

LAPORAN AKHIR PRAKTIKUM

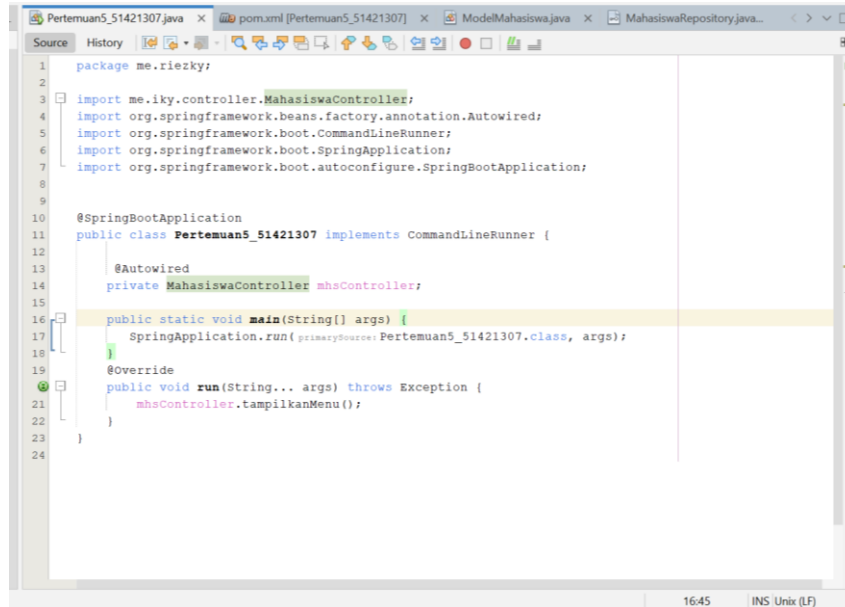
Mata Praktikum : RPL 2
Kelas : 4IA27
Praktikum ke - : 5 & 6
Tanggal : 3 November 2024
Materi :
NPM : 51421307
Nama : Riezky Arif Fadhilah
Ketua Asisten : Suryo Aji Widagdo
Jumlah Lembar : 10



LABORATORIUM TEKNIK INFORMATIKA
UNIVERSITAS GUNADARMA

Laporan Akhir 5

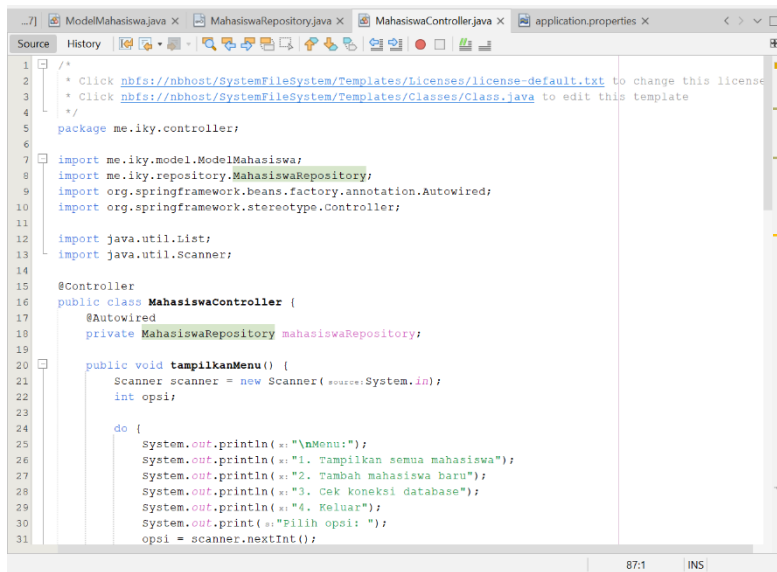
Pertemuan5_51421307



```
1 package me.riezky;
2
3 import me.iky.controller.MahasiswaController;
4 import org.springframework.beans.factory.annotation.Autowired;
5 import org.springframework.boot.CommandLineRunner;
6 import org.springframework.boot.SpringApplication;
7 import org.springframework.boot.autoconfigure.SpringBootApplication;
8
9
10 @SpringBootApplication
11 public class Pertemuan5_51421307 implements CommandLineRunner {
12
13     @Autowired
14     private MahasiswaController mhsController;
15
16     public static void main(String[] args) {
17         SpringApplication.run(Pertemuan5_51421307.class, args);
18     }
19
20     @Override
21     public void run(String... args) throws Exception {
22         mhsController.tampilkanMenu();
23     }
24 }
```

- Pertama untuk line 1 menunjukkan bahwa kelas ini berada di dalam package me.riezky.
- Selanjutnya untuk line 3 – 7 ini adalah import statements import diatas untuk Mahasiswa Controller dari package me.iky.controller, yang tampaknya akan digunakan untuk mengontrol logika terkait mahasiswa. Dan juga terdapat beberapa komponen dari Spring, seperti Autowired, CommandLineRunner, SpringApplication, dan SpringBootApplication, yang diperlukan untuk konfigurasi dan inisialisasi aplikasi Spring Boot.
- Dan untuk line 10 Anotasi @SpringBootApplication menandakan bahwa kelas ini adalah kelas utama aplikasi Spring Boot.
- Jadi pada baris line 11 disini mendeklarasikan kelas utama Pertemuan5_51421307, yang juga mengimplementasikan CommandLineRunner.
- Pada line 12 @Autowired menyuntikkan instance MahasiswaController ke dalam variabel mhsController secara otomatis.
- Selanjutnya untuk lines 14-16 terdapat metode main menjalankan aplikasi dengan SpringApplication.run, yang memulai konteks Spring Boot.
- Dan untuk lines 18-22 metode run dipanggil saat aplikasi mulai, dan menjalankan mhsController.tampilkanMenu() untuk menampilkan menu utama atau logika awal aplikasi.

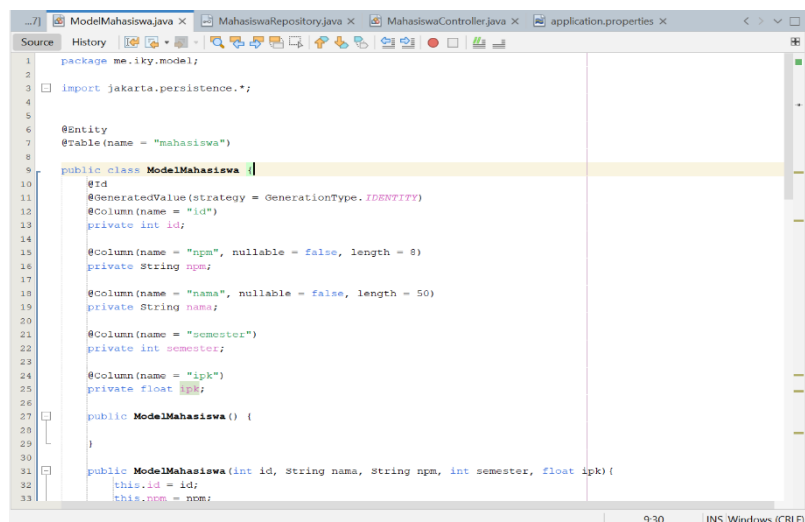
Mahasiswa Controller.java



```
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 me.iky.controller;
6
7  import me.iky.model.ModelMahasiswa;
8  import me.iky.repository.MahasiswaRepository;
9  import org.springframework.beans.factory.annotation.Autowired;
10 import org.springframework.stereotype.Controller;
11
12 import java.util.List;
13 import java.util.Scanner;
14
15 @Controller
16 public class MahasiswaController {
17     @Autowired
18     private MahasiswaRepository mahasiswaRepository;
19
20     public void tampilMenu() {
21         Scanner scanner = new Scanner(System.in);
22         int opsi;
23
24         do {
25             System.out.println("\nMenu:");
26             System.out.println("1. Tampilkan semua mahasiswa");
27             System.out.println("2. Tambah mahasiswa baru");
28             System.out.println("3. Cek koneksi database");
29             System.out.println("4. Keluar");
30             System.out.print("Pilih opsi: ");
31             opsi = scanner.nextInt();
```

- Pada baris line 5-10 yaitu mengimpor beberapa kelas dan library yang diperlukan, seperti ModelMahasiswa, MahasiswaRepository, dan kelas dari Spring untuk mengatur dependency injection (@Autowired) dan menandai kelas sebagai Controller.
- Kemudian untuk line 12 terdapat sebuah code @Controller menandai bahwa kelas MahasiswaController adalah komponen Controller dalam Spring, yang berfungsi untuk menangani logika aplikasi.
- Jika sudah untuk line 13-15 disini mendeklarasikan kelas MahasiswaController dan menggunakan anotasi @Autowired pada variabel mahasiswaRepository. Ini memungkinkan Spring untuk menyuntikkan instance MahasiswaRepository secara otomatis ke variabel tersebut, yang nantinya digunakan untuk berinteraksi dengan data mahasiswa.
- Dan pada line 17-30 yaitu mendefinisikan suatu metode yaittu tampilMenu(), yang menampilkan menu di konsol untuk mengelola data mahasiswa.

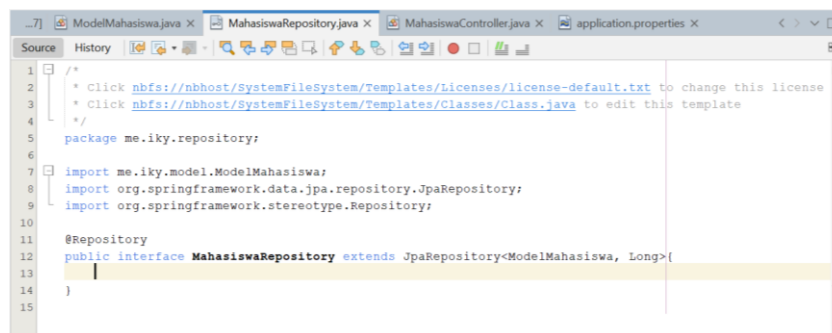
ModelMahasiswa.java



```
1  package me.iky.model;
2
3  import jakarta.persistence.*;
4
5  @Entity
6  @Table(name = "mahasiswa")
7
8  public class ModelMahasiswa {
9
10     @Id
11     @GeneratedValue(strategy = GenerationType.IDENTITY)
12     @Column(name = "id")
13     private int id;
14
15     @Column(name = "npm", nullable = false, length = 8)
16     private String npm;
17
18     @Column(name = "nama", nullable = false, length = 50)
19     private String nama;
20
21     @Column(name = "semester")
22     private int semester;
23
24     @Column(name = "ipk")
25     private float ipk;
26
27     public ModelMahasiswa() {
28
29     }
30
31     public ModelMahasiswa(int id, String nama, String npm, int semester, float ipk) {
32         this.id = id;
33         this.nama = nama;
```

- Untuk baris line 1-2 dengannnn mendefinisikan package dan mengimpor kelas yang diperlukan.
- Pada baris line 4-6 menandai kelas sebagai entitas dan menentukan nama tabel.
- Selanjutnya untuk baris line 8-30 yaitu mendefinisikan field dengan pemetaan dan batasan terhadap kolom di basis data.
- Dan pada baris line 32-34 yaitu menyediakan konstruktor default.
- Terakhir yaitu bari line 36-41 dengan menyediakan konstruktor dengan parameter untuk inisialisasi field.

MahasiswaRepository.java



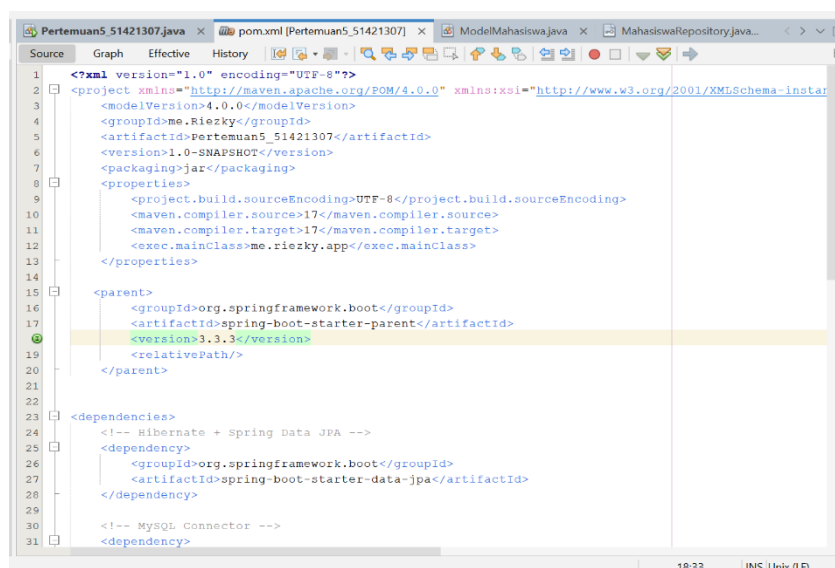
```

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 me.iky.repository;
6
7  import me.iky.model.ModelMahasiswa;
8  import org.springframework.data.jpa.repository.JpaRepository;
9  import org.springframework.stereotype.Repository;
10
11 @Repository
12 public interface MahasiswaRepository extends JpaRepository<ModelMahasiswa, Long>{
13
14 }
15

```

- Jadi pada baris line 1-2 disinni mendefinisikan package me.iky.repository.
- Jika sudah untuk baris linne 4-6 yaitu mengimpor kelas dan anotasi yang diperlukan untuk konfigurasi repository.
- Selanjutnya pada baris line 8 yaitu menandai antarmuka dengan anotasi @Repository.
- Dan pada baris line 9-11 yaitu mendefinisikan antarmuka MahasiswaRepository yang mewarisi JpaRepository, sehingga dapat melakukan operasi CRUD pada entitas ModelMahasiswa.

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"
3  <modelVersion>4.0.0</modelVersion>
4  <groupId>me.Riezky</groupId>
5  <artifactId>Pertemuan5_51421307</artifactId>
6  <version>1.0-SNAPSHOT</version>
7  <packaging>jar</packaging>
8  <properties>
9  <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
10 <maven.compiler.source>17</maven.compiler.source>
11 <maven.compiler.target>17</maven.compiler.target>
12 <exec.mainClass>me.riezky.app</exec.mainClass>
13 </properties>
14
15 <parent>
16 <groupId>org.springframework.boot</groupId>
17 <artifactId>spring-boot-starter-parent</artifactId>
18 <version>3.3.3</version>
19 <relativePath/>
20 </parent>
21
22 <dependencies>
23 <!-- Hibernate + Spring Data JPA -->
24 <dependency>
25 <groupId>org.springframework.boot</groupId>
26 <artifactId>spring-boot-starter-data-jpa</artifactId>
27 </dependency>
28
29 <!-- MySQL Connector -->
30 <dependency>
31

```

- Jadi untuk gambar diatas yaitu file pom.xml, nah pom.xml ini berfungsi untuk mengatur lingkungan Spring Boot, mengatur versi Java yang akan digunakan (Java 17), dan mengelola dependensi untuk Spring Data JPA dan MySQL. Konfigurasi ini memungkinkan aplikasi mengakses database menggunakan JPA/Hibernate dan berinteraksi dengan MySQL.

Application.properties



```

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
5  spring.datasource.url=jdbc:mysql://localhost:3306/tp1_1kv?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 spring.jpa.hibernate.ddl-auto=update
11 spring.jpa.show-sql=true
12

```

- Untuk file ini adalah konfigurasi untuk menghubungkan aplikasi Spring Boot dengan database MySQL, termasuk pengaturan URL, username, password, nama driver, serta opsi Hibernate untuk pengelolaan database.

OUTPUT

```

1. Tampilkan semua mahasiswa
2. Tambah mahasiswa baru
3. Cek koneksi database
4. Keluar
Pilih opsi: 2
Masukkan NPM : 51421307
Masukkan Nama : Riecky
Masukkan Semester : 7
Masukkan IPK : 3.8
Hibernate: insert into mahasiswa (ipk,nama,npm,semester) values (?,?,?,?)
Mahasiswa berhasil ditambahkan.

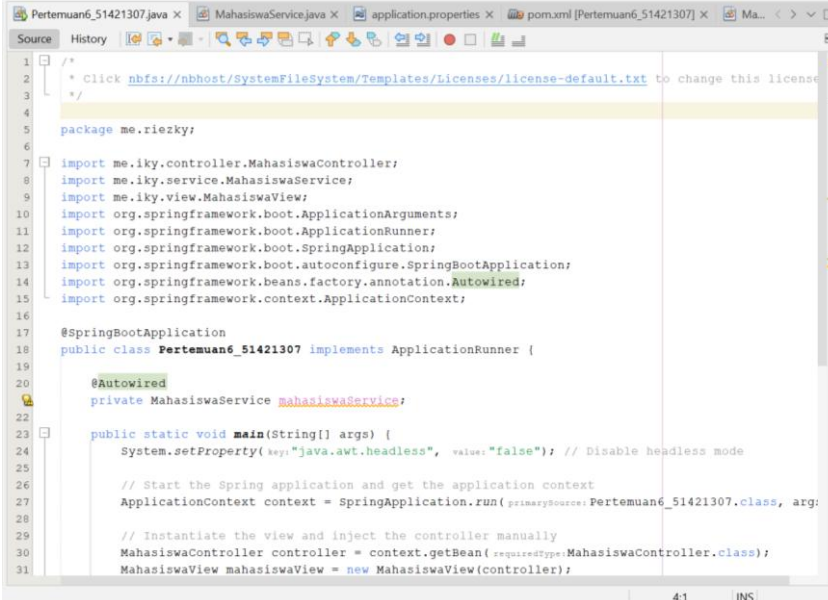
Menu:
1. Tampilkan semua mahasiswa
2. Tambah mahasiswa baru
3. Cek koneksi database
4. Keluar
Pilih opsi: 2
Masukkan NPM : 50421345
Masukkan Nama : Arif
Masukkan Semester : 7
Masukkan IPK : 3.7
Hibernate: insert into mahasiswa (ipk,nama,npm,semester) values (?,?,?,?)
Mahasiswa berhasil ditambahkan.

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, npm 'Fahri', nama '50421445', semester '7', ipk '3.8')
Mahasiswa(id=2, npm 'Riecky', nama '51421307', semester '7', ipk '3.8')
Mahasiswa(id=3, npm 'Arif', nama '50421345', semester '7', ipk '3.7')

```

Laporan Akhir 6

Pertemuan6_51421307

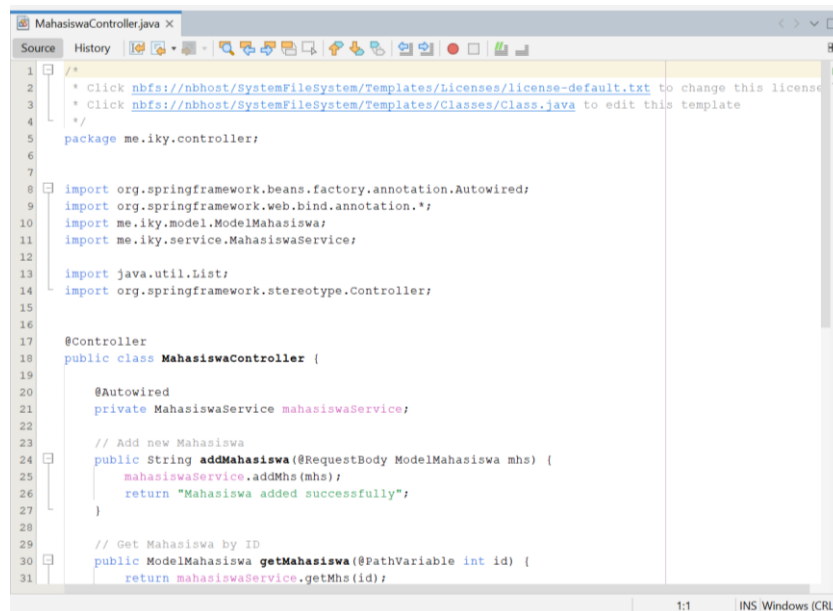


```
1  /*
2   * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
3   */
4
5  package me.riezky;
6
7  import me.iky.controller.MahasiswaController;
8  import me.iky.service.MahasiswaService;
9  import me.iky.view.MahasiswaView;
10 import org.springframework.boot.ApplicationArguments;
11 import org.springframework.boot.ApplicationRunner;
12 import org.springframework.boot.SpringApplication;
13 import org.springframework.boot.autoconfigure.SpringBootApplication;
14 import org.springframework.beans.factory.annotation.Autowired;
15 import org.springframework.context.ApplicationContext;
16
17 @SpringBootApplication
18 public class Pertemuan6_51421307 implements ApplicationRunner {
19
20     @Autowired
21     private MahasiswaService mahasiswaService;
22
23     public static void main(String[] args) {
24         System.setProperty("key:" + "java.awt.headless", value: "false"); // Disable headless mode
25
26         // Start the Spring application and get the application context
27         ApplicationContext context = SpringApplication.run(primarySource: Pertemuan6_51421307.class, arg:
28
29         // Instantiate the view and inject the controller manually
30         MahasiswaController controller = context.getBean(requiredType: MahasiswaController.class);
31         MahasiswaView mahasiswaView = new MahasiswaView(controller);
```

- Pada baris line 1-5 disini terdapat komentar lisensi yang berisi komentar lisensi default yang biasanya otomatis ditambahkan pada file Java.
- Selanjutnya untuk baris line 7 package me.riezky; yaitu mendefinisikan package me.riezky untuk mengelompokkan kelas ini dengan kelas lain dalam package yang sama.
- Nah jika sudah lanjut baris line 9-16 yaitu Import libraries berguna untuk mengimpor kelas dan anotasi yang diperlukan, seperti SpringBootApplication, Autowired, dan ApplicationContext dari Spring Framework, serta beberapa kelas aplikasi (MahasiswaController, MahasiswaService, MahasiswaView).
- Jadi pada baris line 18 @SpringBootApplication kode ini merupakan anotasi menandakan bahwa kelas ini adalah kelas utama aplikasi Spring Boot.
- Lanjut untuk baris line 19 dengan code public class Pertemuan6_51421307 implements ApplicationRunner yaitu mendeklarasikan kelas Pertemuan6_51421307 yang mengimplementasikan ApplicationRunner agar metode run dijalankan otomatis saat aplikasi Spring Boot dimulai.
- Dalam baris 21 ini terdapat code @Autowired menandakan bahwa mahasiswaService akan diinisialisasi secara otomatis oleh Spring melalui dependency injection.
- Dan lanjut baris line 22 ini merupakan private MahasiswaService mahasiswaService; yaitu mendeklarasikan variabel mahasiswaService yang akan diisi oleh Spring dengan instance dari MahasiswaService.
- Untuk baris line 24 public static void main(String[] args) terdapat deklarasi metode main, titik awal untuk menjalankan aplikasi Java.

- Dan ini baris line 25 `System.setProperty("java.awt.headless", "false");` mengatur properti sistem agar aplikasi dapat menggunakan mode tampilan (non-headless).
- Disini untuk baris line 28 yaitu dengan nama code `ApplicationContext context = SpringApplication.run(Pertemuan6_51421307.class, args);` yaitu menjalankan aplikasi Spring Boot dan mengembalikan konteks aplikasi yang berisi semua bean.
- Jadi selanjutnya untuk baris line 31: `MahasiswaController controller = context.getBean(MahasiswaController.class);` Mengambil bean MahasiswaController dari konteks aplikasi dan menyimpannya ke variabel controller.
- Terakhir line 32 `MahasiswaView mahasiswaView = new MahasiswaView(controller);` membuat instance MahasiswaView dan menginisialisasinya dengan controller sebagai parameter, memungkinkan tampilan untuk menggunakan controller.

Mahasiswa Controller.java



```

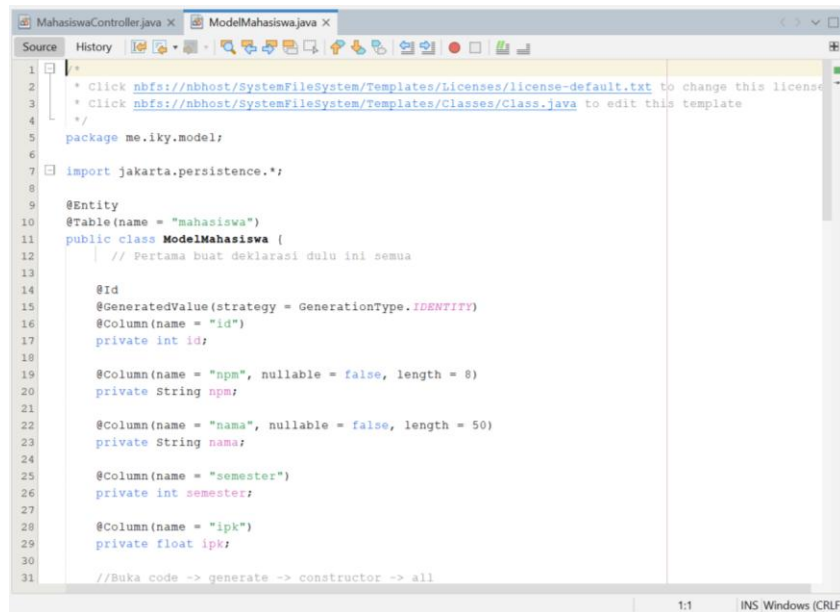
1  /*
2   * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
3   * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this template
4   */
5  package me.iky.controller;
6
7
8  import org.springframework.beans.factory.annotation.Autowired;
9  import org.springframework.web.bind.annotation.*;
10 import me.iky.model.ModelMahasiswa;
11 import me.iky.service.MahasiswaService;
12
13 import java.util.List;
14 import org.springframework.stereotype.Controller;
15
16
17 @Controller
18 public class MahasiswaController {
19
20     @Autowired
21     private MahasiswaService mahasiswaService;
22
23     // Add new Mahasiswa
24     public String addMahasiswa(@RequestBody ModelMahasiswa mhs) {
25         mahasiswaService.addMhs(mhs);
26         return "Mahasiswa added successfully";
27     }
28
29     // Get Mahasiswa by ID
30     public ModelMahasiswa getMahasiswa(@PathVariable int id) {
31         return mahasiswaService.getMhs(id);
32     }
33 }

```

- Untuk baris line 1-2 disini mendefinisikan package `me.iky.controller` dan mengimpor kelas-kelas yang diperlukan untuk fungsionalitas controller
- Selanjutnya pada baris line 8-10 yaitu menandai kelas sebagai controller Spring dengan anotasi `@Controller`. Kemudian, mendefinisikan kelas `MahasiswaController` sebagai kelas controller utama.
- Jika sudah untuk baris line 12-13 mendeklarasikan atribut `mahasiswaService` dari tipe `MahasiswaService` dan menandainya dengan `@Autowired`, yang memungkinkan Spring untuk secara otomatis menyuntikkan instans `MahasiswaService` ke dalam controller ini.
- Dan pada baris line 15-19 mendefinisikan metode `addMahasiswa` untuk menambahkan data mahasiswa baru. Metode ini menggunakan anotasi `@RequestBody` pada parameter `mhs` untuk menerima data dari permintaan HTTP POST dalam bentuk JSON yang akan diubah menjadi objek `ModelMahasiswa`.

- Untuk baris line 21-24 mendefinisikan metode getMahasiswa untuk mendapatkan data mahasiswa berdasarkan id.

ModelMahasiswa.java



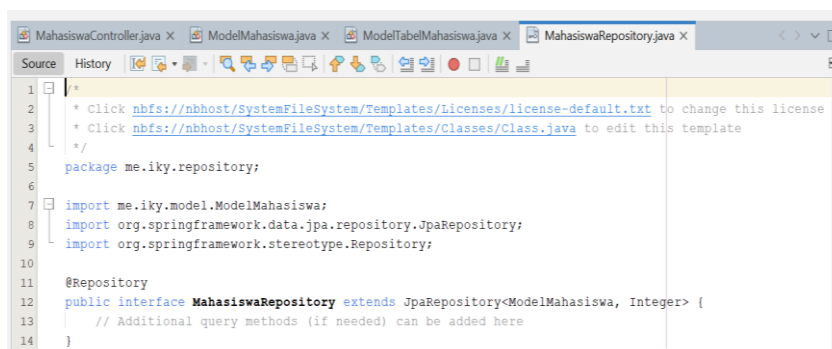
```

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 me.iky.model;
6
7  import jakarta.persistence.*;
8
9  @Entity
10 @Table(name = "mahasiswa")
11 public class ModelMahasiswa {
12     // Pertama buat deklarasi dulu ini semua
13
14     @Id
15     @GeneratedValue(strategy = GenerationType.IDENTITY)
16     @Column(name = "id")
17     private int id;
18
19     @Column(name = "npm", nullable = false, length = 8)
20     private String npm;
21
22     @Column(name = "nama", nullable = false, length = 50)
23     private String nama;
24
25     @Column(name = "semester")
26     private int semester;
27
28     @Column(name = "ipk")
29     private float ipk;
30
31     //Buka code -> generate -> constructor -> all

```

- Pada baris 1-2, kode mendeklarasikan package me.iky.repository.
- Selanjutnya pada baris line 4-6, dengan ini kode mengimpor kelas dan anotasi yang diperlukan untuk konfigurasi yang terdapat di repository, seperti jakarta.persistence.* dan org.springframework.stereotype.Repository.
- Untuk baris line 8, kode menandai antarmuka dengan anotasi @Repository, yang menandakan bahwa antarmuka ini adalah sebuah komponen repositori dalam arsitektur Spring.
- Dan pada baris 9-11, kode mendefinisikan antarmuka MahasiswaRepository yang mewarisi JpaRepository. Ini berarti antarmuka MahasiswaRepository akan mewarisi metode-metode CRUD (Create, Read, Update, Delete) dari JpaRepository untuk entitas ModelMahasiswa.

MahasiswaRepository.java



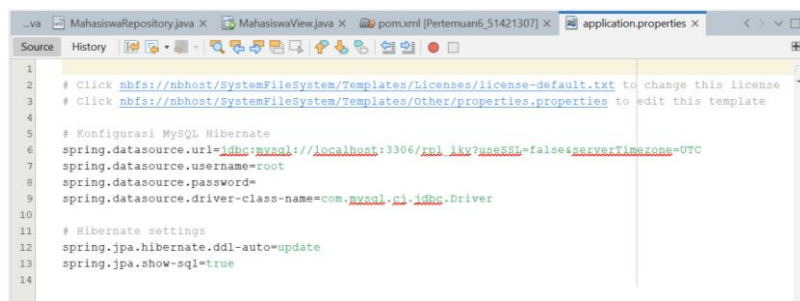
```

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 me.iky.repository;
6
7  import me.iky.model.ModelMahasiswa;
8  import org.springframework.data.jpa.repository.JpaRepository;
9  import org.springframework.stereotype.Repository;
10
11 @Repository
12 public interface MahasiswaRepository extends JpaRepository<ModelMahasiswa, Integer> {
13     // Additional query methods (if needed) can be added here
14 }

```


- Pada baris 3, kode mendeklarasikan package me.iky.repository.
- Nah untuk baris line 6-8, kode mengimpor kelas yang diperlukan untuk mendefinisikan repository, yaitu dengan beberapa seperti ModelMahasiswa, JpaRepository, dan org.springframework.stereotype.Repository.
- Dan untuk baris line 11, kode menandai antarmuka dengan anotasi @Repository, yang menandakan bahwa antarmuka ini adalah sebuah komponen repositori dalam arsitektur Spring.
- Kemudian di baris line 12-14, kode mendefinisikan antarmuka MahasiswaRepository yang mewarisi JpaRepository<ModelMahasiswa, Integer>. Ini berarti antarmuka MahasiswaRepository akan mewarisi metode-metode CRUD (Create, Read, Update, Delete) dari JpaRepository untuk entitas ModelMahasiswa, dengan tipe data primary key-nya adalah Integer.

Application.properties



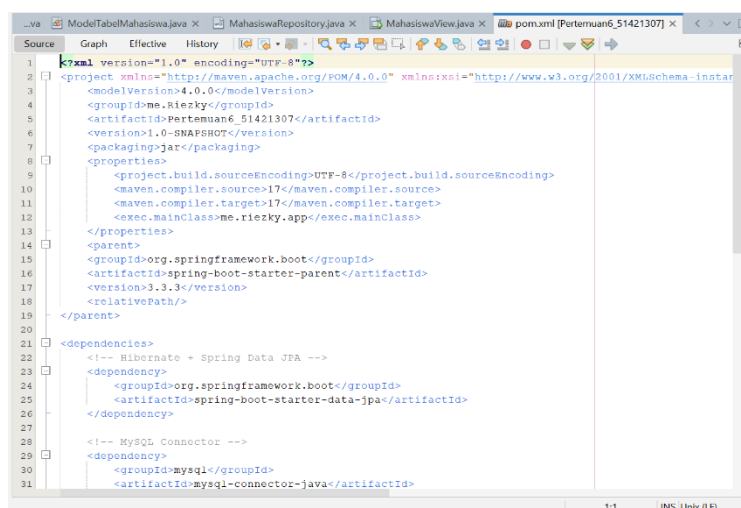
```

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 # Konfigurasi MySQL Hibernate
6 spring.datasource.url=jdbc:mysql://localhost:3306/rpl_iky?useSSL=false&serverTimezone=UTC
7 spring.datasource.username=root
8 spring.datasource.password=
9 spring.datasource.driver-class-name=com.mysql.cj.jdbc.Driver
10
11 # Hibernate settings
12 spring.jpa.hibernate.ddl-auto=update
13 spring.jpa.show-sql=true
14

```

- Jadi secara keseluruhan, file ini berisi konfigurasi yang diperlukan untuk menghubungkan aplikasi Spring Boot dengan database MySQL dan mengatur perilaku Hibernate.

Pom.xml



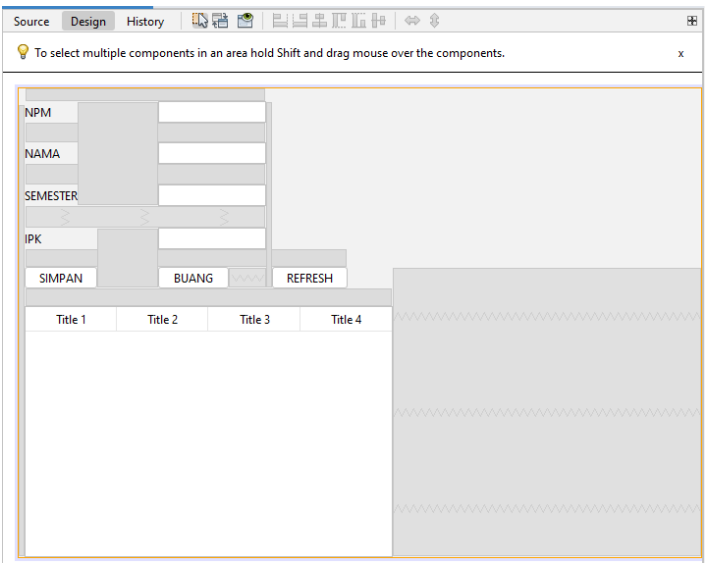
```

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"
3 <modelVersion>4.0.0</modelVersion>
4 <groupId>me.riezky</groupId>
5 <artifactId>Pertemuan6_51421307</artifactId>
6 <version>1.0-SNAPSHOT</version>
7 <packaging>jar</packaging>
8 <properties>
9 <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
10 <maven.compiler.source>17</maven.compiler.source>
11 <maven.compiler.target>17</maven.compiler.target>
12 <exec.mainClass>me.riezky.app</exec.mainClass>
13 </properties>
14 <parent>
15 <groupId>org.springframework.boot</groupId>
16 <artifactId>spring-boot-starter-parent</artifactId>
17 <version>3.3.3</version>
18 <relativePath/>
19 </parent>
20 <dependencies>
21 <!-- Hibernate + Spring Data JPA -->
22 <dependency>
23 <groupId>org.springframework.boot</groupId>
24 <artifactId>spring-boot-starter-data-jpa</artifactId>
25 </dependency>
26 <!-- MySQL Connector -->
27 <dependency>
28 <groupId>mysql</groupId>
29 <artifactId>mysql-connector-java</artifactId>
30 </dependency>
31

```

- Untuk file pom.xml code pada gambar diatas merupakan bagian dari aplikasi Spring Boot yang lebih besar, yang menggunakan Spring Data JPA untuk interaksi database juga untuk mengurangi kode boilerplate. Dan file ini menyiapkan dependensi dan konfigurasi yang diperlukan untuk membangun dan menjalankan aplikasi.

MahasiswaView.Jframeform



Output

