

ACTIVITY PERTEMUAN 6

NAMA : Hafizhan Noor Amril

NPM : 50422639

KELAS : 4IA16

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

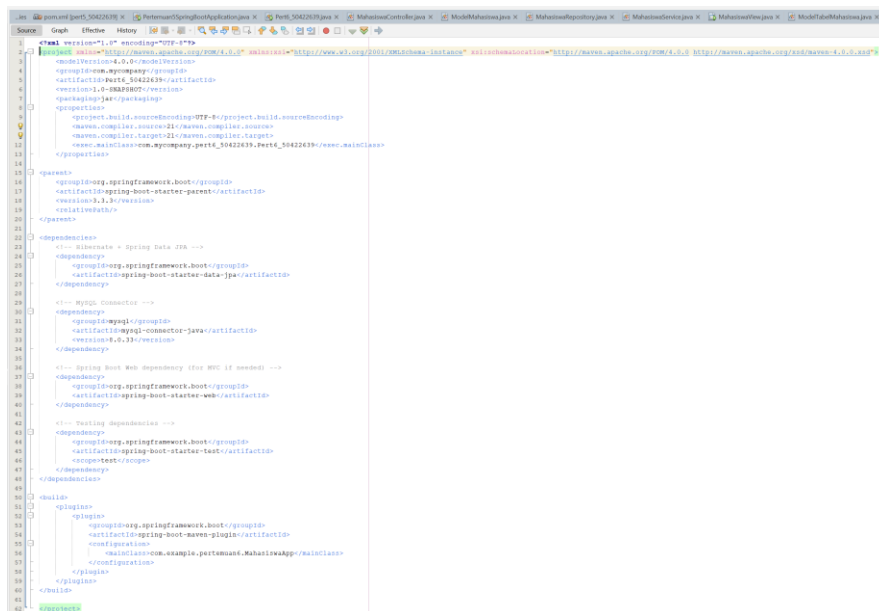
MATA PRAKTIKUM : Rekayasa Perangkat Lunak 2 (RPL 2)

Soal :

1. Screenshoot kode dan output dari program yang telah dibuat

Jawab :

pom.xml



```
1 <?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-instance" xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">
3     <modelVersion>4.0.0</modelVersion>
4     <groupId>com.mycompany</groupId>
5     <artifactId>Perft_50422639</artifactId>
6     <version>1.0-SNAPSHOT</version>
7     <packaging>jar</packaging>
8     <properties>
9         <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
10        <java.compiler>21</java.compiler>
11        <java.compiler.target>21</java.compiler.target>
12        <java.compiler.source>21</java.compiler.source>
13    </properties>
14
15    <parent>
16        <groupId>org.springframework.boot</groupId>
17        <artifactId>spring-boot-starter-parent</artifactId>
18        <version>3.1.3</version>
19        <relativePath></relativePath>
20    </parent>
21
22    <dependencies>
23        <!-- Hiberate - Spring Data JPA -->
24        <dependency>
25            <groupId>org.springframework.boot</groupId>
26            <artifactId>spring-boot-starter-data-jpa</artifactId>
27        </dependency>
28
29        <!-- Myqgl Connector -->
30        <dependency>
31            <groupId>org.springframework</groupId>
32            <artifactId>spring-connector</artifactId>
33            <version>0.8.33</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            <plugin>
57                <groupId>com.example</groupId>
58                <artifactId>Perft_50422639</artifactId>
59            </plugin>
60        </plugins>
61    </build>
62 </project>
```

ModelMahasiswa.java

```
...va application.properties x pom.xml [pert5_50422639] x Pertemuan5SpringBootApplication.java x MahasiswaController.java x ModelMahasiswa.java x MahasiswaRepository.java x M
Source History
1  /*
2   * Click https://www.springframework.io/licenses/license-default.txt to change this license
3   * Click https://www.springframework.io/licenses/license-default.txt to edit this template
4   */
5   package com.mycompany.pert5_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   *
16   * @author USER
17   */
18  @Entity
19  @Table(name = "mahasiswa")
20  public class ModelMahasiswa {
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  }
93
```

ModelTabelMahasiswa.java

```
...va ModelMahasiswa.java x application.properties x pom.xml [pert5_50422639] x Pertemuan5SpringBootApplication.java x MahasiswaController.java x ModelM
Source History
7 import java.util.List;
8 import javax.swing.table.AbstractTableModel;
9
10 /**
11  * @author USER
12  */
13
14 public class ModelTabelMahasiswa extends AbstractTableModel{
15     private List<ModelMahasiswa> mahasiswaList;
16     private final String[] columnNames = {"ID", "NPM", "Nama", "Semester", "IPK"};
17
18     public ModelTabelMahasiswa(List<ModelMahasiswa> mahasiswaList) {
19         this.mahasiswaList = mahasiswaList;
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();
40             case 2:
41                 return mahasiswa.getNama();
42             case 3:
43                 return mahasiswa.getSemester();
44             case 4:
45                 return mahasiswa.getIpk();
46             default:
47                 return null;
48         }
49     }
50
51     @Override
52     public String getColumnName(int column) {
53         return columnNames[column]; // Mengatur nama kolom
54     }
55
56     @Override
57     public boolean isCellEditable(int rowIndex, int columnIndex) {
58         return false; // Semua sel tidak dapat diedit
59     }
60
61     // Method untuk menambahkan atau memodifikasi data, jika dibutuhkan
62     public void setMahasiswaList(List<ModelMahasiswa> mahasiswaList) {
63         this.mahasiswaList = mahasiswaList;
64         fireTableDataChanged(); // Memberitahu JTable bahwa data telah berubah
65     }
66 }
```

MahasiswaRepository.java

```
...va  MahasiswaRepository.java x  MahasiswaService.java x  MahasiswaView.java x  ModelTabelMahasiswa.java x  MahasiswaApp... < > v
Source History
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.repository;
6
7   import com.mycompany.pert6_50422639.model.ModelMahasiswa;
8   import org.springframework.data.jpa.repository.JpaRepository;
9   import org.springframework.stereotype.Repository;
10
11  /**
12   *
13   * @author USER
14   */
15  @Repository
16  public interface MahasiswaRepository extends JpaRepository<ModelMahasiswa, Integer> {
17
18
19  }
20
```

MahasiswaService.java

```
...va  MahasiswaRepository.java x  MahasiswaService.java x  MahasiswaView.java x  ModelTabelMahasiswa.java x  MahasiswaApp... < >
Source History
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.service;
6
7   import com.mycompany.pert6_50422639.model.ModelMahasiswa;
8   import com.mycompany.pert6_50422639.repository.MahasiswaRepository;
9   import java.util.List;
10  import org.springframework.beans.factory.annotation.Autowired;
11  import org.springframework.stereotype.Service;
12
13  /**
14   *
15   * @author USER
16   */
17  @Service
18  public class MahasiswaService {
19
20      @Autowired
21      private MahasiswaRepository repository;
22
23      public void addMhs(ModelMahasiswa mhs) {
24          repository.save(mhs);
25      }
26
27      public ModelMahasiswa getMhs(int id) {
28          return repository.findById(id).orElse(null);
29      }
30
31      public void updateMhs(ModelMahasiswa mhs) {
32          repository.save(mhs);
33      }
34
35      public void deleteMhs(int id) {
36          repository.deleteById(id);
37      }
38
39      public List<ModelMahasiswa> getAllMahasiswa() {
40          return repository.findAll();
41      }
42
43  }
```

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

```
...va ModelTabelMahasiswa.java x MahasiswaApp.java x pom.xml [Pert6_50422639] x application.properties x
Source History
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)

NPM

NAMA

SEMESTER

IPK

Simpan

Refresh

Hapus

| Title 1 | Title 2 | Title 3 | Title 4 |
|---------|---------|---------|---------|
| | | | |

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/Classes/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
44
45
46
47
48
49
50
51
52
53 /**
54  * @param args the command line arguments
55  */
56 public static void main(String args[]) {
57     /* Set the Nimbus look and feel */
58     LookAndFeel lookAndFeel = new NimbusLookAndFeel();
59     lookAndFeel.installLook();
60
61     /* Create and display the form */
62     java.awt.EventQueue.invokeLater(new Runnable() {
63         public void run() {
64             new MahasiswaView().setVisible(true);
65         }
66     });
67 }
```

```

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 jScrollPane1;
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);

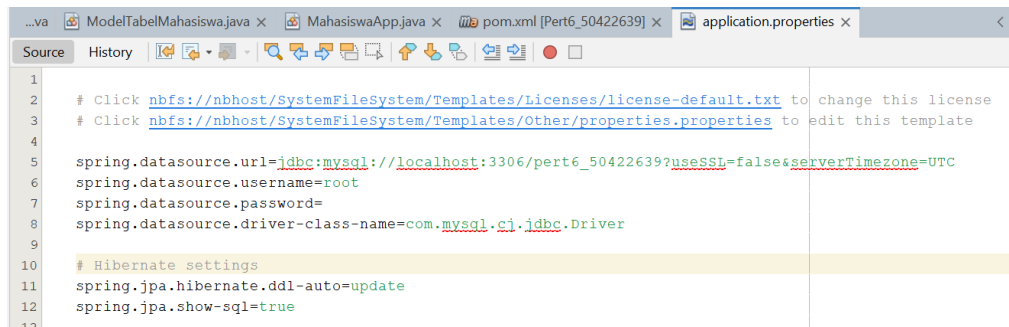
    // Membuat panel untuk menampung JTextField
    JPanel panel = new JPanel();
    panel.add(new JLabel("Masukkan ID yang ingin dihapus:"));
    panel.add(idField);

    // 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



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