringBootApplication
16 public class Pertemuan5SpringBootApplication implements CommandLineRunner{
17
18     @Autowired
19     private MahasiswaController mhsController;
20     public static void main(String[] args) {
21         SpringApplication.run(Pertemuan5SpringBootApplication.class, args);
22     }
23
24     @Override
25     public void run(String... args) throws Exception {
26         mhsController.tampilkanMenu();
27     }
28
29 }
30
```

ModelMahasiswa



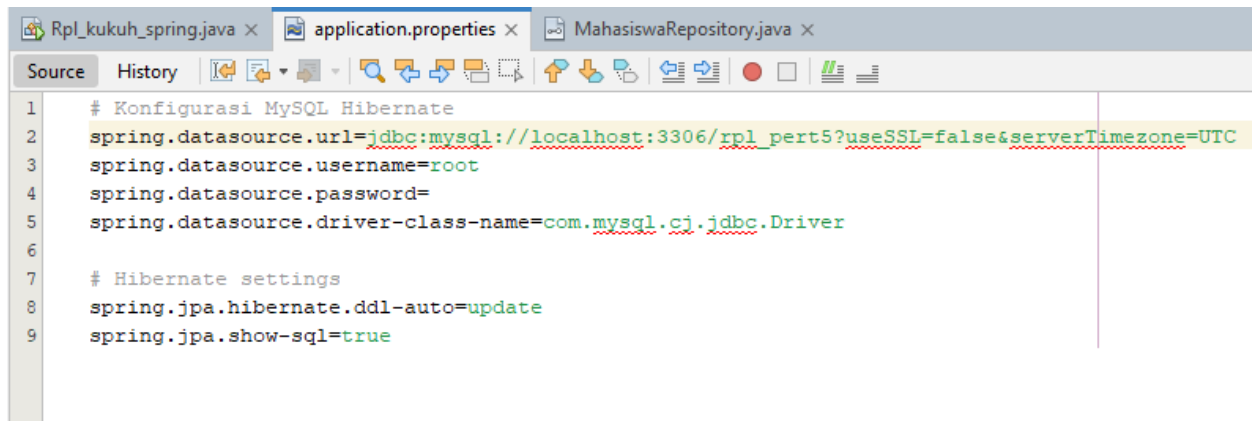
```
5 package com.mahasiswa.model;
6
7 import jakarta.persistence.*;
8
9 /**
10  *
11  * @author bangk
12  */
13
14 @Entity
15 @Table(name = "mahasiswa")
16
17 public class ModelMahasiswa {
18     @Id
19     @GeneratedValue(strategy = GenerationType.IDENTITY)
20     @Column(name = "id")
21     private int id;
22
23     @Column(name = "npm", nullable = false, length = 8)
24     private String npm;
25
26     @Column(name = "nama", nullable = false, length = 50)
27     private String nama;
28
29     @Column(name = "semester")
30     private int semester;
31
32     @Column(name = "ipk")
33     private float ipk;
34
35     public ModelMahasiswa () {
36
37     }
```

POMXML



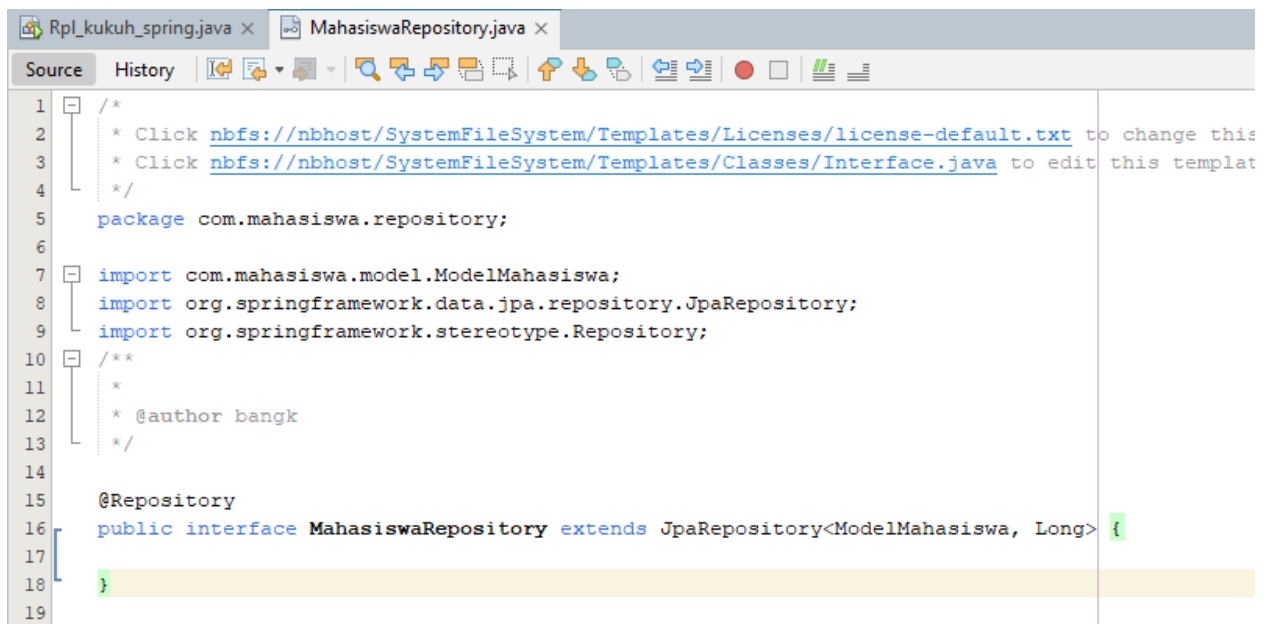
```
20 <dependencies>
21     <!-- Hibernate + Spring Data JPA -->
22     <dependency>
23         <groupId>org.springframework.boot</groupId>
24         <artifactId>spring-boot-starter-data-jpa</artifactId>
25     </dependency>
26
27     <!-- MySQL Connector -->
28     <dependency>
29         <groupId>mysql</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
40     <!-- Testing dependencies -->
41     <dependency>
42         <groupId>org.springframework.boot</groupId>
43         <artifactId>spring-boot-starter-test</artifactId>
44         <scope>test</scope>
45     </dependency>
46 </dependencies>
47
48 <build>
49     <plugins>
50         <plugin>
51             <groupId>org.springframework.boot</groupId>
52             <artifactId>spring-boot-maven-plugin</artifactId>
53         </plugin>
```

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
```

MahasiswaRepository



```
1 /*
2  * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this
3  * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Interface.java to edit this template
4  */
5 package com.mahasiswa.repository;
6
7 import com.mahasiswa.model.ModelMahasiswa;
8 import org.springframework.data.jpa.repository.JpaRepository;
9 import org.springframework.stereotype.Repository;
10
11 /**
12  * @author bangk
13  */
14
15 @Repository
16 public interface MahasiswaRepository extends JpaRepository<ModelMahasiswa, Long> {
17
18 }
19
```

OUTPUT

CEK DATA

```
Output - Run (Pertemuan5SpringBootApplication) x
2024-11-02T10:22:10.062+07:00 INFO 5524 --- [main] c.m.Pertemuan5SpringBootApplication
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
Tidak ada data mahasiswa.
```

TAMBAH DATA

```
Menu:
1. Tampilkan semua mahasiswa
2. Tambah mahasiswa baru
3. Cek koneksi database
4. Keluar
Pilih opsi: 2
Masukkan NPM : 50421871
Masukkan Nama : Kukuh
Masukkan Semester : 7
Masukkan IPK : 3,5
Hibernate: insert into mahasiswa (ipk,nama,npm,semester) values (?, ?, ?, ?)
Mahasiswa berhasil ditambahkan.
```

Output - Run (Pertemuan5SpringBootApplication) x

```
Menu:
1. Tampilkan semua mahasiswa
2. Tambah mahasiswa baru
3. Cek koneksi database
4. Keluar
Pilih opsi: 2
Masukkan NPM : 50421795
Masukkan Nama : Santo
Masukkan Semester : 5
Masukkan IPK : 2,9
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 : 11214598
Masukkan Nama : Sintaksis
Masukkan Semester : 4
Masukkan IPK : 3,2
Hibernate: insert into mahasiswa (ipk,nama,npm,semester) values (?, ?, ?, ?)
Mahasiswa berhasil ditambahkan.
```

TAMPILKAN DATA

```
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='50421871', nama='Kukuh', nama='7', jurusan='3.5'}
Mahasiswa{id=2, nama='50421795', nama='Santo', nama='5', jurusan='2.9'}
Mahasiswa{id=3, nama='11214598', nama='Sintaksis', nama='4', jurusan='3.2'}
```

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.
```

KELUAR PROGRAM

Menu:

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

Pilih opsi: 4

Keluar dari program.

|

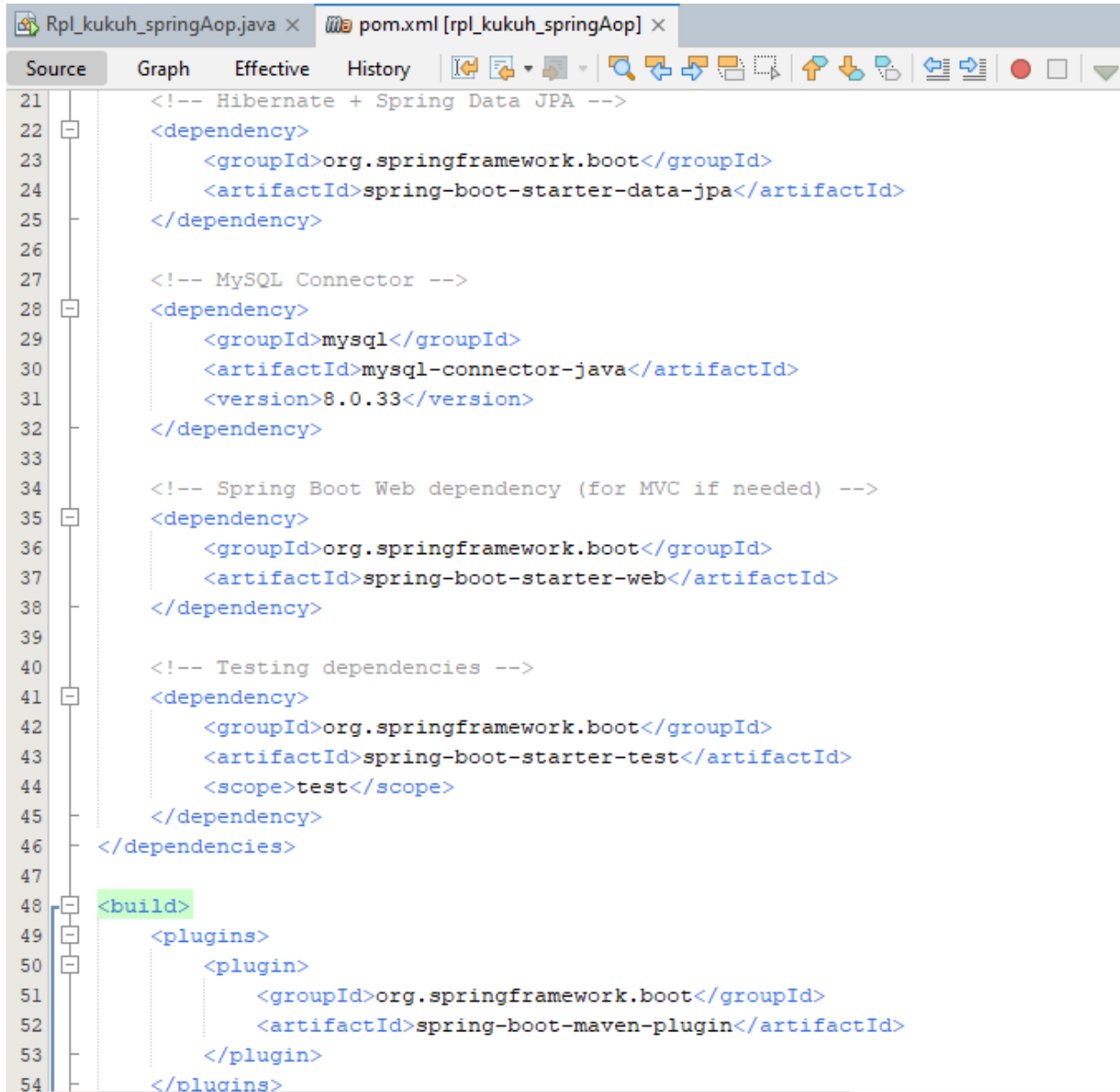
ACT PERTEMUAN 6

Nama : Muhamad Kukuh Fadhilah

NPM : 50421871

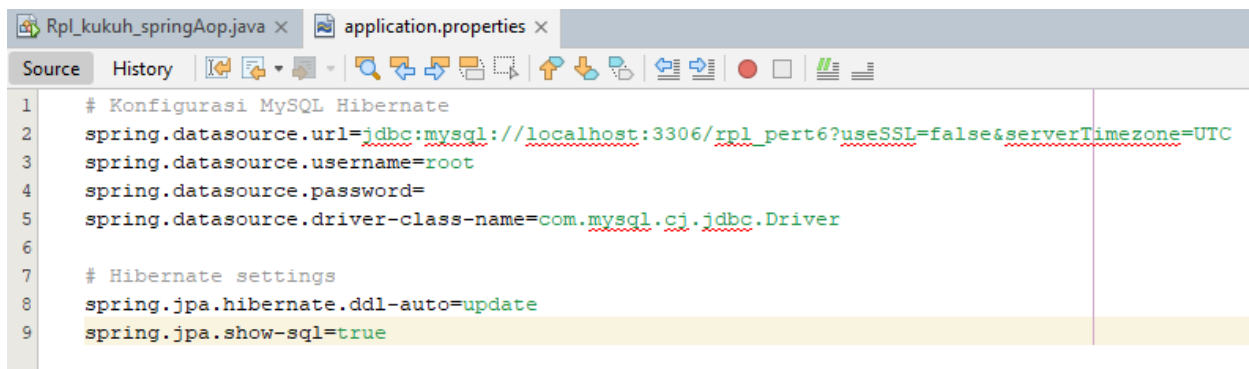
Kelas : 4IA28

Pom.xml



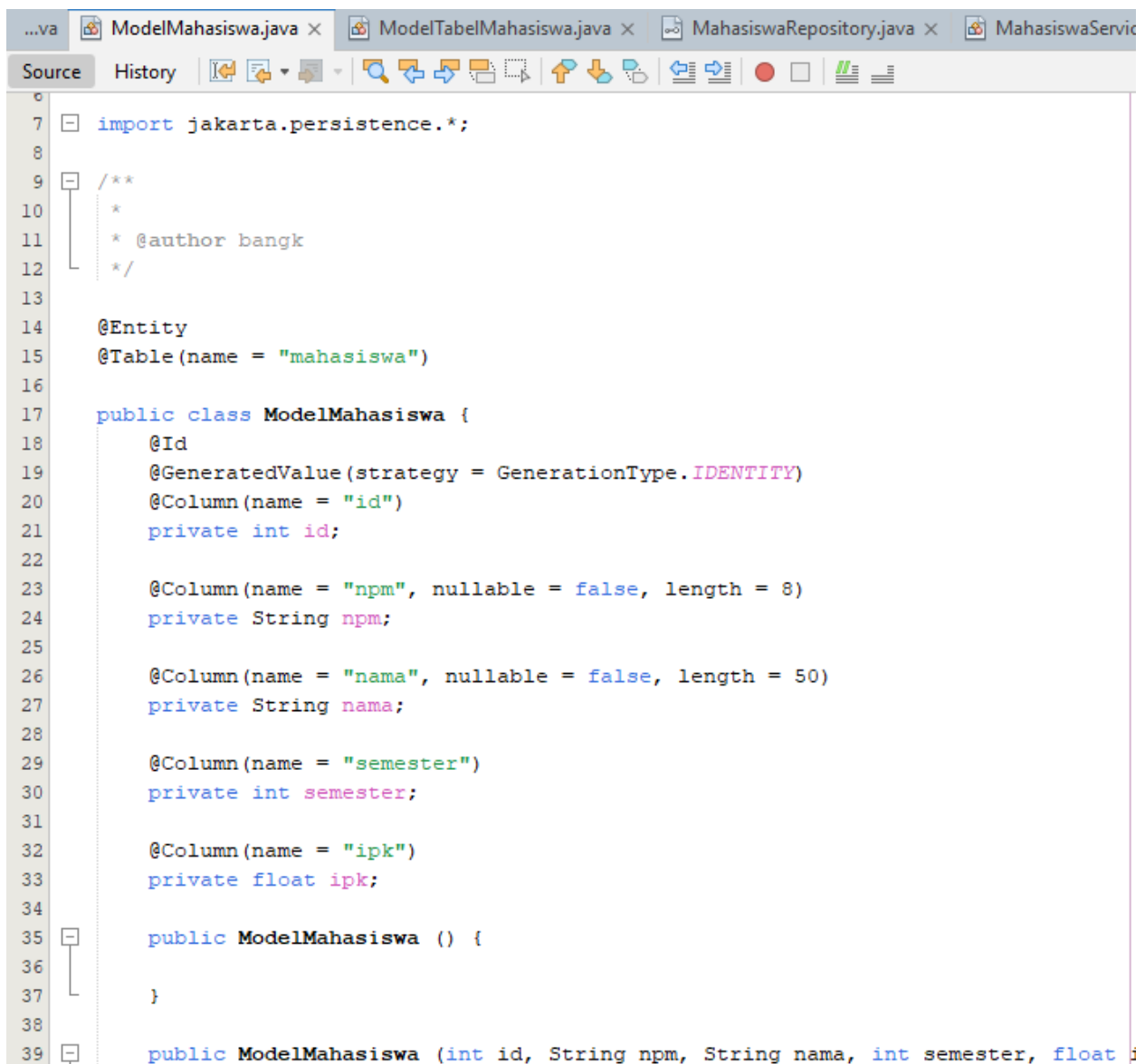
```
21      <!-- Hibernate + Spring Data JPA -->
22      <dependency>
23          <groupId>org.springframework.boot</groupId>
24          <artifactId>spring-boot-starter-data-jpa</artifactId>
25      </dependency>
26
27      <!-- MySQL Connector -->
28      <dependency>
29          <groupId>mysql</groupId>
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
40      <!-- Testing dependencies -->
41      <dependency>
42          <groupId>org.springframework.boot</groupId>
43          <artifactId>spring-boot-starter-test</artifactId>
44          <scope>test</scope>
45      </dependency>
46  </dependencies>
47
48  <build>
49      <plugins>
50          <plugin>
51              <groupId>org.springframework.boot</groupId>
52              <artifactId>spring-boot-maven-plugin</artifactId>
53          </plugin>
54      </plugins>
```

Application.properties

A screenshot of an IDE window titled 'application.properties'. The window has a toolbar with various icons for editing and navigation. The code is as follows:

```
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
```

ModelMahasiswa

A screenshot of an IDE window showing the 'ModelMahasiswa.java' file. The window has a toolbar with various icons for editing and navigation. The code is as follows:

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

ModelTabelMahasiswa.java

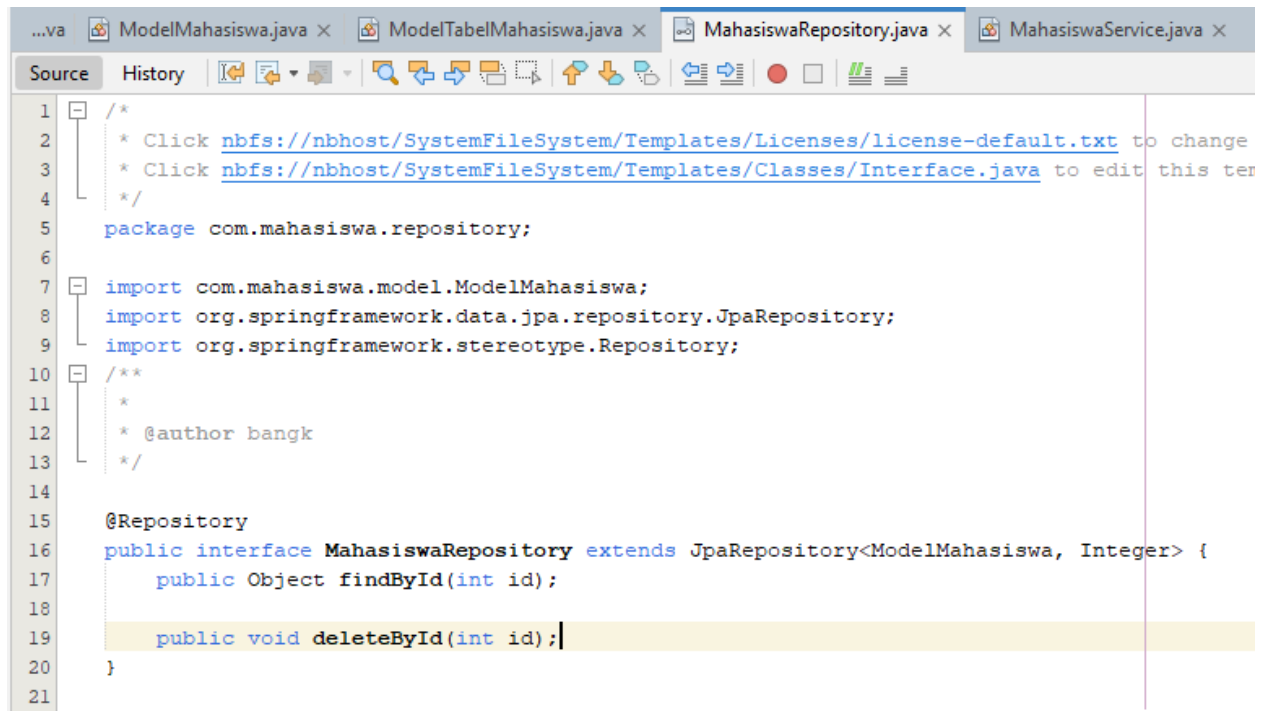
```
...va ModelMahasiswa.java x ModelTabelMahasiswa.java x MahasiswaRepository.java x MahasiswaSen

Source History

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

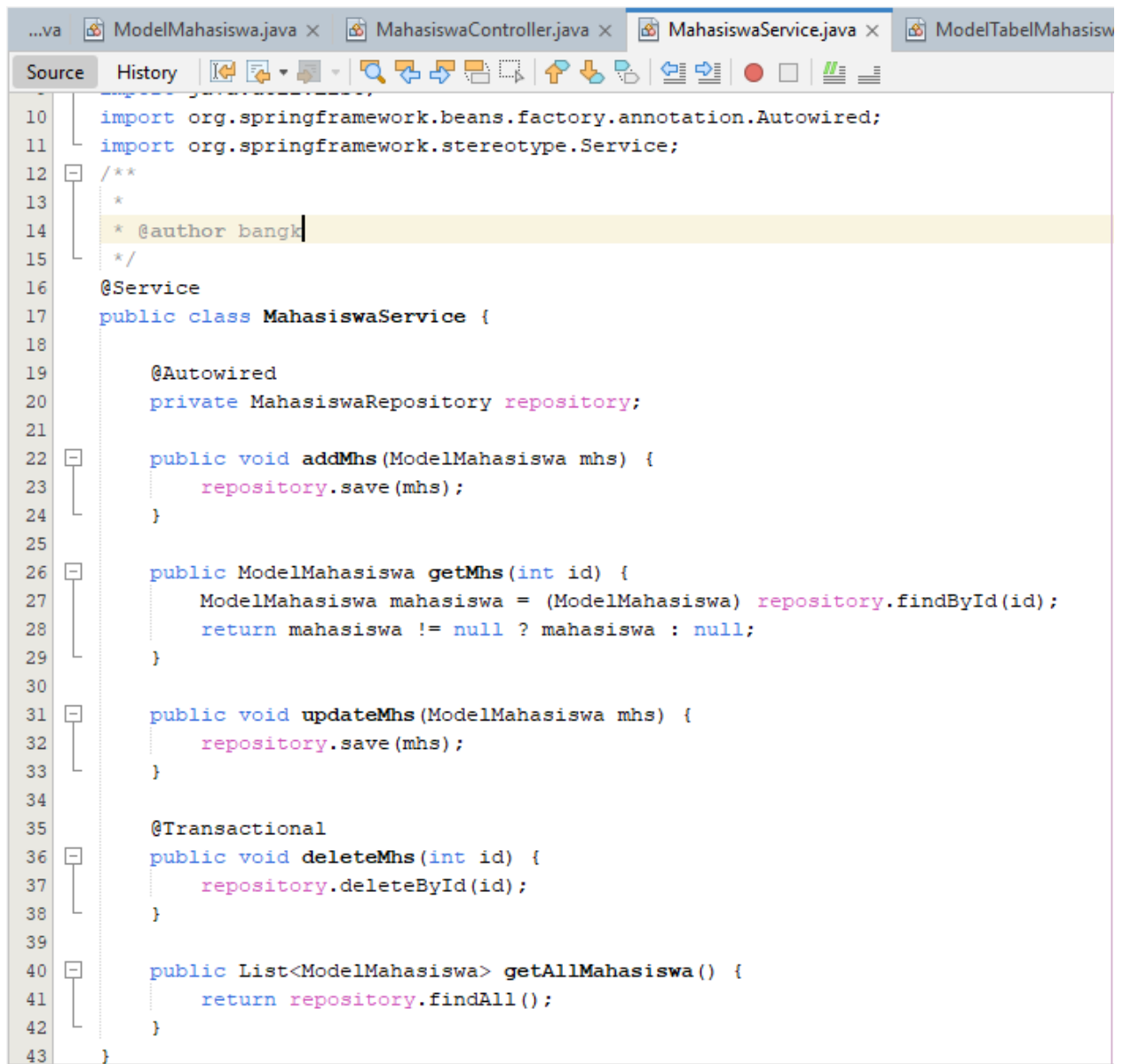
```

MahasiswaRepository



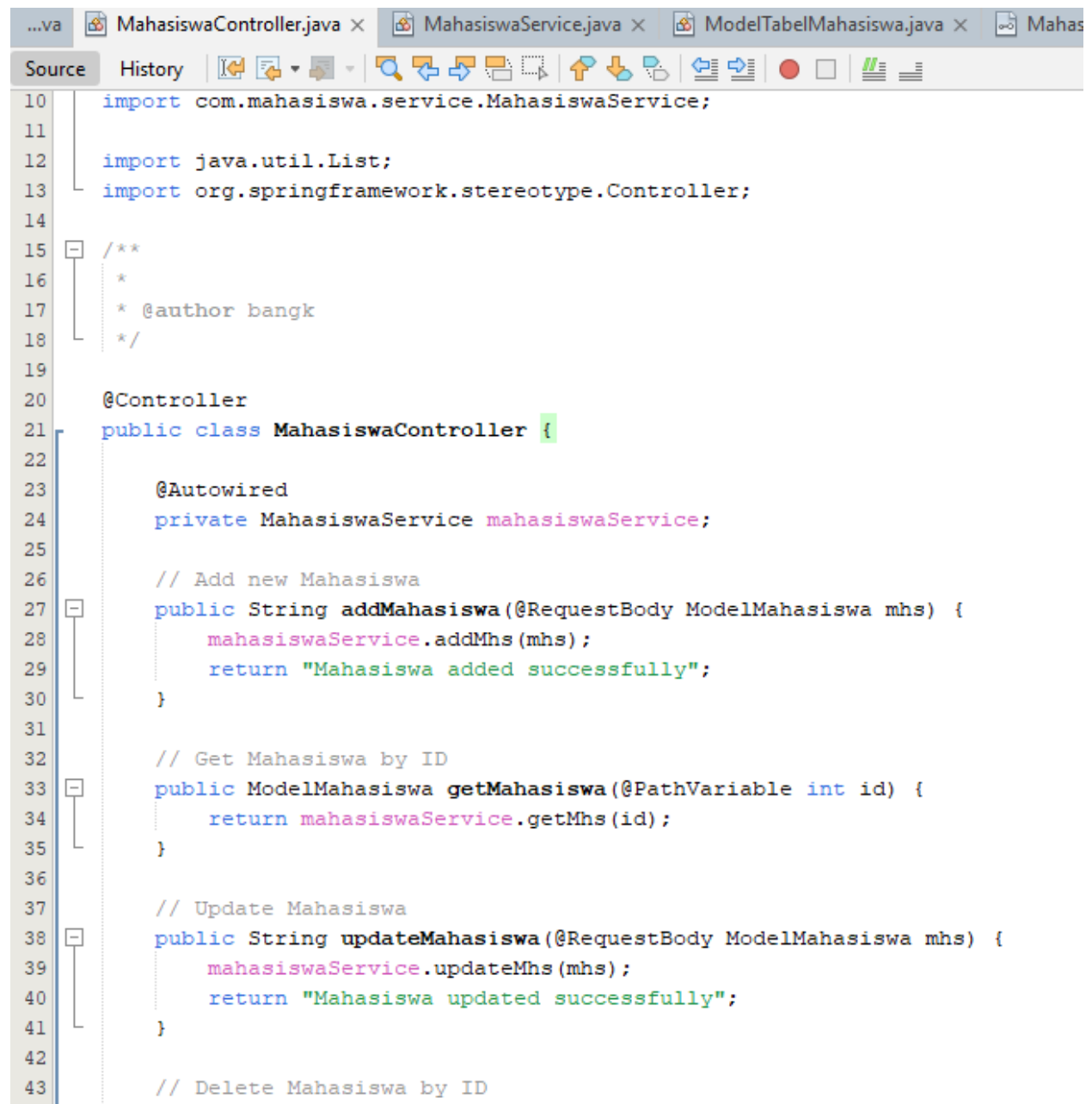
```
1  /*
2   * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change
3   * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Interface.java to edit this template
4   */
5   package com.mahasiswa.repository;
6
7   import com.mahasiswa.model.ModelMahasiswa;
8   import org.springframework.data.jpa.repository.JpaRepository;
9   import org.springframework.stereotype.Repository;
10
11  /**
12   *
13   * @author bangk
14   */
15  @Repository
16  public interface MahasiswaRepository extends JpaRepository<ModelMahasiswa, Integer> {
17      public Object findById(int id);
18
19      public void deleteById(int id);
20  }
21
```

MahasiswaService



```
10  import org.springframework.beans.factory.annotation.Autowired;
11  import org.springframework.stereotype.Service;
12  /**
13   *
14   * @author bangk
15   */
16  @Service
17  public class MahasiswaService {
18
19      @Autowired
20      private MahasiswaRepository repository;
21
22      public void addMhs(ModelMahasiswa mhs) {
23          repository.save(mhs);
24      }
25
26      public ModelMahasiswa getMhs(int id) {
27          ModelMahasiswa mahasiswa = (ModelMahasiswa) repository.findById(id);
28          return mahasiswa != null ? mahasiswa : null;
29      }
30
31      public void updateMhs(ModelMahasiswa mhs) {
32          repository.save(mhs);
33      }
34
35      @Transactional
36      public void deleteMhs(int id) {
37          repository.deleteById(id);
38      }
39
40      public List<ModelMahasiswa> getAllMahasiswa() {
41          return repository.findAll();
42      }
43  }
```

MahasiswaController



```
10 import com.mahasiswa.service.MahasiswaService;
11
12 import java.util.List;
13 import org.springframework.stereotype.Controller;
14
15 /**
16  *
17  * @author bangk
18  */
19
20 @Controller
21 public class MahasiswaController {
22
23     @Autowired
24     private MahasiswaService mahasiswaService;
25
26     // Add new Mahasiswa
27     public String addMahasiswa(@RequestBody ModelMahasiswa mhs) {
28         mahasiswaService.addMhs(mhs);
29         return "Mahasiswa added successfully";
30     }
31
32     // Get Mahasiswa by ID
33     public ModelMahasiswa getMahasiswa(@PathVariable int id) {
34         return mahasiswaService.getMhs(id);
35     }
36
37     // Update Mahasiswa
38     public String updateMahasiswa(@RequestBody ModelMahasiswa mhs) {
39         mahasiswaService.updateMhs(mhs);
40         return "Mahasiswa updated successfully";
41     }
42
43     // Delete Mahasiswa by ID
```

MahasiswaView.java

```
...va  MahasiswaService.java x  ModelTabelMahasiswa.java x  MahasiswaRepository.java x  MahasiswaView.java x
Source  Design  History  [Icons]
13  /**
14  *
15  * @author bangk
16  */
17  public class MahasiswaView extends javax.swing.JFrame {
18      private MahasiswaController controller;
19      /**
20       * Creates new form MahasiswaView
21       */
22      public MahasiswaView(MahasiswaController controller) {
23          this.controller = controller;
24          initComponents();
25          loadMahasiswaTable();
26      }
27
28      private MahasiswaView() {
29          throw new UnsupportedOperationException("Not supported yet."); // Generated f
30      }
31
32      public void loadMahasiswaTable() {
33          // Ambil data dari controller
34          List<ModelMahasiswa> listMahasiswa = controller.getAllMahasiswa();
35
36          // Buat model tabel kustom dengan data mahasiswa
37          ModelTabelMahasiswa tableModel = new ModelTabelMahasiswa(listMahasiswa);
38
39          // Set model pada JTable
40          dataTable.setModel(tableModel);
41      }
42
43      /**
44       * This method is called from within the constructor to initialize the form.
45       * WARNING: Do NOT modify this code. The content of this method is always
```

```
...va MahasiswaService.java x MahasiswaView.java x ModelTabelMahasiswa.java x MahasiswaRepository.java x
Source Design History
166 private void simpanButtonActionPerformed(java.awt.event.ActionEvent evt) {
167     // TODO add your handling code here:
168     String npm = getNpmField().getText();
169     String nama = getNamaField().getText();
170     int semester = Integer.parseInt(getSemesterField().getText());
171     float ipk = Float.parseFloat(getIpkField().getText());
172     ModelMahasiswa mahasiswa = new ModelMahasiswa(0, npm, nama, semester, ipk);
173     System.out.println(mahasiswa.getIpk());
174     System.out.println(mahasiswa.getNama());
175     System.out.println(mahasiswa.getSemester());
176     System.out.println(mahasiswa.getNpm());
177
178     controller.addMahasiswa(mahasiswa);
179     loadMahasiswaTable();
180 }
181
```

```
...va MahasiswaService.java x MahasiswaView.java x ModelTabelMahasiswa.java x MahasiswaRepository.java x
Source Design History
181
182 private void buangButtonActionPerformed(java.awt.event.ActionEvent evt) {
183     // TODO add your handling code here:
184     JTextField idField = new JTextField(5);
185
186     // Membuat panel untuk menampung JTextField
187     JPanel panel = new JPanel();
188     panel.add(new JLabel("Masukkan ID yang ingin dihapus:"));
189     panel.add(idField);
190
191     // Menampilkan dialog box dengan JTextField, tombol OK, dan Cancel
192     int result = JOptionPane.showConfirmDialog(null, panel,
193         "Hapus Mahasiswa", JOptionPane.OK_CANCEL_OPTION, JOptionPane.PLAIN_MESSAGE);
194
195     // Jika tombol OK ditekan
196     if (result == JOptionPane.OK_OPTION) {
197         try {
198             // Mengambil input ID dan memanggil metode deleteMhs
199             int id = Integer.parseInt(idField.getText());
200             controller.deleteMahasiswa(id);
201             JOptionPane.showMessageDialog(null, "Data berhasil dihapus.", "Sukses",
202                 JOptionPane.INFORMATION_MESSAGE);
203         } catch (NumberFormatException e) {
204             // Menangani error jika ID yang dimasukkan bukan angka
205             JOptionPane.showMessageDialog(null, "ID harus berupa angka.", "Error",
206                 JOptionPane.ERROR_MESSAGE);
207         }
208         loadMahasiswaTable();
209     }
210
211     public JTextField getIpkField() {
212         return ipkField;
213     }
214
215     public void setIpkField(JTextField ipkField) {
216
217     }
218 }
219
```


Design

...va

MahasiswaService.java ×

MahasiswaView.java ×

ModelTabelMahasiswa.java...

Source

Design

History

To select multiple components in an area hold Shift and drag mouse over the components.

NPM

Simpan

Nama

Buang


Semester

IPK

Title 1	Title 2	Title 3	Title 4

OUTPUT

Simpan Data

 — □ ×

NPM

50421871

Nama

Kukuh

Semester

7


IPK

3.5

Simpan

Buang

ID	NPM	Nama	Semester	IPK
1	50421871	Kukuh	7	3.5

 — □ ×

NPM

51422672

Nama

Sueb

Semester

4


IPK

2.9

Simpan

Buang

ID	NPM	Nama	Semester	IPK
1	50421871	Kukuh	7	3.5
2	51422672	Sueb	4	2.9

 — □ ×

NPM

51427612

Nama

Santo

Semester

5


IPK

3.0

Simpan

Buang

ID	NPM	Nama	Semester	IPK
1	50421871	Kukuh	7	3.5
2	51422672	Sueb	4	2.9
3	51427612	Santo	5	3.0

 — □ ×

NPM

11223344

Nama

ilmu

Semester

6


IPK

3.2

Simpan

Buang

ID	NPM	Nama	Semester	IPK
1	50421871	Kukuh	7	3.5
2	51422672	Sueb	4	2.9
3	51427612	Santo	5	3.0
4	11223344	ilmu	6	3.2

 — □ ×

NPM

11224562

Nama

sintak

Semester

2

IPK

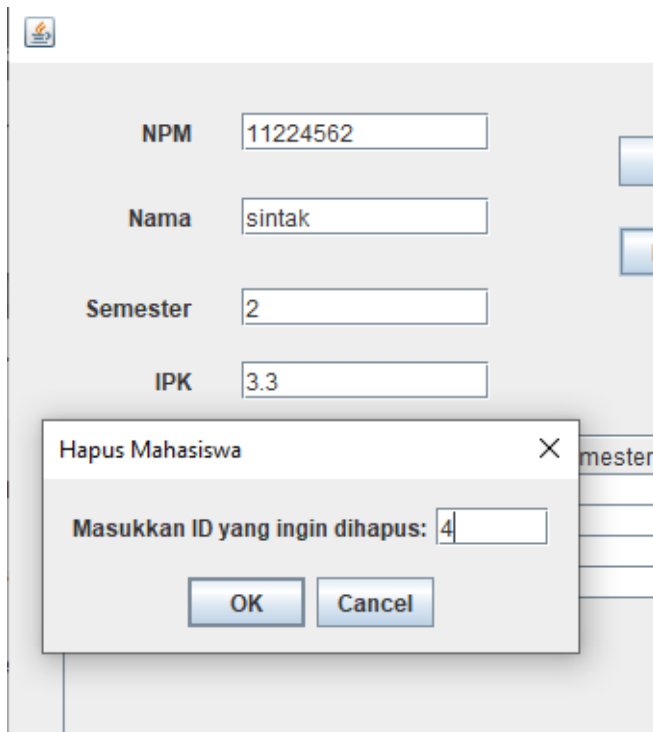
3.3

Simpan

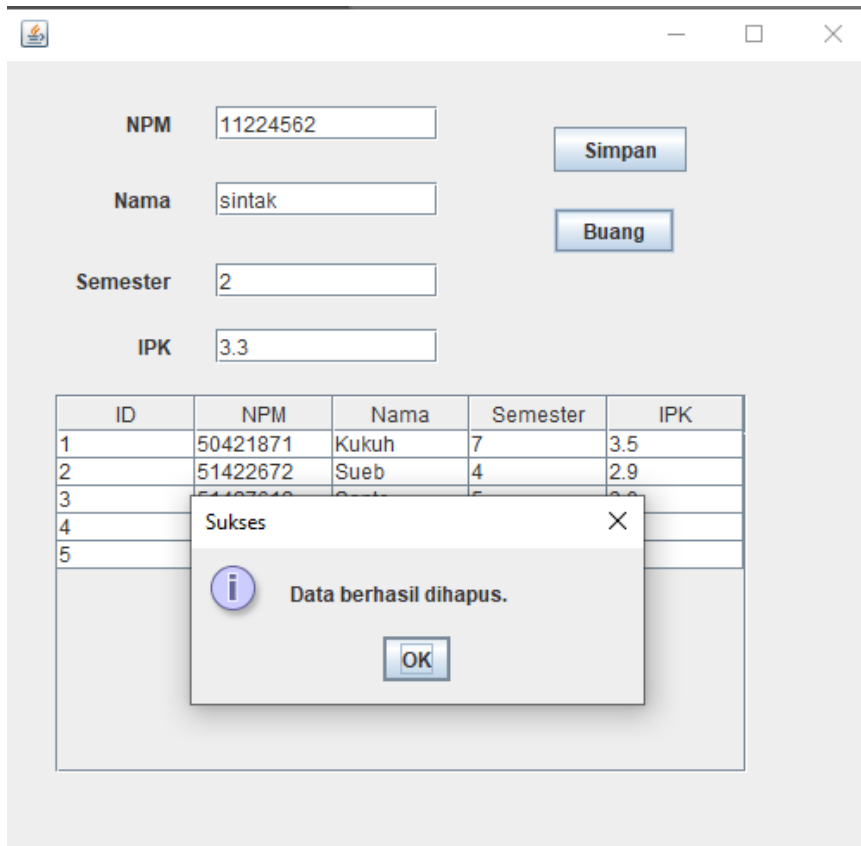
Buang

ID	NPM	Nama	Semester	IPK
1	50421871	Kukuh	7	3.5
2	51422672	Sueb	4	2.9
3	51427612	Santo	5	3.0
4	11223344	ilmu	6	3.2
5	11224562	sintak	2	3.3

Buang Data



A screenshot of a web application for entering student data. The form has four input fields: NPM (11224562), Nama (sintak), Semester (2), and IPK (3.3). To the right of the form are two buttons labeled 'S' and 'E'. A modal dialog titled 'Hapus Mahasiswa' is open in the foreground, asking 'Masukkan ID yang ingin dihapus:' with the number '4' entered in the input field. The dialog has 'OK' and 'Cancel' buttons.



A screenshot of the same web application, but now showing a table of student data and a success message. The form fields are the same. To the right of the form are two buttons labeled 'Simpan' and 'Buang'. Below the form is a table with 5 rows and 5 columns: ID, NPM, Nama, Semester, and IPK. A modal dialog titled 'Sukses' is open in the foreground, displaying an information icon and the text 'Data berhasil dihapus.' with an 'OK' button.

ID	NPM	Nama	Semester	IPK
1	50421871	Kukuh	7	3.5
2	51422672	Sueb	4	2.9
3	51422672	Sueb	4	2.9
4				
5				