

ACTIVITY PERTEMUAN 6

NAMA : Yogi Setiawan

NPM : 5142648

KELAS : 4IA17

MATERI : IMPLEMENTASI AOP DAN DEPENDENCY INJECTION PADA PROJECT SPRING DAN HIBERNATE

MATA PRAKTIKUM : RPL 2

The screenshot shows the NetBeans IDE interface. At the top, the project name 'pert6_yogi [0000000...]' is visible, along with the 'Source Packages' node which contains several Java packages: com.mycompany.pert6_yogi, com.mycompany.pert6_yogi.Controller, com.mycompany.pert6_yogi.Model, com.mycompany.pert6_yogi.Repository, com.mycompany.pert6_yogi.Service, and com.mycompany.pert6_yogi.View.

The main window displays the code editor for 'MahasiswaApp.java'. The code is as follows:

```
/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
 */
package com.mycompany.pert6_yogi;

import com.mycompany.pert6_yogi.Controller.MahasiswaController;
import com.mycompany.pert6_yogi.Service.MahasiswaService;
import com.mycompany.pert6_yogi.View.MahasiswaView;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.boot.ApplicationArguments;
import org.springframework.boot.ApplicationRunner;
import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;
import org.springframework.context.ApplicationContext;

/**
 *
 * @Author ASUS
 */
@SpringBootApplication
public class MahasiswaApp implements ApplicationRunner {
    @Autowired
    private MahasiswaService mahasiswaService;

    public static void main(String[] args) {
        System.setProperty("java.awt.headless", "false"); // Disable headless mode

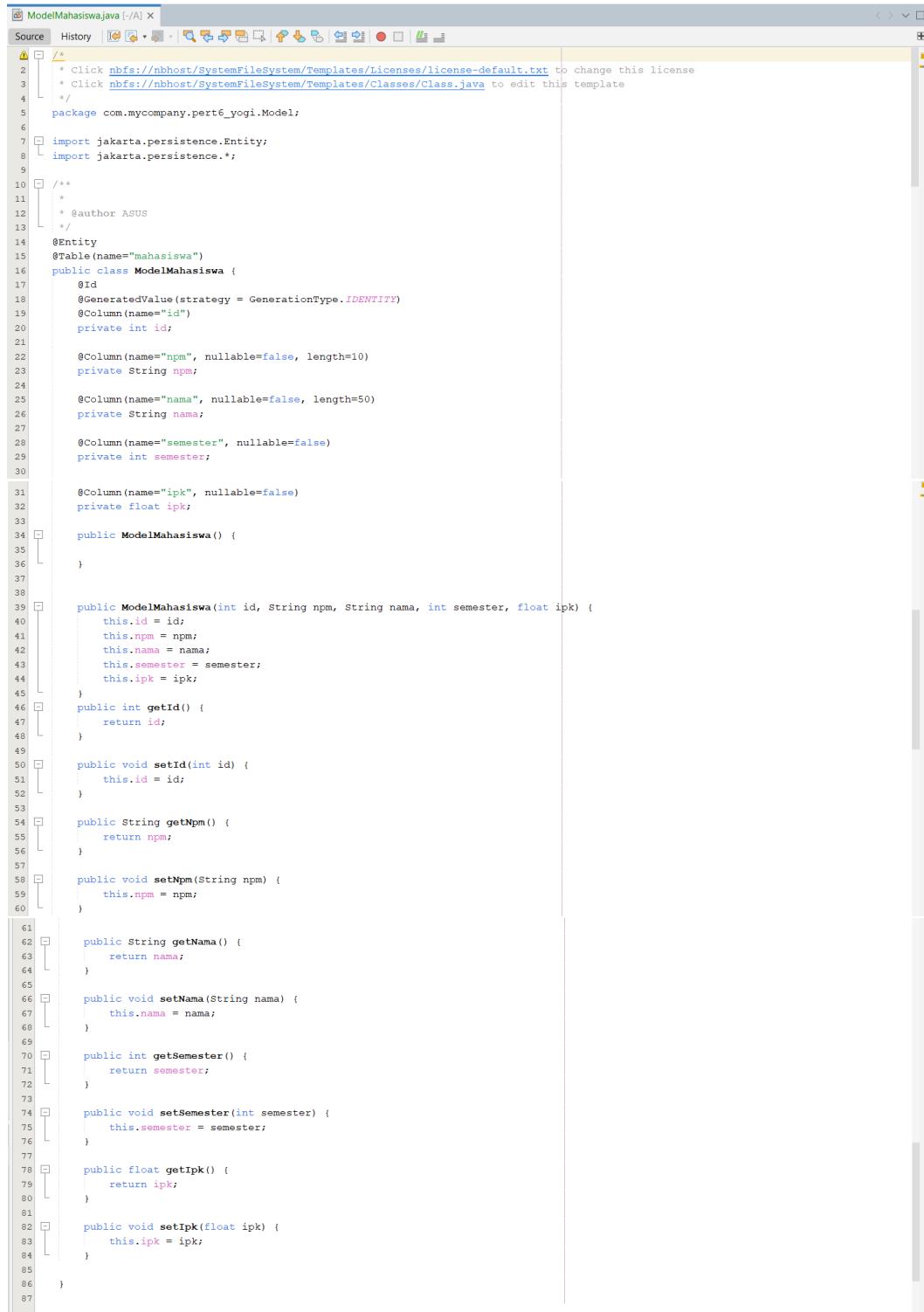
        // Start the Spring application and get the application context
        ApplicationContext context = SpringApplication.run(MahasiswaApp.class, args);

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

The screenshot shows a Java code editor window with the title "MahasiswaController.java [-/A]". The code is a Spring REST controller for managing student data. It includes imports for Model, Service, and various Spring annotations like @Controller, @Autowired, and @RequestBody. The controller has methods for adding, getting, updating, and deleting students, as well as a method to get all students.

```
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_yogi.Controller;
6
7  import com.mycompany.pert6_yogi.Model.ModelMahasiswa;
8  import com.mycompany.pert6_yogi.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  * @author ASUS
17  */
18 @Controller
19 public class MahasiswaController {
20     @Autowired
21     private MahasiswaService mahasiswaService;
22
23     public String addMahasiswa(@RequestBody ModelMahasiswa mhs) {
24         mahasiswaService.addMhs(mhs);
25         return "Mahasiswa added successfully";
26     }
27
28     public ModelMahasiswa getMahasiswa(@PathVariable int id) {
29         return mahasiswaService.getMhs(id);
30     }
31
32     public String updateMahasiswa(@RequestBody ModelMahasiswa mhs){
33         mahasiswaService.updateMhs(mhs);
34         return "Mahasiswa updated successfully";
35     }
36
37     public String deleteMahasiswa(@PathVariable int id) {
38         mahasiswaService.deleteMhs(id);
39         return "Mahasiswa deleted successfully";
40     }
41
42     public List<ModelMahasiswa> getAllMahasiswa() {
43         return mahasiswaService.getAllMahasiswa();
44     }
45
46 }
47
48 }
```



The screenshot shows a Java code editor window with the file `ModelMahasiswa.java` open. The code defines a class `ModelMahasiswa` with annotations for persistence and validation.

```
/*
 * 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_yogi.Model;

import jakarta.persistence.Entity;
import jakarta.persistence.*;

/**
 * @author ASUS
 */
@Entity
@Table(name="mahasiswa")
public class ModelMahasiswa {
    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    @Column(name="id")
    private int id;

    @Column(name="npm", nullable=false, length=10)
    private String npm;

    @Column(name="nama", nullable=false, length=50)
    private String nama;

    @Column(name="semester", nullable=false)
    private int semester;

    @Column(name="ipk", nullable=false)
    private float ipk;

    public ModelMahasiswa() {
    }

    public ModelMahasiswa(int id, String npm, String nama, int semester, float ipk) {
        this.id = id;
        this.npm = npm;
        this.nama = nama;
        this.semester = semester;
        this.ipk = ipk;
    }

    public int getId() {
        return id;
    }

    public void setId(int id) {
        this.id = id;
    }

    public String getNpm() {
        return npm;
    }

    public void setNpm(String npm) {
        this.npm = npm;
    }

    public String getNama() {
        return nama;
    }

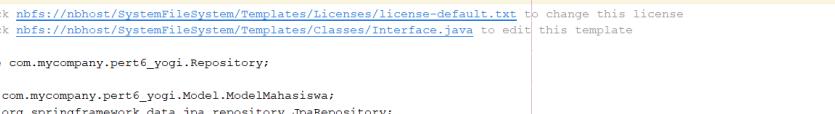
    public void setNama(String nama) {
        this.nama = nama;
    }

    public int getSemester() {
        return semester;
    }

    public void setSemester(int semester) {
        this.semester = semester;
    }

    public float getIpk() {
        return ipk;
    }

    public void setIpk(float ipk) {
        this.ipk = ipk;
    }
}
```

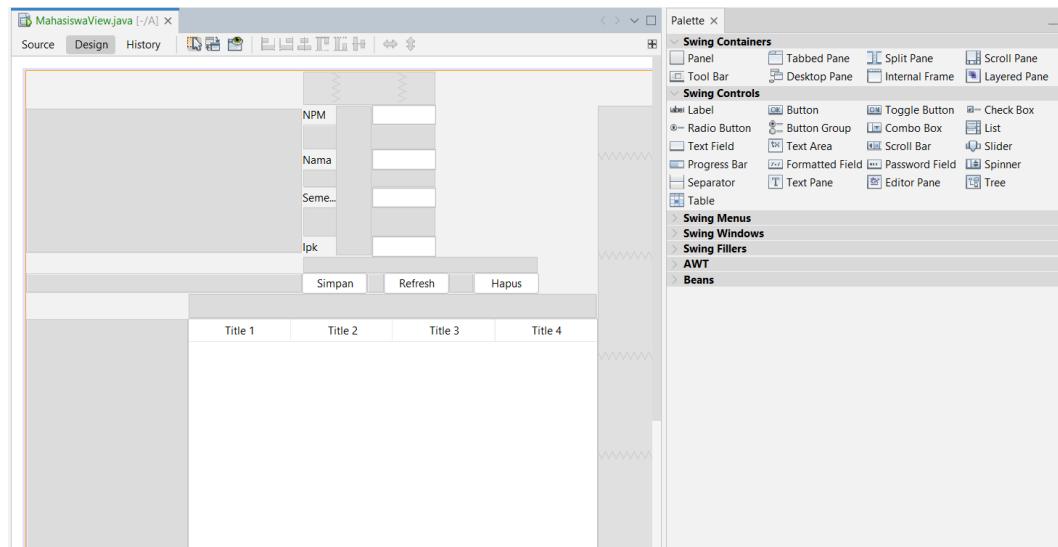


The screenshot shows the NetBeans IDE interface with the title bar "MahasiswaRepository.java [-/A] X". The left sidebar shows the project structure with a single node "MahasiswaRepository.java". The main editor area displays the Java code for the repository interface:

```
1 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
2 * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Interface.java to edit this template
3 */
4
5 package com.mycompany.pert6_yogi.Repository;
6
7 import com.mycompany.pert6_yogi.Model.ModelMahasiswa;
8 import org.springframework.data.jpa.repository.JpaRepository;
9 import org.springframework.stereotype.Repository;
10
11 /**
12  * @author ASUS
13  */
14
15
16 @Repository
17 public interface MahasiswaRepository extends JpaRepository <ModelMahasiswa, Integer> {
18
19 }
```

The screenshot shows a Java code editor window titled "MahasiswaService.java". The code implements a service interface for managing student data. It includes imports for ModelMahasiswa, MahasiswaRepository, and various Spring annotations. The class contains methods for adding, getting, updating, and deleting students, as well as a method to get all students.

```
1  * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
2  * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this template
3
4  package com.mycompany.pert6_yogi.Service;
5
6
7  import com.mycompany.pert6_yogi.Model.ModelMahasiswa;
8  import com.mycompany.pert6_yogi.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 ASUS
16  */
17 @Service
18 public class MahasiswaService {
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         return repository.findById(id).orElse(null);
28     }
29
30     public void updateMhs(ModelMahasiswa mhs) {
31         repository.save(mhs);
32     }
33
34     public void deleteMhs(int id) {
35         repository.deleteById(id);
36     }
37
38     public List<ModelMahasiswa> getAllMahasiswa() {
39         return repository.findAll();
40     }
41
42 }
43
```



```
/* Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
 * Click nbfs://nbhost/SystemFileSystem/Templates/GUIForms/JFrame.java to edit this template
 */
package com.mycompany.pert6_yogi.View;

import com.mycompany.pert6_yogi.Controller.MahasiswaController;
import com.mycompany.pert6_yogi.Model.ModelMahasiswa;
import com.mycompany.pert6_yogi.Model.ModelTabelMahasiswa;
import java.util.List;

/**
 *
 * @author ASUS
 */
public class MahasiswaView extends javax.swing.JFrame {
    private MahasiswaController controller;

    /**
     * Creates new form MahasiswaView
     */
    public MahasiswaView(MahasiswaController controller) {
        this.controller = controller;
        initComponents();
        loadMahasiswaTable();
    }

    private MahasiswaView() {
        throw new UnsupportedOperationException("Not Supported yet.");
    }

    private MahasiswaView() {
        throw new UnsupportedOperationException("Not Supported yet.");
    }

    public void loadMahasiswaTable() {
        List <ModelMahasiswa> listMahasiswa = controller.getAllMahasiswa();

        ModelTabelMahasiswa tabelModel = new ModelTabelMahasiswa(listMahasiswa);

        dataTable.setModel(tabelModel);
    }

    /**
     * This method is called from within the constructor to initialize the form.
     * WARNING: Do NOT modify this code. The content of this method is always
     * regenerated by the Form Editor.
     */
    @SuppressWarnings("unchecked")
    // Generated Code
    private void btnRefreshActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
    }

    private void btnSimpanActionPerformed(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.getNpm());
        System.out.println(mahasiswa.getSemester());

        controller.addMahasiswa(mahasiswa);
        loadMahasiswaTable();
        // TODO add your handling code here:
    }

    private void btnHapusActionPerformed(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
    }
}
```

```

211     if (result == JOptionPane.OK_OPTION) {
212         try {
213             // Mengambil input ID dan memanggil metode deleteMhs
214             int id = Integer.parseInt(idField.getText());
215             controller.deleteMahasiswa(id);
216             JOptionPane.showMessageDialog(null, "Data berhasil dihapus.", "Sukses", JOptionPane.INFORMATION_MESSAGE);
217         } catch (NumberFormatException e) {
218             // Menangani error jika ID yang dimasukkan bukan angka
219             JOptionPane.showMessageDialog(null, "ID harus berupa angka.", "Error", JOptionPane.ERROR_MESSAGE);
220         }
221     }
222     // TODO add your handling code here:
223 }
224
225 /**
226 * @param args the command line arguments
227 */
228 public static void main(String args[]) {
229     /* Set the Nimbus look and feel */
230     /* Look and feel setting code (optional) */
231
232     /* Create and display the form */
233     java.awt.EventQueue.invokeLater(new Runnable() {
234         public void run() {
235             new MahasiswaView().setVisible(true);
236         }
237     });
238 }
239
240 // Variables declaration - do not modify
241 private javax.swing.JButton btnHapus;
242 private javax.swing.JButton btnRefresh;
243 private javax.swing.JButton btnSimpan;
244 private javax.swing.JTable dataTable;
245 private javax.swing.JTextField ipkField;
246 private javax.swing.JLabel jLabel1;
247 private javax.swing.JLabel jLabel2;
248 private javax.swing.JLabel jLabel3;
249 private javax.swing.JLabel jLabel4;
250 private javax.swing.JScrollPane jScrollPane;
251 private javax.swing.JTextField namaField;
252 private javax.swing.JTextField npmField;
253 private javax.swing.JTextField semesterField;
254 // End of variables declaration
255 }

```

localhost/phpmyadmin/index.php?route=db_per16&table=mahasiswa&pos=0

phpMyAdmin

Recent Favorites

Server: 127.0.0.1 | Databases: db_per16 | Table: mahasiswa

Browse Structure SQL Search Insert Export Import Privileges Operations Tracking Triggers

Showing rows 0 - 0 (1 total, Query took 0.0003 seconds.)

SELECT * FROM `mahasiswa`

Profiling | Edit inline | Edit | Explain SQL | Create PHP code | Refresh

Show all Number of rows: 25 Filter rows: Search this table

Extra options

	id	ipk	name	npm	semester
<input type="checkbox"/>	1	4	yogi	51422648	7

Check all With selected: Edit Copy Delete Export

Show all Number of rows: 25 Filter rows: Search this table

Query results operations

Print Copy to clipboard Export Display chart Create view

Bookmark this SQL query

Label: Let every user access this bookmark

Bookmark this SQL query

NPM

Nama

Sem...

Ipk

ID	NPM	Nama	Semester	IPK
4	51422648	YOGI	7	4.0
5	51422648	YOGI	7	4.0

A screenshot of a Windows application window titled "Mahasiswa". The window contains four text input fields and three buttons:

NPM	51422648
Nama	YOGI
Sem...	7
Ipk	4

Buttons: Simpan, Refresh, Hapus.

Below the form is a table:

ID	NPM	Nama	Semester	IPK
4	51422648	YOGI	7	4.0
5	51422648	YOGI	7	4.0

A modal dialog box titled "Hapus Mahasiswa" is displayed, asking for confirmation:

Masukkan ID yang ingin dihapus:

Buttons: OK, Cancel.

— □ ×

NPM	<input type="text" value="51422648"/>
Nama	<input type="text" value="YOGI"/>
Sem...	<input type="text" value="7"/>
Ipk	<input type="text" value="4"/>

Simpan **Refresh** **Hapus**

ID	NPM	Nama	Semester	IPK
4	51422648	YOGI	7	4.0