

ACTIVITY PERTEMUAN 5

NAMA : Muhammad Rifqi Dwi Putra
NPM : 51422145
KELAS : 4IA08
MATERI : Konsep framework spring, pembuatan project spring dan hibernate
MATA PRAKTIKUM : Rekayasa Perangkat Lunak 2

(Screenshot langkah-langkah sesuai video pembelajaran dan jelaskan dengan ringkas)

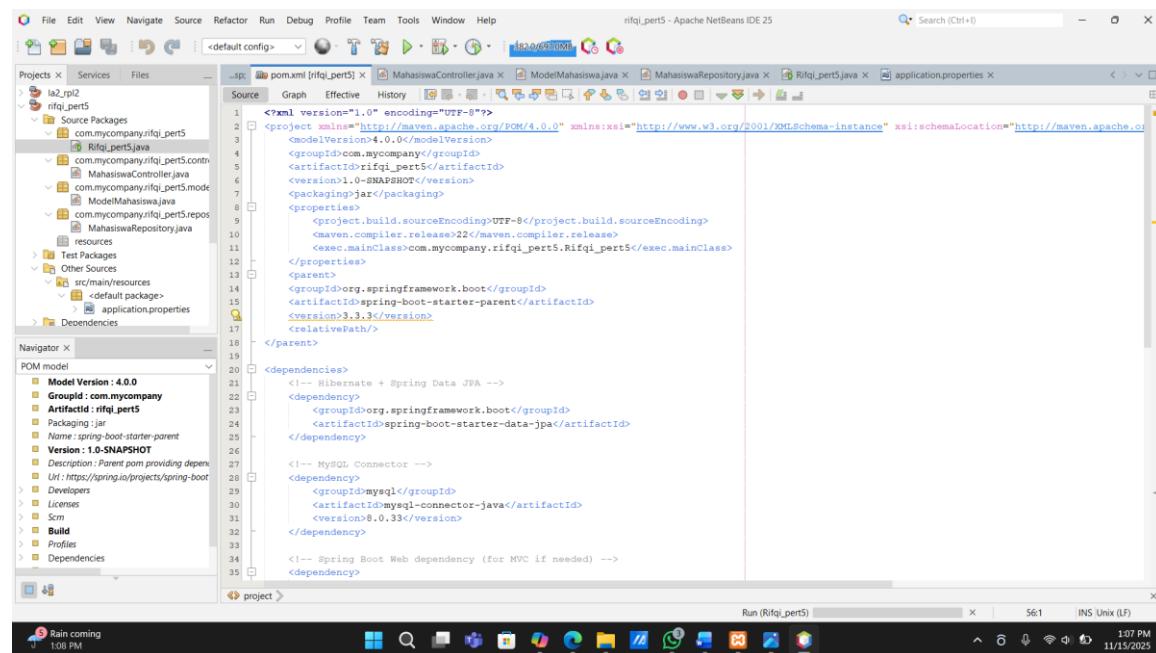
1. Dependency Injection (DI) adalah konsep penting dalam Spring Framework. Jelaskan apa itu Dependency Injection dan mengapa DI sangat penting dalam pengembangan aplikasi berbasis Spring Boot.

Dependency Injection (DI) adalah teknik dalam Spring Framework yang memungkinkan objek-objek yang dibutuhkan oleh suatu class disediakan oleh container, bukan dibuat langsung oleh class tersebut. Dengan DI, komponen dalam aplikasi menjadi lebih modular dan mudah diuji karena dependensi dapat disuntikkan secara otomatis oleh Spring Boot. Hal ini mempercepat pengembangan, mempermudah pemeliharaan, dan mendukung prinsip pemisahan tanggung jawab antar komponen.

2. Screenshot Kode dan Output Program!

Kode :

Pom.xml



```
<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://www.w3.org/2001/XMLSchema-instance">
  <modelVersion>4.0.0</modelVersion>
  <groupId>com.mycompany</groupId>
  <artifactId>rifqi_pert5</artifactId>
  <version>1.0-SNAPSHOT</version>
  <packaging>jar</packaging>
  <properties>
    <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
    <maven.compiler.release>22</maven.compiler.release>
    <exec.mainClass>com.mycompany.rifqi_perts.Rifqi_pert5</exec.mainClass>
  </properties>
  <parent>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-parent</artifactId>
    <version>3.3.3</version>
    <relativePath/>
  </parent>
  <dependencies>
    <!-- Hibernate + Spring Data JPA -->
    <dependency>
      <groupId>org.springframework.boot</groupId>
      <artifactId>spring-boot-starter-data-jpa</artifactId>
    </dependency>
    <!-- MySQL Connector -->
    <dependency>
      <groupId>mysql</groupId>
      <artifactId>mysql-connector-java</artifactId>
      <version>8.0.33</version>
    </dependency>
    <!-- Spring Boot Web dependency (for MVC if needed) -->
    <dependency>
```

Apache NetBeans IDE 25

File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help

Search (Ctrl+I)

Projects Services Files <default config>

Source Graph Effective History

File pom.xml [rifqi_pert5] MahasiswaController.java ModelMahasiswa.java MahasiswaRepository.java Rifqi_pert5.java application.properties

Source Packages com.mycompany.rifqi_pert5 com.mycompany.rifqi_pert5 MahasiswaController.java com.mycompany.rifqi_pert5.mode ModelMahasiswa.java com.mycompany.rifqi_pert5.repo MahasiswaRepository.java resources Test Packages Other Sources src/main/resources default package application.properties

Dependencies

POM model

Model Version : 4.0.0 GroupId : com.mycompany ArtifactId : rifqi_pert5 Packaging : jar Name: spring-boot-starter-parent Version : 1.0-SNAPSHOT Description: Parent pom providing dependencies. Uri: https://spring.io/projects/spring-boot Developers Licenses SCM Build Profiles Dependencies

Rain coming 108 PM

Run (Rifqi_pert5) 561 INS Unix (LF) 1:07 PM 11/15/2025

```
<!-- Spring Boot Web dependency (for MVC if needed) -->
<dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-web</artifactId>
</dependency>

<!-- Testing dependencies -->
<dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-test</artifactId>
    <scope>test</scope>
</dependency>
</dependencies>

<build>
    <plugins>
        <plugin>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-maven-plugin</artifactId>
        </plugin>
    </plugins>
</build>
</project>
```

MahasiswaController.java

Apache NetBeans IDE 25

File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help

Search (Ctrl+I)

Projects Services Files <default config>

Source Graph Effective History

File pom.xml [rifqi_pert5] MahasiswaController.java ModelMahasiswa.java MahasiswaRepository.java Rifqi_pert5.java application.properties

Source Packages com.mycompany.rifqi_pert5 com.mycompany.rifqi_pert5 MahasiswaController.java com.mycompany.rifqi_pert5.mode ModelMahasiswa.java com.mycompany.rifqi_pert5.repo MahasiswaRepository.java resources Test Packages Other Sources src/main/resources default package application.properties

Dependencies

MahasiswaController.java - Navigator

Members

- MahasiswaController
- cekKoneksi()
- tambahMahasiswa(Scanner scanner)
- tampilkanMenu()
- tampilkanSemuaMahasiswa()
- mahasiswaRepository: MahasiswaRepo

History

Click nbfs://nhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license.

Click nbfs://nhost/SystemFileSystem/Templates/Classes/Class.java to edit this template.

```
/*
 * Click nbfs://nhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license.
 * Click nbfs://nhost/SystemFileSystem/Templates/Classes/Class.java to edit this template.
 */
package com.mycompany.rifqi_pert5.controller;

import com.mycompany.rifqi_pert5.model.ModelMahasiswa;
import com.mycompany.rifqi_pert5.repository.MahasiswaRepository;
import java.util.List;
import java.util.Scanner;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Controller;

/**
 *
 * @author mrifq
 */
@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.println("3. Cek koneksi database");
            System.out.println("4. Keluar");
            System.out.print("Pilih opsi: ");
            opsi = scanner.nextInt();
            scanner.nextLine(); // menangkap newline
        } while (opsi != 4);
    }
}
```

Rain mostly cloudy 90:1 INS Windows (CRLF) 1:07 PM 11/15/2025

```
switch (opsi) {
    case 1:
        tampilanSemuaMahasiswa();
        break;
    case 2:
        tambahMahasiswa(scanner);
        break;
    case 3:
        cekKoneksi();
        break;
    case 4:
        System.out.println("Keluar dari program.");
        break;
    default:
        System.out.println("Opsi tidak valid, coba lagi.");
}
}

} while (opsi != 4);

private void tampilanSemuaMahasiswa() {
    List<ModelMahasiswa> mahasiswaList = mahasiswaRepository.findAll();
    if (mahasiswaList.isEmpty()) {
        System.out.println("Tidak ada data mahasiswa.");
    } else {
        mahasiswaList.forEach(mahasiswa -> System.out.println(mahasiswa));
    }
}

private void tambahMahasiswa(Scanner scanner) {
    System.out.print("Masukkan NPM : ");
    String npm = scanner.nextLine();
    System.out.print("Masukkan Nama : ");
    String name = scanner.nextLine();
    System.out.print("Masukkan Semester : ");
    int semester = scanner.nextInt();
    System.out.print("Masukkan IPK : ");
    float ipk = scanner.nextFloat();

    ModelMahasiswa mahasiswa = new ModelMahasiswa(0, npm, name, semester, ipk);
    mahasiswaRepository.save(mahasiswa);
    System.out.println("Mahasiswa berhasil ditambahkan.");
}

private void cekKoneksi() {
    try {
        mahasiswaRepository.findAll();
        System.out.println("Koneksi ke database berhasil.");
    } catch (Exception e) {
        System.out.println("Gagal terhubung ke database.");
    }
}
```

```
private void tambahMahasiswa(Scanner scanner) {
    System.out.print("Masukkan NPM : ");
    String npm = scanner.nextLine();
    System.out.print("Masukkan Nama : ");
    String name = scanner.nextLine();
    System.out.print("Masukkan Semester : ");
    int semester = scanner.nextInt();
    System.out.print("Masukkan IPK : ");
    float ipk = scanner.nextFloat();

    ModelMahasiswa mahasiswa = new ModelMahasiswa(0, npm, name, semester, ipk);
    mahasiswaRepository.save(mahasiswa);
    System.out.println("Mahasiswa berhasil ditambahkan.");
}

private void cekKoneksi() {
    try {
        mahasiswaRepository.findAll();
        System.out.println("Koneksi ke database berhasil.");
    } catch (Exception e) {
        System.out.println("Gagal terhubung ke database.");
    }
}

private void tampilanSemuaMahasiswa() {
    List<ModelMahasiswa> mahasiswaList = mahasiswaRepository.findAll();
    if (mahasiswaList.isEmpty()) {
        System.out.println("Tidak ada data mahasiswa.");
    } else {
        mahasiswaList.forEach(mahasiswa -> System.out.println(mahasiswa));
    }
}
```

ModelMahasiswa.java

The screenshot shows the Apache NetBeans IDE interface. The title bar reads "rifqi_pert5 - Apache NetBeans IDE 25". The menu bar includes File, Edit, View, Navigate, Source, Refactor, Run, Debug, Profile, Team, Tools, Window, Help. The toolbar has icons for file operations like Open, Save, Find, and Build. The left sidebar shows the project structure under "Projects X Services Files" for a project named "rifqi_pert5". It contains packages like "com.mycompany.rifqi_pert5" and "com.mycompany.rifqi_pert5.controllers", and files like "ModelMahasiswa.java", "MahasiswaController.java", "MahasiswaRepository.java", and "application.properties". Below the project tree is a "Dependencies" section. The main editor area displays the Java code for "ModelMahasiswa.java". The code defines a class "ModelMahasiswa" with fields for id, npm, nama, semester, and ipk, along with their corresponding getters and setters. The code uses annotations from the jakarta.persistence package. The status bar at the bottom shows "Run (Rifqi_pert5)" and "INS Windows (CR LF)".

```
1  /*
2  * Click nbfs://nbshost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
3  * Click nbfs://nbshost/SystemFileSystem/Templates/Classes/Class.java to edit this template
4  */
5  package com.mycompany.rifqi_pert5.model;
6
7  import jakarta.persistence.*;
8
9  /**
10  *
11  * @author mrifqi
12  */
13
14  @Entity
15  @Table(name="mahasiswa")
16  public class ModelMahasiswa {
17
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 name;
28
29      @Column(name = "semester")
30      private int semester;
31
32      @Column(name = "ipk")
33      private float ipk;
34
35      // constructor kosong untuk ORM
36      public ModelMahasiswa() {
37
38  }
```

The screenshot shows the Apache NetBeans IDE interface with the following details:

- Menu Bar:** File, Edit, View, Navigate, Source, Refactor, Run, Debug, Profile, Team, Tools, Window, Help.
- Title Bar:** rifqi_pert5 - Apache NetBeans IDE 25
- Toolbar:** Includes icons for file operations like New, Open, Save, Cut, Copy, Paste, Find, and Run.
- Project Explorer (Projects Tab):** Shows the project structure with packages like la2_rp2, rifqi_pert5, and sub-packages com.mycompany.rifqi_pert5, com.mycompany.rifqi_pert5.controllers, com.mycompany.rifqi_pert5.models, and com.mycompany.rifqi_pert5.repositories.
- Code Editor (ModelMahasiswa.java):** Displays Java code for a ModelMahasiswa class. The code includes fields for id, npm, nama, semester, and ipk, and methods for setting and getting these values.
- Navigator (ModelMahasiswa.java - Navigator):** Shows the members of the ModelMahasiswa class, including its constructor and various get and set methods for id, npm, nama, semester, and ipk.
- Bottom Status Bar:** Shows system information including battery level (81%), network status, and system date/time (19:09 PM 11/15/2025).

The screenshot shows the Apache NetBeans IDE interface with the title bar "nfq1_pert5 - Apache NetBeans IDE 25". The left sidebar displays the project structure for "nfq1_pert5" under "Source Packages", including files like "MahasiswaController.java", "ModelMahasiswa.java", and "MahasiswaRepository.java". The main editor area shows the code for "ModelMahasiswa.java". The code defines a class "ModelMahasiswa" with methods for getting and setting semester, ipk, and a toString() method that concatenates id, name, npm, and semester. The code editor has syntax highlighting and line numbers from 12 to 99. The bottom status bar shows "Run (Rifqi_pert5)" and "99.1 INS Windows (CRFL)".

```
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;
}

@Override
public String toString() {
    return "Mahasiswa{" +
        "id=" + id +
        ", nama='"+ nama + '\'' +
        ", npm='"+ npm + '\'' +
        ", semester='"+ semester + '\'' +
        ", jurusan='"+ ipk + '\'' +
        '}';
}
```

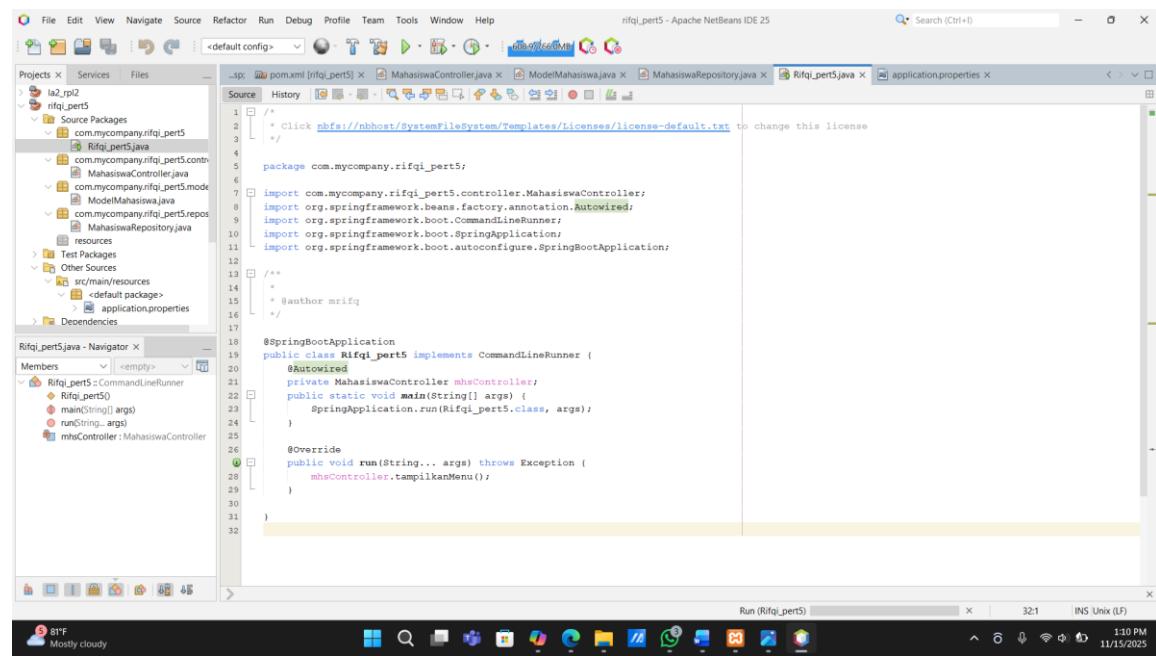
MahasiswaRepository.java

The screenshot shows the Apache NetBeans IDE interface with the following details:

- MenuBar:** File, Edit, View, Navigate, Source, Refactor, Run, Debug, Profile, Team, Tools, Window, Help.
- Toolbar:** Standard NetBeans toolbar with icons for file operations, search, and project navigation.
- Project Explorer (Projects X):** Shows the project structure:
 - Source Packages:
 - com.mycompany.rifqi_pert5 (selected)
 - Rifqi_pert5.java
 - MahasiswaController.java
 - ModelMahasiswa.java
 - MahasiswaRepository.java
 - Test Packages
 - Other Files:
 - src/main/resources (selected)
 - application.properties
 - Dependencies
- Code Editor:** The current file is MahasiswaRepository.java, which contains the following code:

```
1  /*
2  * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
3  * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Interface.java to edit this template
4  */
5
6  package com.mycompany.rifqi_pert5.repository;
7
8  import com.mycompany.rifqi_pert5.model.ModelMahasiswa;
9  import org.springframework.data.jpa.repository.JpaRepository;
10 import org.springframework.stereotype.Repository;
11
12 /**
13  * @author mrfiq
14  */
15 @Repository
16 public interface MahasiswaRepository extends JpaRepository<ModelMahasiswa, Long> {
```
- Bottom Status Bar:** Shows system status (81°F, Mostly cloudy), battery level (100%), and system date/time (11/15/2025, 10:09 PM).

Rifqi_pert5.java



```
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help rifqi_pert5 - Apache NetBeans IDE 25 <default config> Search (Ctrl+F) Projects Services Files ... -sp pom.xml [rifqi_pert5] X MahasiswaController.java X ModelMahasiswa.java X MahasiswaRepository.java X Rifqi_pert5Java X application.properties X Source History ... Rifqi_pert5 Java com.mycompany.rifqi_pert5 com.mycompany.rifqi_pert5.controllers MahasiswaController.java com.mycompany.rifqi_pert5.mode ModelMahasiswa.java com.mycompany.rifqi_pert5.repos MahasiswaRepository.java resources Test Packages Other Sources src/main/resources <default package> application.properties Dependencies Rifqi_pert5Java - Navigator Members ... Rifqi_pert5 CommandLineRunner Rifqi_pert5 main(String[] args) run(String... args) mhsController: MahasiswaController
```

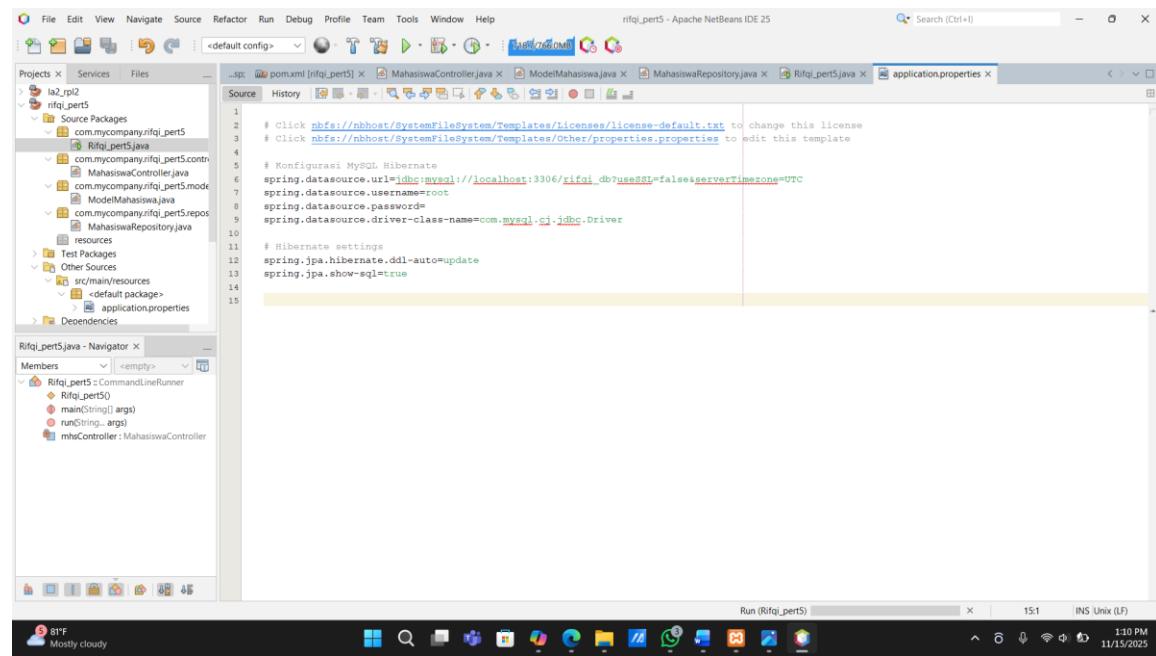
```
/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
 * Click nbfs://nbhost/SystemFileSystem/Templates/Other/properties.properties to edit this template
 */
package com.mycompany.rifqi_pert5;

import com.mycompany.rifqi_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 mrifq
 */
@SpringBootApplication
public class Rifqi_pert5 implements CommandLineRunner {
    @Autowired
    private MahasiswaController mhsController;
    public static void main(String[] args) {
        SpringApplication.run(Rifqi_pert5.class, args);
    }

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

Application.properties



```
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help rifqi_pert5 - Apache NetBeans IDE 25 <default config> Search (Ctrl+F) Projects Services Files ... -sp pom.xml [rifqi_pert5] X MahasiswaController.java X ModelMahasiswa.java X MahasiswaRepository.java X Rifqi_pert5Java X application.properties X Source History ... Rifqi_pert5 Java com.mycompany.rifqi_pert5 com.mycompany.rifqi_pert5.controllers MahasiswaController.java com.mycompany.rifqi_pert5.mode ModelMahasiswa.java com.mycompany.rifqi_pert5.repos MahasiswaRepository.java resources Test Packages Other Sources src/main/resources <default package> application.properties Dependencies Rifqi_pert5Java - Navigator Members ... Rifqi_pert5 CommandLineRunner Rifqi_pert5 main(String[] args) run(String... args) mhsController: MahasiswaController
```

```
# Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
# Click nbfs://nbhost/SystemFileSystem/Templates/Other/properties.properties to edit this template

# Konfigurasi MySQL Hibernate
spring.datasource.url=jdbc:mysql://localhost:3306/rifqi_db?useSSL=false&serverTimezone=UTC
spring.datasource.username=root
spring.datasource.password=
spring.datasource.driver-class-name=com.mysql.cj.jdbc.Driver

# Hibernate settings
spring.jpa.hibernate.ddl-auto=update
spring.jpa.show-sql=true
```

Membuat tabel di phpmyadmin

The screenshot shows the phpMyAdmin interface on a Windows desktop. The title bar says "localhost / localhost:3306 | phpMyAdmin". The main menu bar includes "Databases", "SQL", "Status", "User accounts", "Export", "Import", "Settings", "Replication", "Variables", "Charsets", "Engines", and "Plugins". Below the menu is a sub-menu for "Databases" with tabs for "Create database" (selected), "SQL", "Status", "User accounts", "Export", "Import", "Settings", "Replication", "Variables", "Charsets", "Engines", and "Plugins". A search bar is at the top right. The main content area is titled "Databases" and shows a list of existing databases: ap2b, bank, bank_51422145, information_schema, klib, la_perit3_db, mhs01, mysql, pasien_51422145, and rify_db (which is currently being created). There are checkboxes for "Check all" and "Drop" actions, and a "Create" button. At the bottom, there's a taskbar with icons for File Explorer, Task View, Start, Search, and other applications, along with system status indicators like battery level and date/time.

Output :

Menu 1 :

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.
```

Menu 2 :

Menu:

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

Pilih opsi: 2

Masukkan NPM : 51422145

Masukkan Nama : Muhammad Rifqi Dwi Putra

Masukkan Semester : 7

Masukkan IPK : 3.89

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

Mahasiswa berhasil ditambahkan.

Menu 1 :

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='51422145', nama='Muhammad Rifqi Dwi Putra', nama='7', jurusan='3.89'}
```

Menu 3 :

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 4 :

Menu:

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

Pilih opsi: 4

Keluar dari program.