

ACTIVITY PERTEMUAN 5

NAMA : Felix Windriyareksa Hardyan

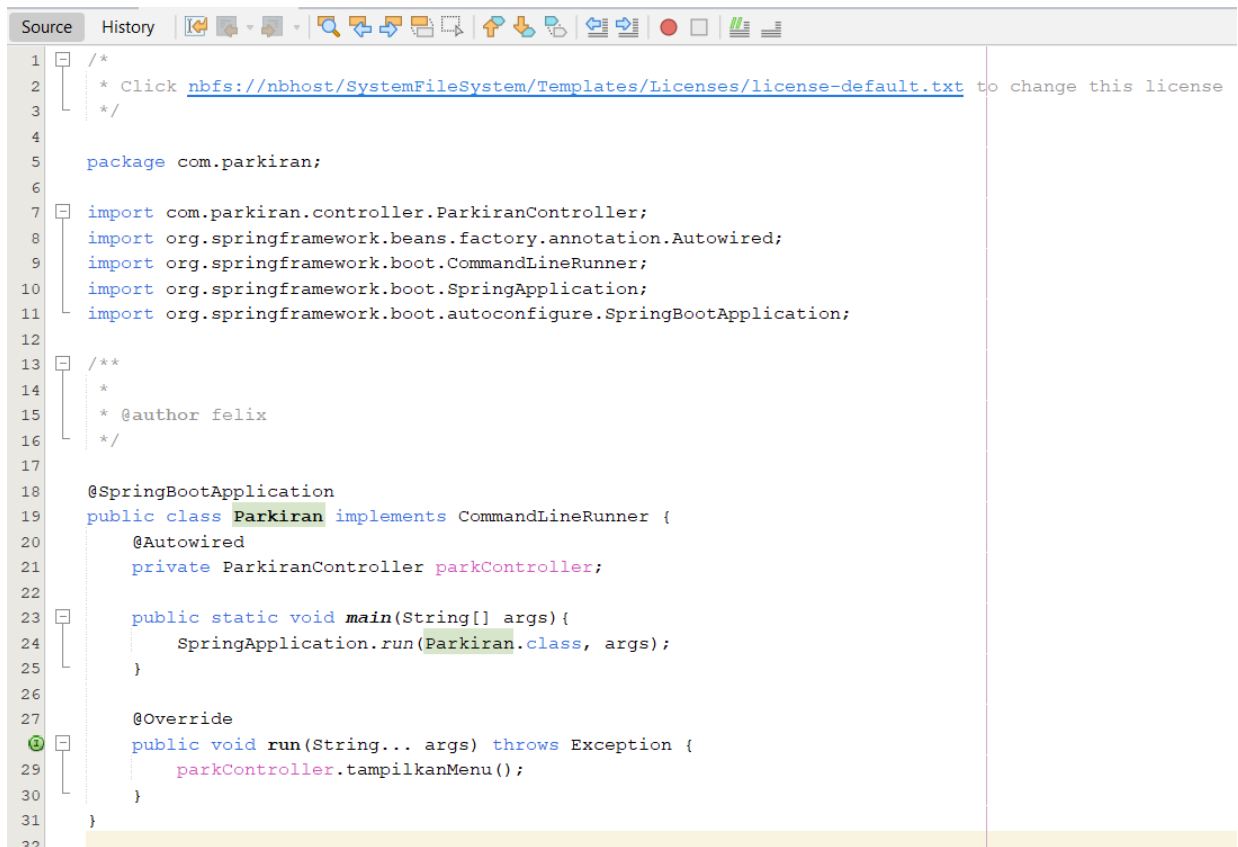
NPM : 50421506

KELAS : 4IA12

MATERI : Springboot

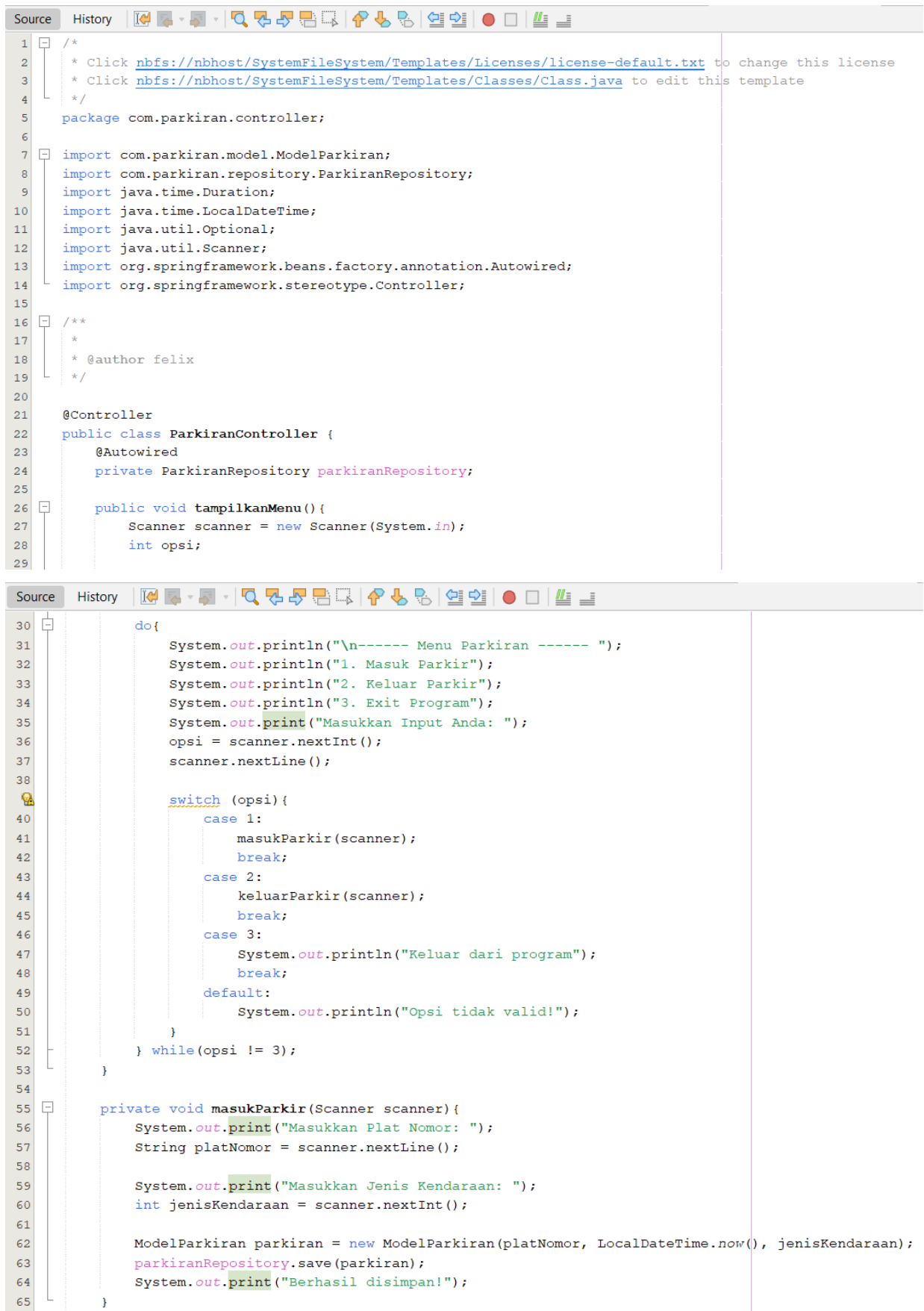
MATA PRAKTIKUM : Rekayasa Perangkat Lunak 2

1. Parkiran.java



```
1  /**
2   * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
3   */
4
5   package com.parkiran;
6
7   import com.parkiran.controller.ParkiranController;
8   import org.springframework.beans.factory.annotation.Autowired;
9   import org.springframework.boot.CommandLineRunner;
10  import org.springframework.boot.SpringApplication;
11  import org.springframework.boot.autoconfigure.SpringBootApplication;
12
13  /**
14   *
15   * @author felix
16   */
17
18  @SpringBootApplication
19  public class Parkiran implements CommandLineRunner {
20      @Autowired
21      private ParkiranController parkController;
22
23      public static void main(String[] args) {
24          SpringApplication.run(Parkiran.class, args);
25      }
26
27      @Override
28      public void run(String... args) throws Exception {
29          parkController.tampilkanMenu();
30      }
31  }
```

2. ParkiranController.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.parkiran.controller;
6
7   import com.parkiran.model.ModelParkiran;
8   import com.parkiran.repository.ParkiranRepository;
9   import java.time.Duration;
10  import java.time.LocalDateTime;
11  import java.util.Optional;
12  import java.util.Scanner;
13  import org.springframework.beans.factory.annotation.Autowired;
14  import org.springframework.stereotype.Controller;
15
16  /**
17   *
18   * @author felix
19   */
20
21  @Controller
22  public class ParkiranController {
23      @Autowired
24      private ParkiranRepository parkiranRepository;
25
26      public void tampilkanMenu() {
27          Scanner scanner = new Scanner(System.in);
28          int opsi;
29
30          do {
31              System.out.println("\n----- Menu Parkiran ----- ");
32              System.out.println("1. Masuk Parkir");
33              System.out.println("2. Keluar Parkir");
34              System.out.println("3. Exit Program");
35              System.out.print("Masukkan Input Anda: ");
36              opsi = scanner.nextInt();
37              scanner.nextLine();
38
39              switch (opsi) {
40                  case 1:
41                      masukParkir(scanner);
42                      break;
43                  case 2:
44                      keluarParkir(scanner);
45                      break;
46                  case 3:
47                      System.out.println("Keluar dari program");
48                      break;
49                  default:
50                      System.out.println("Opsi tidak valid!");
51              }
52          } while (opsi != 3);
53      }
54
55      private void masukParkir(Scanner scanner) {
56          System.out.print("Masukkan Plat Nomor: ");
57          String platNomor = scanner.nextLine();
58
59          System.out.print("Masukkan Jenis Kendaraan: ");
60          int jenisKendaraan = scanner.nextInt();
61
62          ModelParkiran parkiran = new ModelParkiran(platNomor, LocalDateTime.now(), jenisKendaraan);
63          parkiranRepository.save(parkiran);
64          System.out.print("Berhasil disimpan!");
65      }
66  }
```

```
Source History
66
67 private void keluarParkir(Scanner scanner){
68     System.out.print("Masukkan Plat Nomor: ");
69     String platNomor = scanner.nextLine();
70
71     Optional<ModelParkiran> optionalParkiran = parkiranRepository.findByPlatNomor(platNomor);
72     if (optionalParkiran.isPresent()){
73         ModelParkiran parkiran = optionalParkiran.get();
74         LocalDateTime waktuKeluar = LocalDateTime.now();
75
76         Duration durasi = Duration.between(parkiran.getWaktuMasuk(), waktuKeluar);
77         long jamParkir = durasi.toHours();
78         if (durasi.toMinutesPart() > 0){
79             jamParkir++;
80         }
81
82         int tarifPerJam = (parkiran.getJenisKendaraan() == 1) ? 5000 : 10000;
83         long totalBiaya = jamParkir * tarifPerJam;
84
85         System.out.println("Waktu Masuk: " + parkiran.getWaktuMasuk());
86         System.out.println("Waktu Keluar: " + waktuKeluar);
87         System.out.println("Lama Parkir: " + jamParkir + "jam");
88         System.out.println("Total Biaya Parkir: Rp " + totalBiaya);
89
90         parkiranRepository.delete(parkiran);
91     } else{
92         System.out.println("Kendaraan dengan plat nomor " + platNomor + " tidak ditemukan.");
93     }
94 }
95 }
```

3. ModelParkiran.java

```
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.parkiran.model;
6  import jakarta.persistence.*;
7  import java.time.LocalDateTime;
8
9  /**
10   *
11   * @author felix
12   */
13
14  @Entity
15  @Table(name = "parkiran")
16  public class ModelParkiran {
17      @Id
18      @GeneratedValue(strategy = GenerationType.IDENTITY)
19      @Column(name = "id")
20      private int id;
21
22      @Column(name = "plat_nomor", nullable = false, length = 8)
23      private String platNomor;
24
25      @Column(name = "waktu_masuk", nullable = false)
26      private LocalDateTime waktuMasuk;
27
28      @Column(name = "jenis_kendaraan", nullable = false)
29      private int jenisKendaraan;
30
31
32  public ModelParkiran(){
33
34  }
```

```
Source History [Icons]
36 public ModelParkiran(String platNomor, LocalDateTime waktuMasuk, int jenisKendaraan){
37     this.platNomor = platNomor;
38     this.waktuMasuk = waktuMasuk;
39     this.jenisKendaraan = jenisKendaraan;
40 }
41
42 public int getId() {
43     return id;
44 }
45
46 public void setId(int id) {
47     this.id = id;
48 }
49
50 public String getPlatNomor() {
51     return platNomor;
52 }
53
54 public void setPlatNomor(String platNomor) {
55     this.platNomor = platNomor;
56 }
57
58 public LocalDateTime getWaktuMasuk() {
59     return waktuMasuk;
60 }
61
62 public void setWaktuMasuk(LocalDateTime waktuMasuk) {
63     this.waktuMasuk = waktuMasuk;
64 }
65
66 public int getJenisKendaraan() {
67     return jenisKendaraan;
68 }
69
70 public void setJenisKendaraan(int jenisKendaraan) {
71     this.jenisKendaraan = jenisKendaraan;
72 }
```

4. ParkiranRepository.java

```
Source History [Icons]
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.parkiran.repository;
6
7  import com.parkiran.model.ModelParkiran;
8  import java.util.Optional;
9  import org.springframework.data.jpa.repository.JpaRepository;
10 import org.springframework.stereotype.Repository;
11
12 /**
13  *
14  * @author felix
15  */
16
17 @Repository
18 public interface ParkiranRepository extends JpaRepository<ModelParkiran, Long>{
19     public Optional<ModelParkiran> findByPlatNomor(String platNomor);
20 }
```

5. Application.properties

```
Source History
1  spring.datasource.url=jdbc:mysql://localhost:3306/spring_db?useSSL=false&serverTimezone=UTC
2  spring.datasource.username=root
3  spring.datasource.password=
4  spring.datasource.driver-class-name=com.mysql.cj.jdbc.Driver
5
6  spring.jpa.hibernate.ddl-auto=update
7  spring.jpa.show-sql=true
8
```

6. Pom.xml

```
Source Graph Effective History
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://
3  <modelVersion>4.0.0</modelVersion>
4  <groupId>com.parkiran</groupId>
5  <artifactId>parkiran</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>21</maven.compiler.source>
11 <maven.compiler.target>21</maven.compiler.target>
12 <exec.mainClass>com.parkiran.Parkiran</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 <dependency>
24 <groupId>org.springframework.boot</groupId>
25 <artifactId>spring-boot-starter-data-jpa</artifactId>
26 </dependency>
27
28 <dependency>
29 <groupId>mysql</groupId>
30 <artifactId>mysql-connector-java</artifactId>
31 <version>8.0.33</version>
32 </dependency>
33
34 <dependency>
35 <groupId>org.springframework.boot</groupId>
36 <artifactId>spring-boot-starter-web</artifactId>
37 </dependency>
38
39 <dependency>
40 <groupId>org.springframework.boot</groupId>
41 <artifactId>spring-boot-starter-test</artifactId>
42 <scope>test</scope>
43 </dependency>
44 </dependencies>
45
46 <build>
47 <plugins>
48 <plugin>
49 <groupId>org.springframework.boot</groupId>
50 <artifactId>spring-boot-maven-plugin</artifactId>
51 </plugin>
52 </plugins>
53 </build>
54 </project>
```

7. Output Program

a. Masuk Parkir

```
Output - Run (Parkiran) x
----- Menu Parkiran -----
1. Masuk Parkir
2. Keluar Parkir
3. Exit Program
Masukkan Input Anda: 1
Masukkan Plat Nomor: B1234XYZ
Masukkan Jenis Kendaraan: 2
Hibernate: insert into parkir (jenis_kendaraan,plat_nomor,waktu_masuk) values (?,?,?)
Berhasil disimpan!
```

b. Keluar Parkir

```
Output - Run (Parkiran) x
Masukkan Jenis Kendaraan: 2
Hibernate: insert into parkir (jenis_kendaraan,plat_nomor,waktu_masuk) values (?,?,?)
Berhasil disimpan!
----- Menu Parkiran -----
1. Masuk Parkir
2. Keluar Parkir
3. Exit Program
Masukkan Input Anda: 2
Masukkan Plat Nomor: B1234XYZ
Hibernate: select mpl_0.id,mpl_0.jenis_kendaraan,mpl_0.plat_nomor,mpl_0.waktu_masuk from parkir mpl_0 where mpl_0.plat_nomor=?
Waktu Masuk: 2024-11-15T11:30:35
Waktu Keluar: 2024-11-15T14:30:57.132862500
Lama Parkir: 3jam
Total Biaya Parkir: Rp 30000
Hibernate: select mpl_0.id,mpl_0.jenis_kendaraan,mpl_0.plat_nomor,mpl_0.waktu_masuk from parkir mpl_0 where mpl_0.id=?
Hibernate: delete from parkir where id=?
```

c. Exit Program

```
----- Menu Parkiran -----
1. Masuk Parkir
2. Keluar Parkir
3. Exit Program
Masukkan Input Anda: 3
Keluar dari program
```

8. Phpmyadmin

The screenshot shows the phpMyAdmin interface for a database named 'spring_db'. The 'parkiran' table is selected, and the 'UPDATE' query has been executed successfully, affecting 1 row. The table structure is as follows:

id	jenis_kendaraan	plat_nomor	waktu_masuk
3	2	B1234XYZ	2024-11-15 04:30:35.000000

The interface also shows the 'Query results operations' section with options to print, copy to clipboard, export, display chart, and create view. A console window is visible at the bottom.