

ACTIVITY PERTEMUAN 5

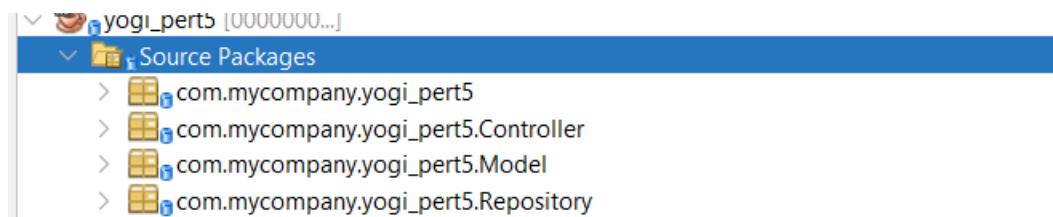
NAMA : Yogi Setiawan

NPM : 5142648

KELAS : 4IA17

MATERI : Konsep Framework Spring, Pembuatan Project Spring dan Hibernate

MATA PRAKTIKUM : RPL 2



The image shows two Java code editors side-by-side in a development environment.

Yogi_pert5.java

```
/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
 */
package com.mycompany.yogi_pert5;

import com.mycompany.yogi_pert5.Controller.MahasiswaController;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.boot.CommandLineRunner;
import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;

/**
 *
 * @author ASUS
 */
@SpringBootApplication
public class Yogi_pert5 implements CommandLineRunner {

    @Autowired
    private MahasiswaController mhsController;
    public static void main(String[] args) {
        SpringApplication.run(Yogi_pert5.class, args);
    }

    @Override
    public void run(String... args) throws Exception {
        mhsController.tampilkanMenu();
    }
}
```

MahasiswaController.java

```
/*
 * 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.yogi_pert5.Controller;

import com.mycompany.yogi_pert5.Model.MahasiswaModel;
import com.mycompany.yogi_pert5.Repository.MahasiswaRepository;
import java.util.List;
import java.util.Scanner;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Controller;

/**
 *
 * @author ASUS
 */
@Controller
public class MahasiswaController {
    @Autowired
    private MahasiswaRepository mahasiswaRepository;

    public void tampilkanMenu() {
        Scanner scanner = new Scanner(System.in);
        int opsi;

        do {
            System.out.println("\nMenu:");
            System.out.println("1. Tampilkan semua mahasiswa");
            System.out.println("2. Tambah mahasiswa baru");
            System.out.print("Pilih menu: ");
            opsi = scanner.nextInt();
        } while (opsi < 1 || opsi > 2);
    }
}
```

```
31     System.out.println("3. Cek koneksi database");
32     System.out.println("4. Keluar");
33     System.out.print("Pilih opsi: ");
34     opsi = scanner.nextInt();
35     scanner.nextLine(); // menangkap newline
36
37     switch (opsi) {
38         case 1:
39             tampilkanSemuaMahasiswa();
40             break;
41         case 2:
42             tambahMahasiswa(scanner);
43             break;
44         case 3:
45             cekKoneksi();
46             break;
47         case 4:
48             System.out.println("Keluar dari program.");
49             break;
50         default:
51             System.out.println("Opsi tidak valid, coba lagi.");
52     }
53
54 } while (opsi != 4);
55
56
57 private void tampilkanSemuaMahasiswa() {
58     List<MahasiswaModel> mahasiswaList = mahasiswaRepository.findAll();
59     if (mahasiswaList.isEmpty()) {
60         System.out.println("Tidak ada data mahasiswa.");
61     } else {
62         mahasiswaList.forEach(mahasiswa -> System.out.println(mahasiswa));
63     }
64 }
65
66 private void tambahMahasiswa(Scanner scanner) {
67     System.out.print("Masukkan NFM : ");
68     String npm = scanner.nextLine();
69     System.out.print("Masukkan Nama : ");
70     String nama = scanner.nextLine();
71     System.out.print("Masukkan Semester : ");
72     int semester = scanner.nextInt();
73     System.out.print("Masukkan IPK : ");
74     float ipk = scanner.nextFloat();
75
76     MahasiswaModel mahasiswa = new MahasiswaModel(0, npm, nama, semester, ipk);
77     mahasiswaRepository.save(mahasiswa);
78     System.out.println("Mahasiswa berhasil ditambahkan.");
79 }
80
81 private void cekKoneksi() {
82     try {
83         mahasiswaRepository.findAll();
84         System.out.println("Koneksi ke database berhasil.");
85     } catch (Exception e) {
86         System.out.println("Gagal terhubung ke database.");
87     }
88 }
89
90 }
91
```

```
/*
 * 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.yogi_pert5.Model;

import jakarta.persistence.*;

@Entity
@Table(name="mahasiswa")
public class MahasiswaModel {
    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    @Column(name = "id")
    private int id;

    @Column(name = "npm", nullable = false, length = 8)
    private String npm;

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

    @Column(name = "semester")
    private int semester;

    @Column(name = "ipk")
    private float ipk;

    // constructor kosong untuk ORM
    public MahasiswaModel() {
    }

    //constructor ada isinya untuk dipanggil di aplikasi
    public MahasiswaModel(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) {
    }
}
```

```
61     this.npm = npm;
62 }
63
64     public String getNama() {
65         return nama;
66     }
67
68     public void setNama(String nama) {
69         this.nama = nama;
70     }
71
72     public int getSemester() {
73         return semester;
74     }
75
76     public void setSemester(int semester) {
77         this.semester = semester;
78     }
79
80     public float getIpk() {
81         return ipk;
82     }
83
84     public void setIpk(float ipk) {
85         this.ipk = ipk;
86     }
87
88     @Override
89     public String toString() {
90         return "Mahasiswa" +
91                 "... ..." +
92                         "id=" + id +
93                         ", nama='" + npm + '\'' +
94                         ", nama='" + nama + '\'' +
95                         ", nama='" + semester + '\'' +
96                         ", jurusan='" + ipk + '\'' +
97                         '}';
98
99     }
100 }
```

Project Files Services

JavaLibrary1 [00000000]

persit [00000000]

gpt_gmt [00000000]

gpt_gmt18 [00000000]

armiper1 [00000000]

yogipert5 [00000000]

gptc

main

java

resources

application.properties [-A]

test

target

pom.xml [-A]

application.properties [-A]

MahasiswaRepository.java - Navigator

Members

MahasiswaRepository :none :IpaRepository<MahasiswaModel, Long>

```

1 # Click nbfs://nbshost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
2 # Click nbfs://nbshost/SystemFileSystem/Templates/Other/properties.properties to edit this template
3
4
5
6 # Konfigurasi MySQL Database
7 spring.datasource.url=jdbc:mysql://localhost:3306/yogi_db?useSSL=false&serverTimezone=UTC
8 spring.datasource.username=root
9 spring.datasource.password=
10 spring.datasource.driver-class-name=com.mysql.cj.jdbc.Driver
11
12 # Hibernate settings
13 spring.jpa.hibernate.ddl-auto=update
14 spring.jpa.show-sql=true
15
16

```

Output X

ASUS - C:\Users\ASUS X Run (Yogi_pert5) X

```

2025-11-15T15:12:30.240+07:00 INFO 6272 --- [           main] j.LocalContainerEntityManagerFactoryBean : Initialized JPA EntityManagerFactory for persistence unit 'Default Unit'
2025-11-15T15:12:30.652+07:00 WARD 6272 --- [           main] JpaBaseConfiguration$JpaWebConfiguration : spring.jpa.open-in-view is enabled by default. Add 'spring.jpa.open-in-view' to the configuration to disable.
2025-11-15T15:12:31.268+07:00 INFO 6272 --- [           main] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat started on port 8080 (http://localhost:8080)
2025-11-15T15:12:31.286+07:00 INFO 6272 --- [           main] com.mycompany.yogi_pert5.Yogi_pert5 : Started Yogi_pert5 in 6.01 seconds (JVM running for 6.01)

```

Menu:

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

Pilih opsi: 2

Masukkan NPM : 51422648

Masukkan Nama : Yogi

Masukkan Semester : 7

Masukkan IPK : 4

Hibernate: insert into mahasiswa (ipk,nama,npn,semester) values (?, ?, ?, ?)

Mahasiswa berhasil ditambahkan.

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.

Menu:

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

Pilih opsi:

localhost/phpmyadmin/index.php?route=/sql&db=yogi_db&table=mahasiswa&pos=0

phpMyAdmin 4.8.0.1

Server: 127.0.0.1 > Database: yogi_db > 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	nama	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