

ACTIVITY PERTEMUAN 6

NAMA : Hafizhan Noor Amril

NPM : 50422639

KELAS : 4IA16

MATERI : Implementasi AOP dan Dependency Injection Pada Project Spring dan Hibernate

MATA PRAKTIKUM : Rekayasa Perangkat Lunak 2 (RPL 2)

Soal :

1. Screenshoot kode dan output dari program yang telah dibuat

Jawab :

pom.xml

```
1  [INFO] pom.xml (pert6_50422637) | PertemanSpringBootApplication.java | Pert6_50422639.java | MahasiswaController.java | ModelMahasiswa.java | MahasiswaRepository.java | MahasiswaService.java | MahasiswaView.java | ModelTabMahasiswa.java
2
3  <project>
4      <modelVersion>4.0.0</modelVersion>
5      <groupId>com.perteman.pert6</groupId>
6      <artifactId>spring-boot-starter-parent</artifactId>
7      <version>1.0-SNAPSHOT</version>
8      <packaging>jar</packaging>
9
10     <parent>
11         <projectBuildSourceEncoding>UTF-8</projectBuildSourceEncoding>
12         <mavenCompilerSource>1.8</mavenCompilerSource>
13         <mavenCompilerTarget>1.8</mavenCompilerTarget>
14         <execMainClass>com.mycompany.pert6_50422639.Pert6_50422639.mainClass</execMainClass>
15     </parent>
16
17     <groupId>org.springframework.boot</groupId>
18     <artifactId>spring-boot-starter-parent</artifactId>
19     <version>1.3.9</version>
20     <relativePath/>
21 </parent>
22
23     <dependencies>
24         <!-- Allgemein + Spring Data JPA -->
25         <dependency>
26             <groupId>org.springframework.boot</groupId>
27             <artifactId>spring-boot-starter-data-jpa</artifactId>
28         </dependency>
29
30         <!-- MySQL Connector -->
31         <dependency>
32             <groupId>mysql:mysql</groupId>
33             <artifactId>mysql-connector-java</artifactId>
34             <version>8.0.33</version>
35         </dependency>
36
37         <!-- Spring Boot Web dependency (für MVC if needed) -->
38         <dependency>
39             <groupId>org.springframework.boot</groupId>
40             <artifactId>spring-boot-starter-web</artifactId>
41         </dependency>
42
43         <!-- Testing dependencies -->
44         <dependency>
45             <groupId>org.springframework.boot</groupId>
46             <artifactId>spring-boot-starter-test</artifactId>
47             <scope>test</scope>
48         </dependency>
49     </dependencies>
50
51     <build>
52         <plugins>
53             <plugin>
54                 <groupId>org.springframework.boot</groupId>
55                 <artifactId>spring-boot-maven-plugin</artifactId>
56                 <configuration>
57                     <mainClass>com.perteman.MahasiswaApp.mainClass</mainClass>
58                 </configuration>
59             </plugin>
60         </plugins>
61     </build>
62
63 </project>
```

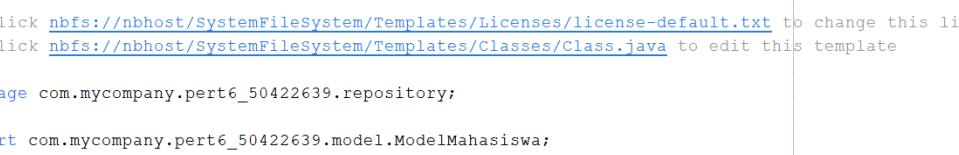
ModelMahasiswa.java

```
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.mycompany.perit_50422639.model;
6
7  import jakarta.persistence.Column;
8  import jakarta.persistence.Entity;
9  import jakarta.persistence.GeneratedValue;
10 import jakarta.persistence.GenerationType;
11 import jakarta.persistence.Id;
12 import jakarta.persistence.Table;
13
14 /**
15  * @author USER
16  */
17 @Entity
18 @Table(name = "mahasiswa")
19 public class ModelMahasiswa {
20
21
22     @Id
23     @GeneratedValue(strategy = GenerationType.IDENTITY)
24     @Column(name = "id")
25     private int id;
26
27     @Column(name = "npm", nullable = false, length = 8)
28     private String npm;
29
30     @Column(name = "nama", nullable = false, length = 50)
31     private String nama;
32
33     @Column(name = "semester")
34     private int semester;
35
36     @Column(name = "ipk")
37     private float ipk;
38
39     // constructor kosong untuk ORM
40     public ModelMahasiswa() {
41
42     }
43
44     //constructor ada isinya untuk dipanggil di aplikasi
45     public ModelMahasiswa(int id, String npm, String nama, int semester, float ipk) {
46         this.id = id;
47         this.npm = npm;
48         this.nama = nama;
49         this.semester = semester;
50         this.ipk = ipk;
51     }
52
53     public int getId() {
54         return id;
55     }
56
57     public void setId(int id) {
58         this.id = id;
59     }
60
61     public String getNpm() {
62         return npm;
63     }
64
65     public void setNpm(String npm) {
66         this.npm = npm;
67     }
68
69     public String getNama() {
70         return nama;
71     }
72
73     public void setNama(String nama) {
74         this.nama = nama;
75     }
76
77     public int getSemester() {
78         return semester;
79     }
80
81     public void setSemester(int semester) {
82         this.semester = semester;
83     }
84
85     public float getIpk() {
86         return ipk;
87     }
88
89     public void setIpk(float ipk) {
90         this.ipk = ipk;
91     }
92 }
```

ModelTabelMahasiswa.java

```
7 import java.util.List;
8 import javax.swing.table.AbstractTableModel;
9
10 /**
11 * @author USER
12 */
13 public class ModelTabelMahasiswa extends AbstractTableModel{
14     private List<ModelMahasiswa> mahasiswaList;
15     private final String[] columnNames = {"ID", "NPM", "Nama", "Semester", "IPK"};
16
17     public ModelTabelMahasiswa(List<ModelMahasiswa> mahasiswaList) {
18         this.mahasiswaList = mahasiswaList;
19     }
20
21
22     @Override
23     public int getRowCount() {
24         return mahasiswaList.size(); // Jumlah baris sesuai dengan jumlah data mahasiswa
25     }
26
27     @Override
28     public int getColumnCount() {
29         return columnNames.length; // Jumlah kolom sesuai dengan jumlah elemen dalam columnNames
30     }
31
32     @Override
33     public Object getValueAt(int rowIndex, int columnIndex) {
34         ModelMahasiswa mahasiswa = mahasiswaList.get(rowIndex);
35         switch (columnIndex) {
36             case 0:
37                 return mahasiswa.getId();
38             case 1:
39                 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:
38                 return null;
39         }
40     }
41
42     @Override
43     public String getColumnName(int column) {
44         return columnNames[column]; // Mengatur nama kolom
45     }
46
47     @Override
48     public boolean isCellEditable(int rowIndex, int columnIndex) {
49         return false; // Semua sel tidak dapat diedit
50     }
51
52     // Method untuk menambahkan atau memodifikasi data, jika dibutuhkan
53     public void setMahasiswaList(List<ModelMahasiswa> mahasiswaList) {
54         this.mahasiswaList = mahasiswaList;
55         fireTableDataChanged(); // Memberitahu JTable bahwa data telah berubah
56     }
57
58 }
```

MahasiswaRepository.java



The screenshot shows the Java code for the `MahasiswaRepository`. The code is annotated with comments explaining the template file used and how to change the license. It imports the `ModelMahasiswa` class and the `JpaRepository` interface from Spring Data JPA. The repository is annotated with `@Repository` and implements the `JpaRepository` interface with type parameters `ModelMahasiswa` and `Integer`.

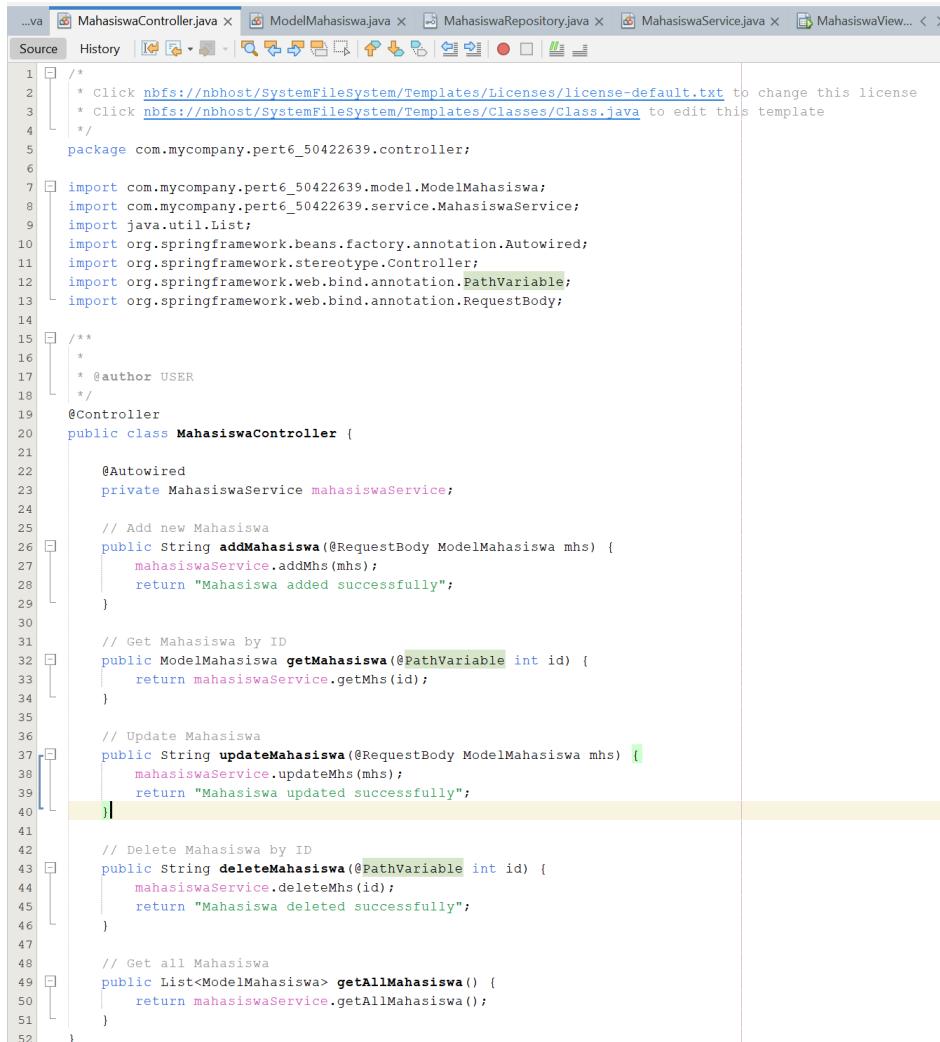
```
/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
 * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this template
 */
package com.mycompany.pert6_50422639.repository;

import com.mycompany.pert6_50422639.model.ModelMahasiswa;
import org.springframework.data.jpa.repository.JpaRepository;
import org.springframework.stereotype.Repository;

/**
 *
 * @author USER
 */
@Repository
public interface MahasiswaRepository extends JpaRepository<ModelMahasiswa, Integer> {
```

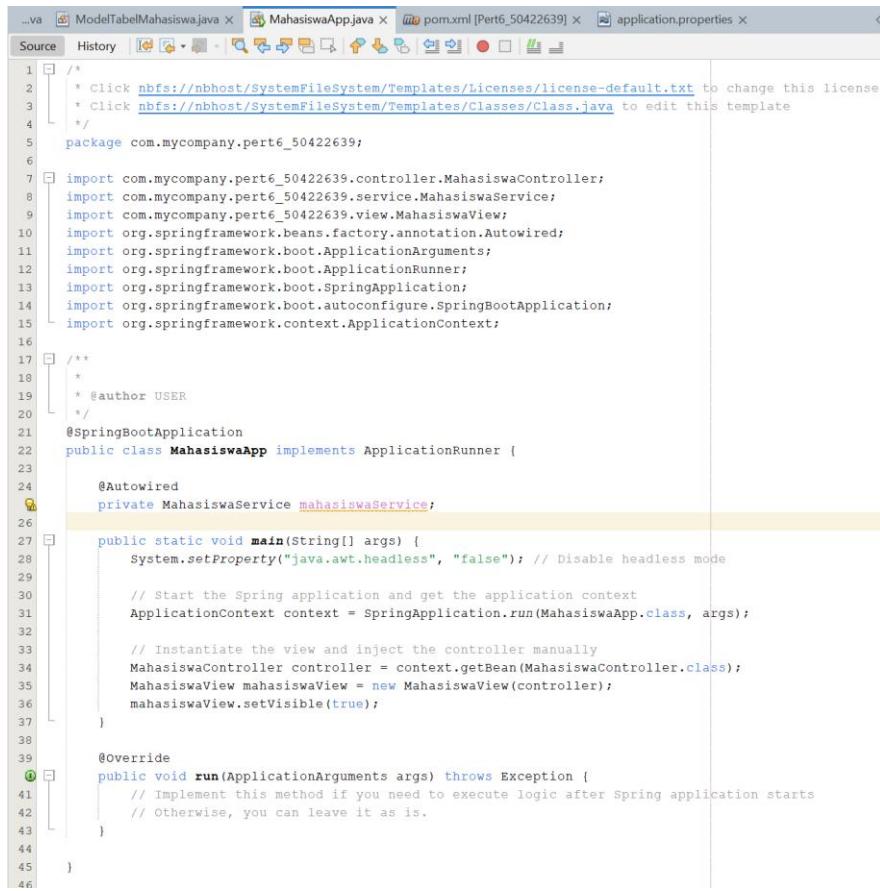
MahasiswaService.java

MahasiswaController.java



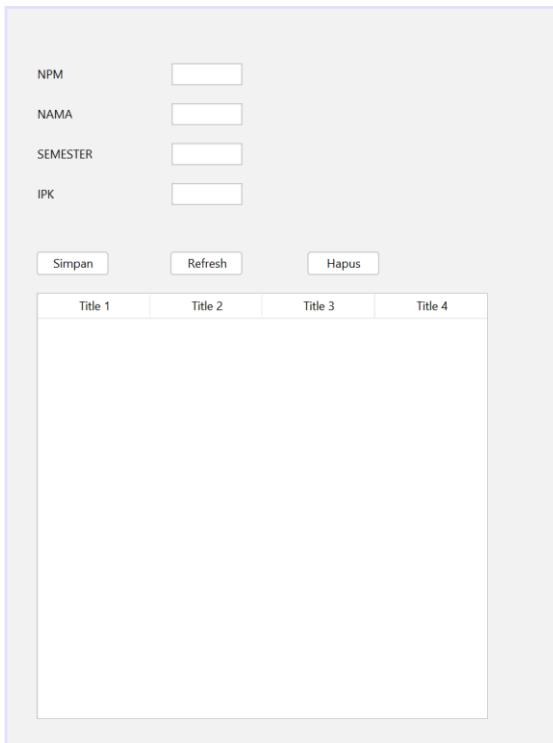
```
1  /*
2  * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
3  * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this template
4  */
5  package com.mycompany.pert6_50422639.controller;
6
7  import com.mycompany.pert6_50422639.model.ModelMahasiswa;
8  import com.mycompany.pert6_50422639.service.MahasiswaService;
9  import java.util.List;
10 import org.springframework.beans.factory.annotation.Autowired;
11 import org.springframework.stereotype.Controller;
12 import org.springframework.web.bind.annotation.PathVariable;
13 import org.springframework.web.bind.annotation.RequestBody;
14
15 /**
16  * 
17  * @author USER
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 }
```

MahasiswaApp.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.mycompany.pert6_50422639;
6
7  import com.mycompany.pert6_50422639.controller.MahasiswaController;
8  import com.mycompany.pert6_50422639.service.MahasiswaService;
9  import com.mycompany.pert6_50422639.view.MahasiswaView;
10 import org.springframework.beans.factory.annotation.Autowired;
11 import org.springframework.boot.ApplicationArguments;
12 import org.springframework.boot.ApplicationRunner;
13 import org.springframework.boot.SpringApplication;
14 import org.springframework.boot.autoconfigure.SpringBootApplication;
15 import org.springframework.context.ApplicationContext;
16
17 /**
18  *
19  * @author USER
20  */
21 @SpringBootApplication
22 public class MahasiswaApp implements ApplicationRunner {
23
24     @Autowired
25     private MahasiswaService mahasiswaService;
26
27     public static void main(String[] args) {
28         System.setProperty("java.awt.headless", "false"); // Disable headless mode
29
30         // Start the Spring application and get the application context
31         ApplicationContext context = SpringApplication.run(MahasiswaApp.class, args);
32
33         // Instantiate the view and inject the controller manually
34         MahasiswaController controller = context.getBean(MahasiswaController.class);
35         MahasiswaView mahasiswaView = new MahasiswaView(controller);
36         mahasiswaView.setVisible(true);
37     }
38
39     @Override
40     public void run(ApplicationArguments args) throws Exception {
41         // Implement this method if you need to execute logic after Spring application starts
42         // Otherwise, you can leave it as is.
43     }
44
45 }
46
```

MahasiswaView.java (design)



MahasiswaView.java (source)

```
7 import com.mycompany.pert6_50422639.controller.MahasiswaController;
8 import com.mycompany.pert6_50422639.model.ModelMahasiswa;
9 import com.mycompany.pert6_50422639.model.ModelTabelMahasiswa;
10 import java.util.List;
11
12 /**
13  * 
14  * @author USER
15 */
16 public final class MahasiswaView extends javax.swing.JFrame {
17
18     private MahasiswaController controller;
19
20     public MahasiswaView(MahasiswaController controller) {
21         this.controller = controller;
22         initComponents();
23         loadMahasiswaTable();
24     }
25     private MahasiswaView() {
26         throw new UnsupportedOperationException("Not supported yet."); // Generated from nbfs://nbhost/SystemFileSystem/Templates/Code/GeneratedMethodBody
27     }
28     public void loadMahasiswaTable() {
29         // Ambil data dari controller
30         List<ModelMahasiswa> listMahasiswa = controller.getAllMahasiswa();
31
32         // Buat model tabel kustom dengan data mahasiswa
33         ModelTabelMahasiswa tableModel = new ModelTabelMahasiswa(listMahasiswa);
34
35         // Set model pada JTable
36         dataTable.setModel(tableModel);
37     }
38
39     @SuppressWarnings("unchecked")
40     [Generated code]
41
42     /**
43      * @param args the command line arguments
44      */
45     public static void main(String args[]) {
46         /* Set the Nimbus look and feel */
47         [Look and feel setting code (optional)]
48
49         /* Create and display the form */
50         java.awt.EventQueue.invokeLater(new Runnable() {
51             public void run() {
52                 new MahasiswaView().setVisible(true);
53             }
54         });
55     }
56 }
```

```

188 // Variables declaration - do not modify
189 private javax.swing.JTable dataTable;
190 private javax.swing.JButton hapusButton;
191 private javax.swing.JTextField ipkField;
192 private javax.swing.JLabel jLabel1;
193 private javax.swing.JLabel jLabel2;
194 private javax.swing.JLabel jLabel3;
195 private javax.swing.JLabel jLabel4;
196 private javax.swing.JScrollPane jScrollPane;
197 private javax.swing.JTextField namaField;
198 private javax.swing.JTextField npmField;
199 private javax.swing.JButton refreshButton;
200 private javax.swing.JTextField semesterField;
201 private javax.swing.JButton simpanButton;
202 // End of variables declaration
203 }

```

simpanButton (function)

```

private void simpanButtonActionPerformed(java.awt.event.ActionEvent evt) {
    String npm = npmField.getText();
    String nama = namaField.getText();
    int semester = Integer.parseInt(semesterField.getText());
    float ipk = Float.parseFloat(ipkField.getText());
    ModelMahasiswa mahasiswa = new ModelMahasiswa(0, npm, nama, semester, ipk);
    System.out.println(mahasiswa.getIpk());
    System.out.println(mahasiswa.getNama());
    System.out.println(mahasiswa.getSemester());
    System.out.println(mahasiswa.getNpm());

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

```

refreshButton (function)

```

private void refreshButtonActionPerformed(java.awt.event.ActionEvent evt) {
    loadMahasiswaTable ();
}

```

hapusButton (function)

```

private void hapusButtonActionPerformed(java.awt.event.ActionEvent evt) {
    JTextField idField = new JTextField(5);

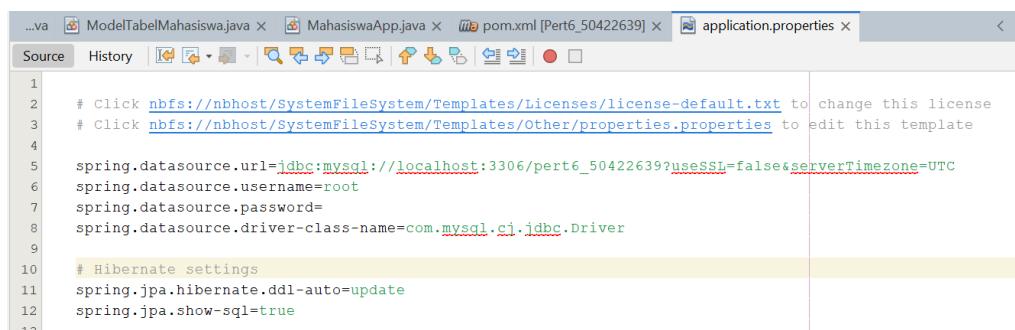
    // Membuat panel untuk menampung JTextField
    JPanel panel = new JPanel();
    panel.add(new JLabel("Masukkan ID yang ingin dihapus:"));
    panel.add(idField);

    // Menampilkan dialog box dengan JTextField, tombol OK, dan Cancel
    int result = JOptionPane.showConfirmDialog(null, panel,
        "Hapus Mahasiswa", JOptionPane.OK_CANCEL_OPTION, JOptionPane.PLAIN_MESSAGE);

    // Jika tombol OK ditekan
    if (result == JOptionPane.OK_OPTION) {
        try {
            // Mengambil input ID dan memanggil metode deleteMhs
            int id = Integer.parseInt(idField.getText());
            controller.deleteMahasiswa(id);
            JOptionPane.showMessageDialog(null, "Data berhasil dihapus.", "Sukses", JOptionPane.INFORMATION_MESSAGE);
        } catch (NumberFormatException e) {
            // Menangani error jika ID yang dimasukkan bukan angka
            JOptionPane.showMessageDialog(null, "ID harus berupa angka.", "Error", JOptionPane.ERROR_MESSAGE);
        }
    }
}

```

application.properties



The screenshot shows a Java IDE interface with multiple tabs at the top: "....java", "ModelTabelMahasiswa.java", "MahasiswaApp.java", "pom.xml [Pert6_50422639]", and "application.properties". The "application.properties" tab is currently active. Below the tabs is a toolbar with various icons. The main area is a code editor showing the following content:

```
1 # Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
2 # Click nbfs://nbhost/SystemFileSystem/Templates/Other/properties.properties to edit this template
3
4 spring.datasource.url=jdbc:mysql://localhost:3306/pert6_50422639?useSSL=false&serverTimezone=UTC
5 spring.datasource.username=root
6 spring.datasource.password=
7 spring.datasource.driver-class-name=com.mysql.cj.jdbc.Driver
8
9
10 # Hibernate settings
11 spring.jpa.hibernate.ddl-auto=update
12 spring.jpa.show-sql=true
```