

LAPORAN AKHIR PRAKTIKUM

Mata Praktikum : Rekayasa Perangkat Lunak 2

Kelas : 4IA11

Praktikum ke- 5

Tanggal : 15 November 2024

Materi : parkiran

NPM : 50421786

Nama : Marsillo Dito Saputra

Ketua Asisten :

Paraf Asisten :

Nama Asisten : Dimas Renaldy Sriwirawan

Jumlah Lembar : Lembar

LABORATORIUM TEKNIK INFORMATIKA

UNIVERSITAS GUNADARMA

2024

LISTING PROGRAM

(Jawab Soal Untuk Laporan Akhir)

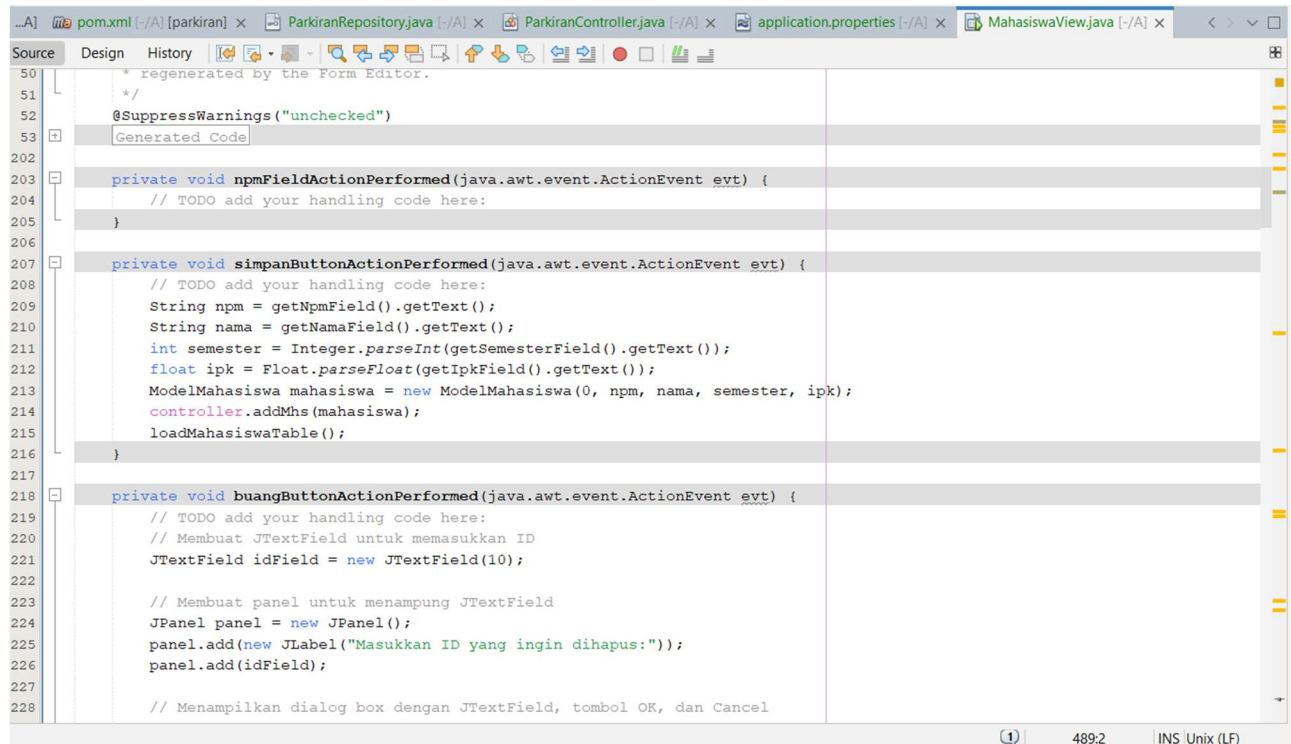
Soal Essay

1. Apa itu spring boot framework ?
2. Apa itu aspect oriented programming ?
3. Jelaskan dengan bahasa anda sendiri perbedaan antara codingan pertemuan 4 dan pertemuan 5 (boleh dibedakan menggunakan screenshot dan ada penjelasan dibawahnya)

Jawab :

1. Spring Boot adalah sebuah framework open-source yang digunakan untuk membangun aplikasi Java dengan cepat dan mudah. Framework ini dibuat untuk memudahkan pengembangan aplikasi berbasis Spring, dengan mengurangi kebutuhan konfigurasi manual yang biasanya diperlukan pada proyek Spring standar. Spring Boot menyediakan berbagai fitur seperti auto-configuration, embedded web server, dan starter templates, sehingga memungkinkan developer untuk membuat aplikasi dengan lebih sedikit konfigurasi. Tujuan utama dari Spring Boot adalah untuk mempercepat proses pengembangan aplikasi Java dengan menyederhanakan setup dan meningkatkan efisiensi pengelolaan dependensi.
2. Aspect-Oriented Programming (AOP) adalah paradigma pemrograman yang memungkinkan pemisahan kode yang memiliki kepedulian transversal (cross-cutting concerns), seperti logging, security, dan transaksi, dari logika bisnis utama aplikasi. Dengan AOP, fitur-fitur ini bisa dikelola secara terpisah dan diterapkan pada berbagai bagian program tanpa mengubah logika inti. AOP menggunakan konsep "aspect" (sebuah modul yang mengenkapsulasi kepedulian transversal) yang diterapkan pada kode menggunakan "advice" yang terhubung ke titik-titik tertentu dalam eksekusi aplikasi, yang disebut "join points". Hal ini memudahkan pemeliharaan dan pengembangan aplikasi dengan memisahkan fungsionalitas yang tidak terkait langsung dengan logika utama aplikasi.

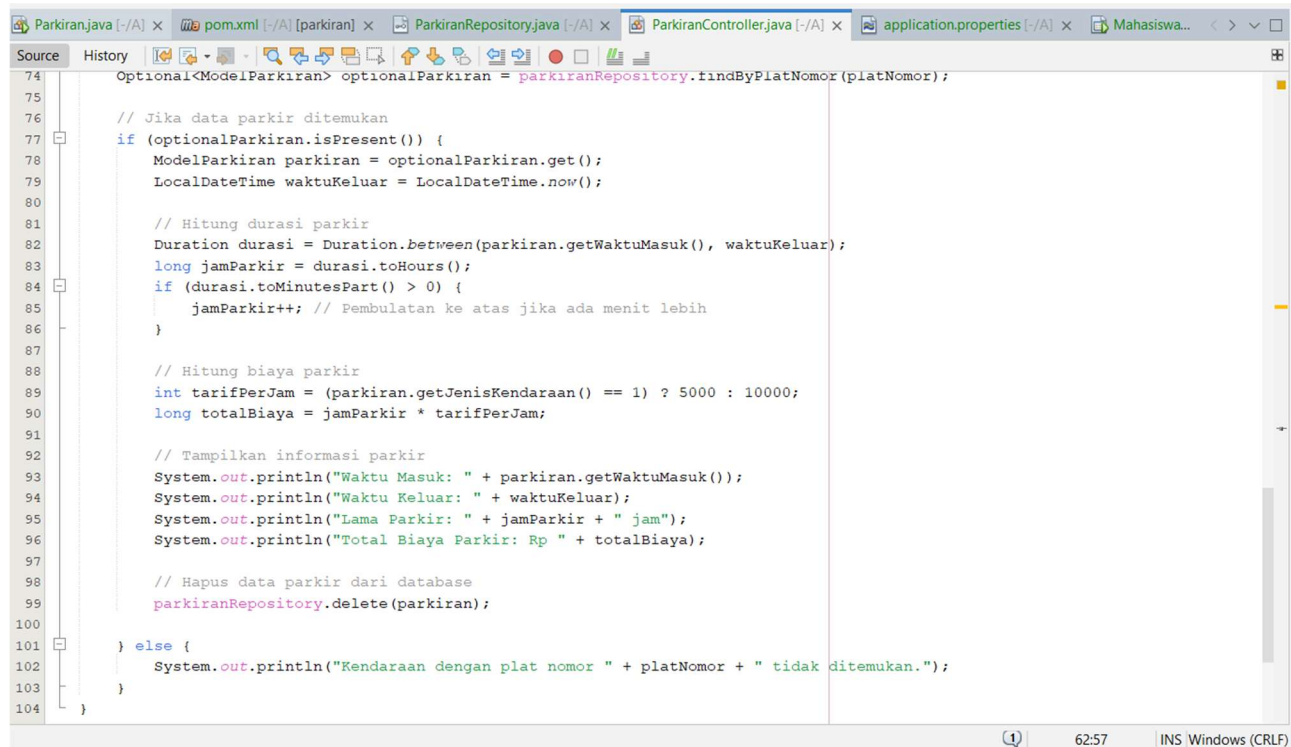
3.



```
50  * regenerated by the Form Editor.
51  */
52  @SuppressWarnings("unchecked")
53  Generated Code
202
203  private void npmFieldActionPerformed(java.awt.event.ActionEvent evt) {
204      // TODO add your handling code here:
205  }
206
207  private void simpanButtonActionPerformed(java.awt.event.ActionEvent evt) {
208      // TODO add your handling code here:
209      String npm = getNpmField().getText();
210      String nama = getNamaField().getText();
211      int semester = Integer.parseInt(getSemesterField().getText());
212      float ipk = Float.parseFloat(getIpkField().getText());
213      ModelMahasiswa mahasiswa = new ModelMahasiswa(0, npm, nama, semester, ipk);
214      controller.addMhs(mahasiswa);
215      loadMahasiswaTable();
216  }
217
218  private void buangButtonActionPerformed(java.awt.event.ActionEvent evt) {
219      // TODO add your handling code here:
220      // Membuat JTextField untuk memasukkan ID
221      JTextField idField = new JTextField(10);
222
223      // Membuat panel untuk menampung JTextField
224      JPanel panel = new JPanel();
225      panel.add(new JLabel("Masukkan ID yang ingin dihapus:"));
226      panel.add(idField);
227
228      // Menampilkan dialog box dengan JTextField, tombol OK, dan Cancel
```

Entitas utama adalah mahasiswa, dengan atribut seperti:

- NPM: Nomor Pokok Mahasiswa.
- Nama: Nama mahasiswa.
- Semester: Semester mahasiswa.
- IPK: Indeks Prestasi Kumulatif.



```
74  Optional<ModelParkiran> optionalParkiran = parkiranRepository.findByIdPlatNomor(platNomor);
75
76  // Jika data parkir ditemukan
77  if (optionalParkiran.isPresent()) {
78      ModelParkiran parkir = optionalParkiran.get();
79      LocalDateTime waktuKeluar = LocalDateTime.now();
80
81      // Hitung durasi parkir
82      Duration durasi = Duration.between(parkir.getWaktuMasuk(), waktuKeluar);
83      long jamParkir = durasi.toHours();
84      if (durasi.toMinutesPart() > 0) {
85          jamParkir++; // Pembulatan ke atas jika ada menit lebih
86      }
87
88      // Hitung biaya parkir
89      int tarifPerJam = (parkir.getJenisKendaraan() == 1) ? 5000 : 10000;
90      long totalBiaya = jamParkir * tarifPerJam;
91
92      // Tampilkan informasi parkir
93      System.out.println("Waktu Masuk: " + parkir.getWaktuMasuk());
94      System.out.println("Waktu Keluar: " + waktuKeluar);
95      System.out.println("Lama Parkir: " + jamParkir + " jam");
96      System.out.println("Total Biaya Parkir: Rp " + totalBiaya);
97
98      // Hapus data parkir dari database
99      parkirRepository.delete(parkir);
100
101  } else {
102      System.out.println("Kendaraan dengan plat nomor " + platNomor + " tidak ditemukan.");
103  }
104  }
```

Entitas utama biasanya adalah kendaraan atau parkir, dengan atribut seperti:

- Nomor Kendaraan: Identitas kendaraan.
- Waktu Masuk/Keluar: Informasi waktu parkir.
- Jenis Kendaraan: Motor atau mobil.
- Tarif Parkir: Perhitungan biaya berdasarkan waktu parkir.

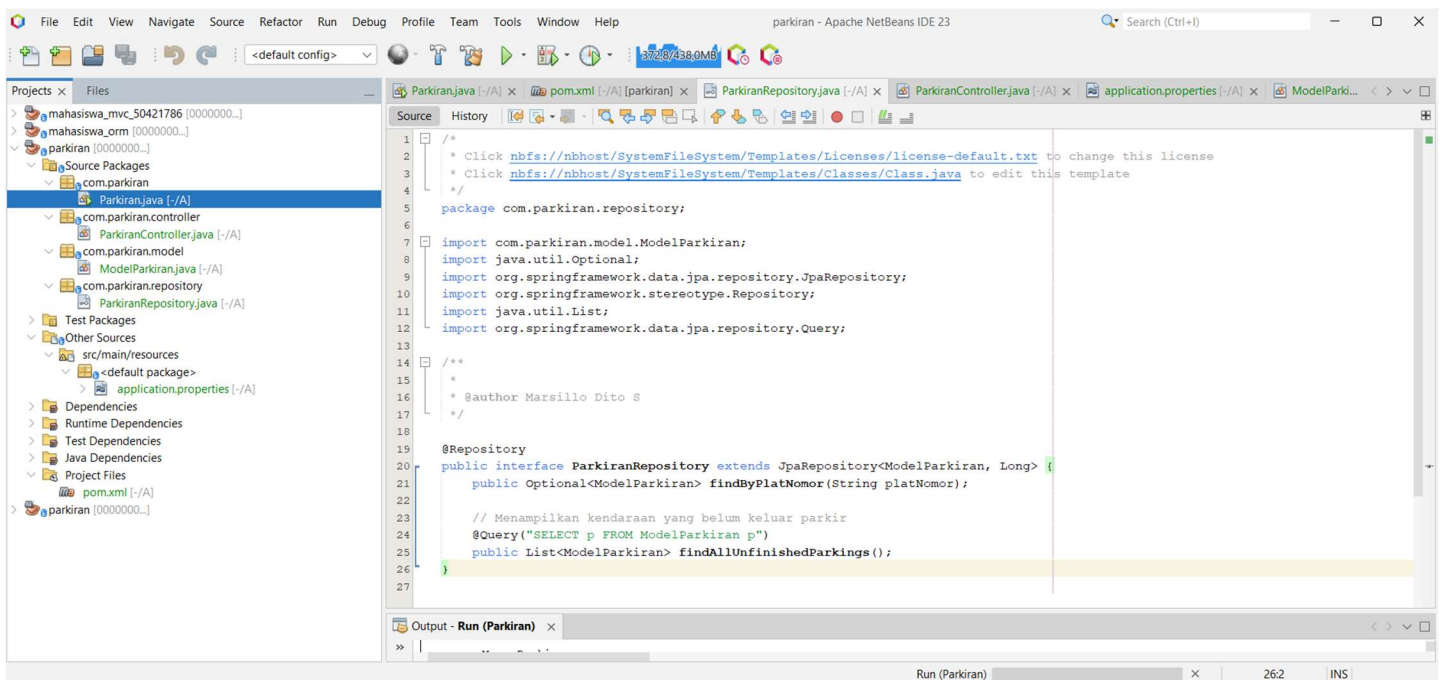
Soal Koding

1. Buatlah fitur baru untuk menampilkan semua kendaraan yang masuk parkir dan belum keluar
2. Ubahlah output agar menampilkan Motor untuk tipe kendaraan 1 dan Mobil untuk tipe kendaraan 2

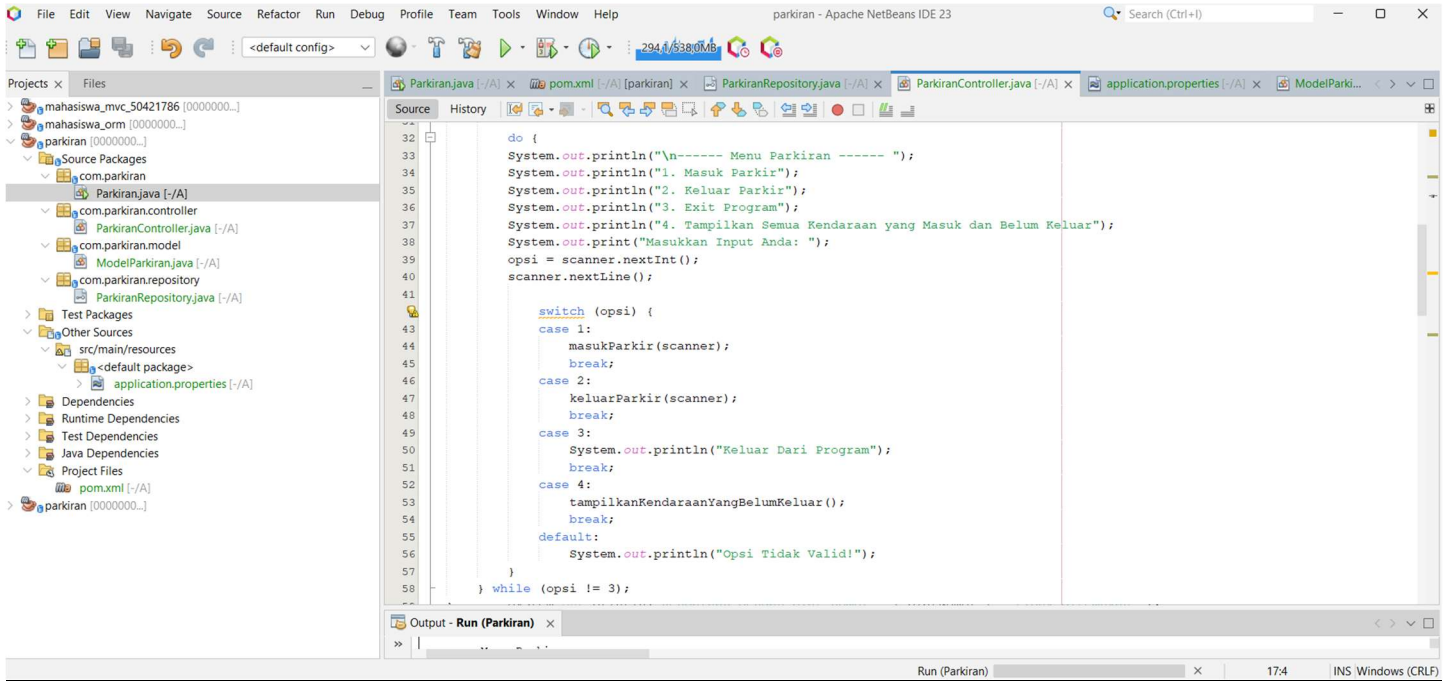
Jawab :

1.

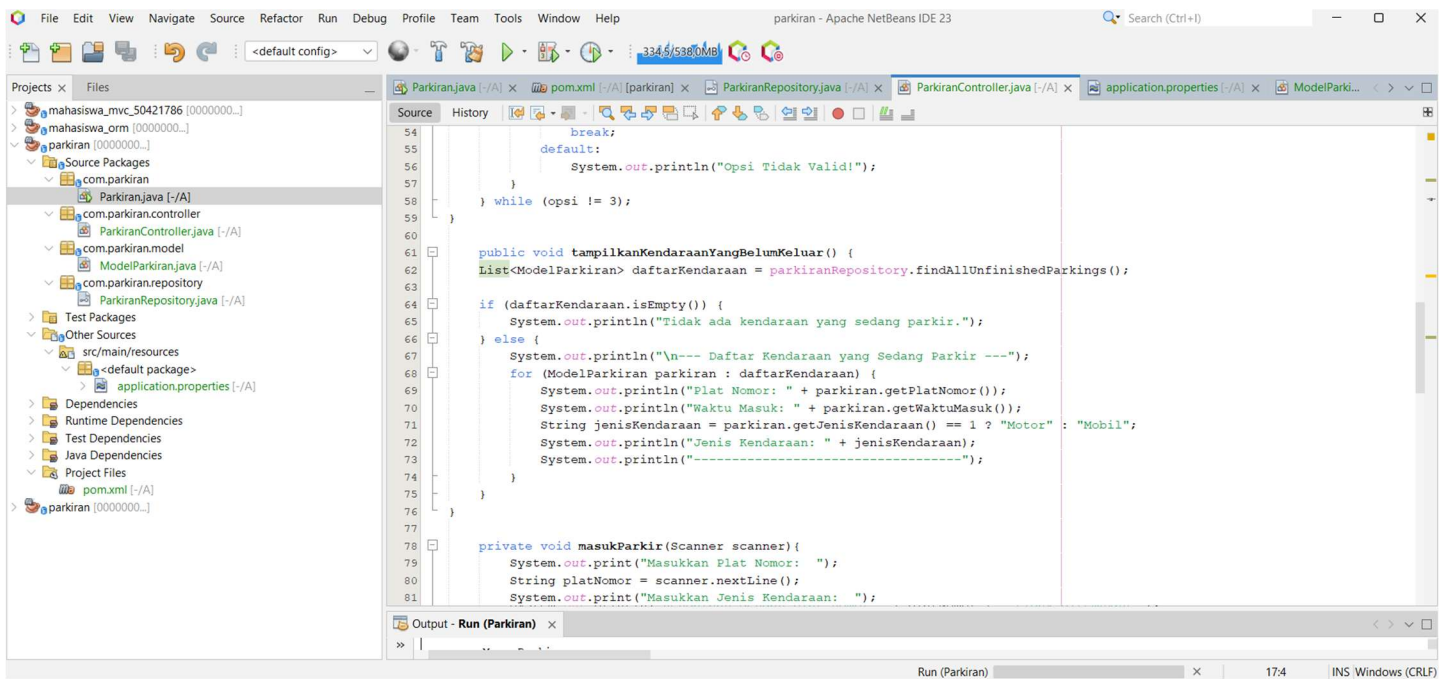
PROGRAM pada ParkiranRepository



PROGRAM pada ParkiranController



```
32 do {
33     System.out.println("\n----- Menu Parkiran ----- ");
34     System.out.println("1. Masuk Parkir");
35     System.out.println("2. Keluar Parkir");
36     System.out.println("3. Exit Program");
37     System.out.println("4. Tampilkan Semua Kendaraan yang Masuk dan Belum Keluar");
38     System.out.print("Masukkan Input Anda: ");
39     opsi = scanner.nextInt();
40     scanner.nextLine();
41
42     switch (opsi) {
43         case 1:
44             masukParkir(scanner);
45             break;
46         case 2:
47             keluarParkir(scanner);
48             break;
49         case 3:
50             System.out.println("Keluar Dari Program");
51             break;
52         case 4:
53             tampilkanKendaraanYangBelumKeluar();
54             break;
55         default:
56             System.out.println("Opsi Tidak Valid!");
57     }
58 } while (opsi != 3);
```



```
54     break;
55     default:
56         System.out.println("Opsi Tidak Valid!");
57     }
58 } while (opsi != 3);
59
60
61 public void tampilkanKendaraanYangBelumKeluar() {
62     List<ModelParkiran> daftarKendaraan = parkiranRepository.findAllUnfinishedParkings();
63
64     if (daftarKendaraan.isEmpty()) {
65         System.out.println("Tidak ada kendaraan yang sedang parkir.");
66     } else {
67         System.out.println("\n--- Daftar Kendaraan yang Sedang Parkir ---");
68         for (ModelParkiran parkir : daftarKendaraan) {
69             System.out.println("Plat Nomor: " + parkir.getPlatNomor());
70             System.out.println("Waktu Masuk: " + parkir.getWaktuMasuk());
71             String jenisKendaraan = parkir.getJenisKendaraan() == 1 ? "Motor" : "Mobil";
72             System.out.println("Jenis Kendaraan: " + jenisKendaraan);
73             System.out.println("-----");
74         }
75     }
76 }
77
78 private void masukParkir(Scanner scanner) {
79     System.out.print("Masukkan Plat Nomor: ");
80     String platNomor = scanner.nextLine();
81     System.out.print("Masukkan Jenis Kendaraan: ");
```

OUTPUT YANG DIHASILKAN jika tidak ada kendaraan yang masuk

```
Output - Run (Parkiran) x
2024-11-20T18:52:38.966+07:00 INFO 6832 --- [main] w.s.c.ServletWebServerApplicationContext : Root WebApplicationContext: initialize
2024-11-20T18:52:39.140+07:00 INFO 6832 --- [main] o.hibernate.jpa.internal.util.LogHelper : HHH000204: Processing PersistenceUnitI
2024-11-20T18:52:39.200+07:00 INFO 6832 --- [main] org.hibernate.Version : HHH000412: Hibernate ORM core version
2024-11-20T18:52:39.236+07:00 INFO 6832 --- [main] o.h.c.internal.RegionFactoryInitiator : HHH000026: Second-level cache disabled
2024-11-20T18:52:39.551+07:00 INFO 6832 --- [main] o.s.o.j.p.SpringPersistenceUnitInfo : No LoadTimeWeaver setup: ignoring JPA
2024-11-20T18:52:39.582+07:00 INFO 6832 --- [main] com.zaxxer.hikari.HikariDataSource : HikariPool-1 - Starting...
2024-11-20T18:52:39.757+07:00 INFO 6832 --- [main] com.zaxxer.hikari.pool.HikariPool : HikariPool-1 - Added connection com.my
2024-11-20T18:52:39.760+07:00 INFO 6832 --- [main] com.zaxxer.hikari.HikariDataSource : HikariPool-1 - Start completed.
2024-11-20T18:52:39.815+07:00 WARN 6832 --- [main] org.hibernate.dialect.Dialect : HHH000511: The 5.5.5 version for [org.
2024-11-20T18:52:40.540+07:00 INFO 6832 --- [main] o.h.e.t.j.p.i.JtaPlatformInitiator : HHH000489: No JTA platform available (
2024-11-20T18:52:40.591+07:00 INFO 6832 --- [main] j.LocalContainerEntityManagerFactoryBean : Initialized JPA EntityManagerFactory f
2024-11-20T18:52:40.863+07:00 INFO 6832 --- [main] o.s.d.j.r.query.QueryEnhancerFactory : Hibernate is in classpath; If applicab
2024-11-20T18:52:41.295+07:00 WARN 6832 --- [main] JpaBaseConfiguration$JpaWebConfiguration : spring.jpa.open-in-view is enabled by
2024-11-20T18:52:41.655+07:00 INFO 6832 --- [main] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat started on port 8082 (http) wit
2024-11-20T18:52:41.666+07:00 INFO 6832 --- [main] com.parkiran.Parkiran : Started Parkiran in 4.524 seconds (pro

----- Menu Parkiran -----
1. Masuk Parkir
2. Keluar Parkir
3. Exit Program
4. Tampilkan Semua Kendaraan yang Masuk dan Belum Keluar
Masukkan Input Anda: 4
Tidak ada kendaraan yang sedang parkir.
```

OUTPUT YANG DIHASILKAN jika ada kendaraan yang masuk

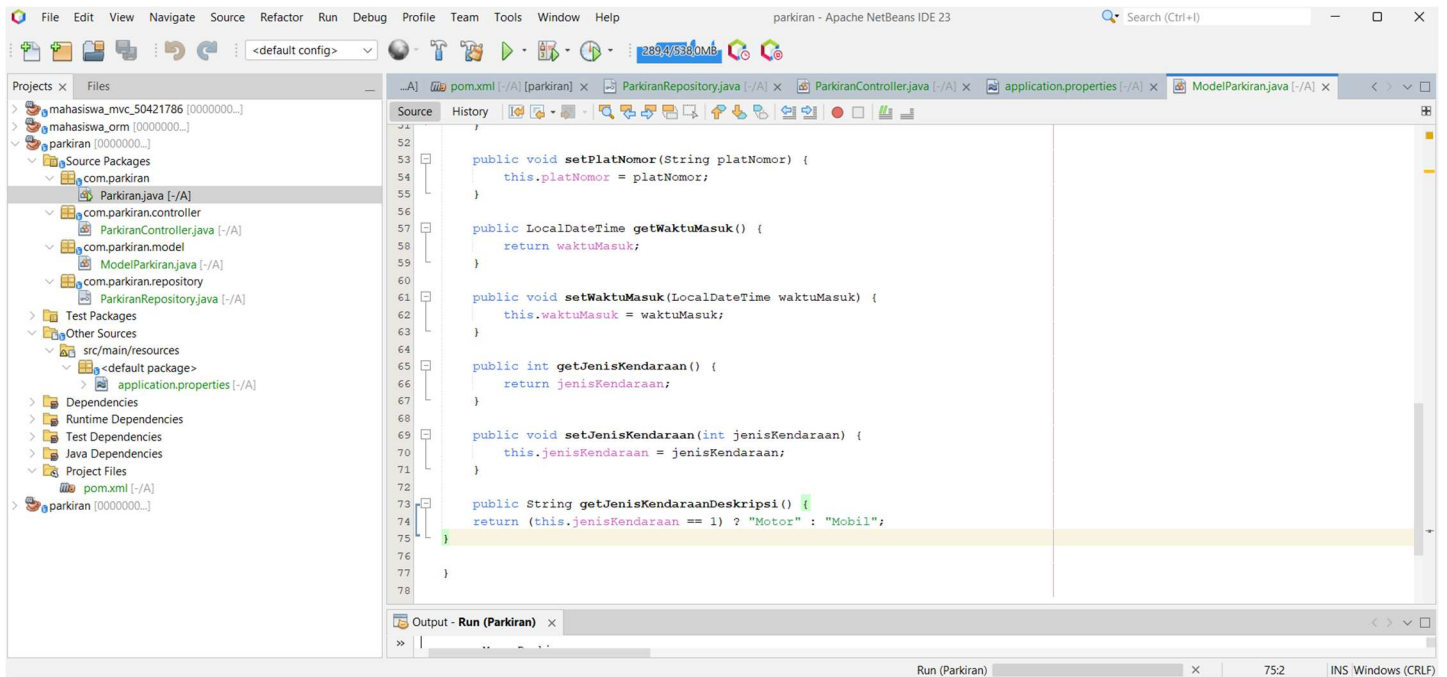
```
Parkiran.java [-/A] x pom.xml [-/A] [parkiran] x ParkiranRepository.java [-/A] x ParkiranController.java [-/A] x application.properties [-/A] x ModelParki.
Source History
106 // Hitung biaya parkir
107 int tarifPerJam = (parkiran.getJenisKendaraan() == 1) ? 5000 : 10000;
108 long totalBiaya = jamParkir * tarifPerJam;
109
110 // Tampilkan informasi parkir
111

Output - Run (Parkiran) x
----- Menu Parkiran -----
1. Masuk Parkir
2. Keluar Parkir
3. Exit Program
4. Tampilkan Semua Kendaraan yang Masuk dan Belum Keluar
Masukkan Input Anda: 4

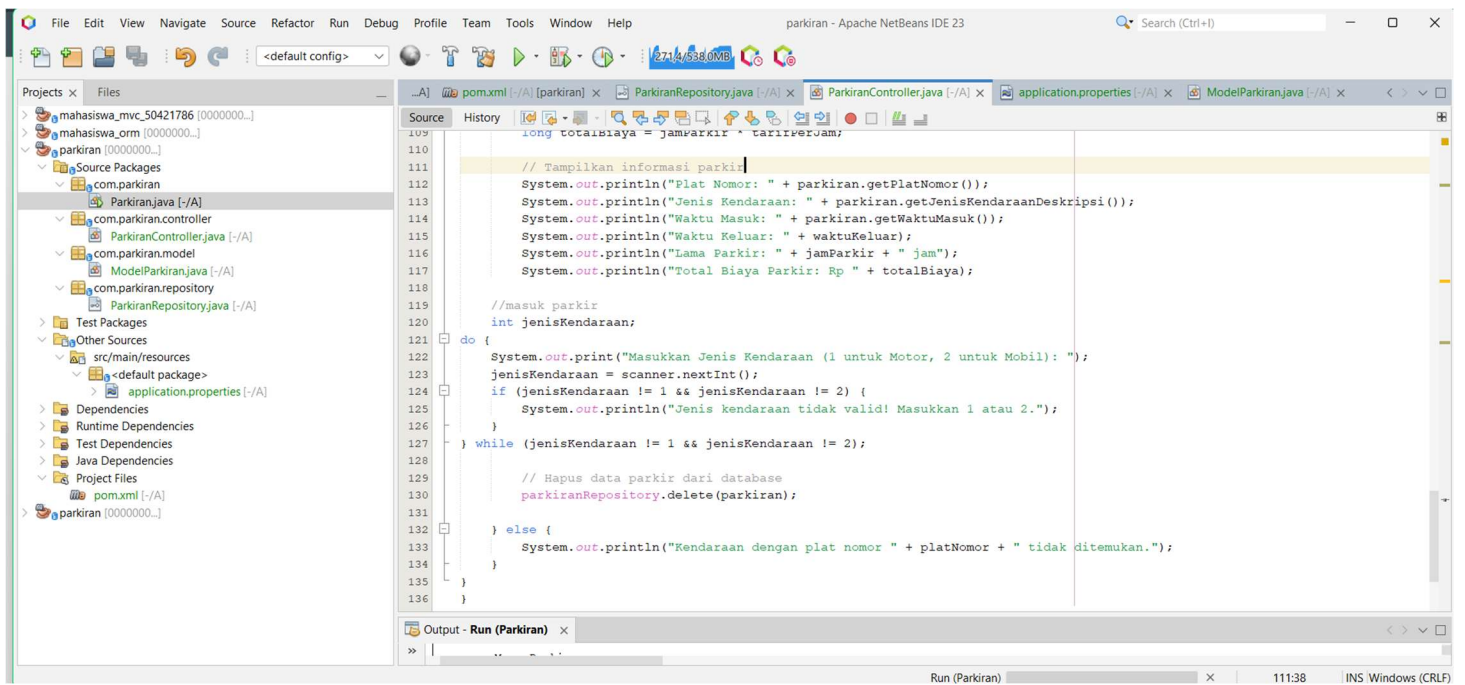
--- Daftar Kendaraan yang Sedang Parkir ---
Plat Nomor: B9999XXY
Waktu Masuk: 2024-11-20T17:54:03
Jenis Kendaraan: Motor
-----
Plat Nomor: B8888XXX
Waktu Masuk: 2024-11-20T17:54:22
Jenis Kendaraan: Mobil
-----
```


2.

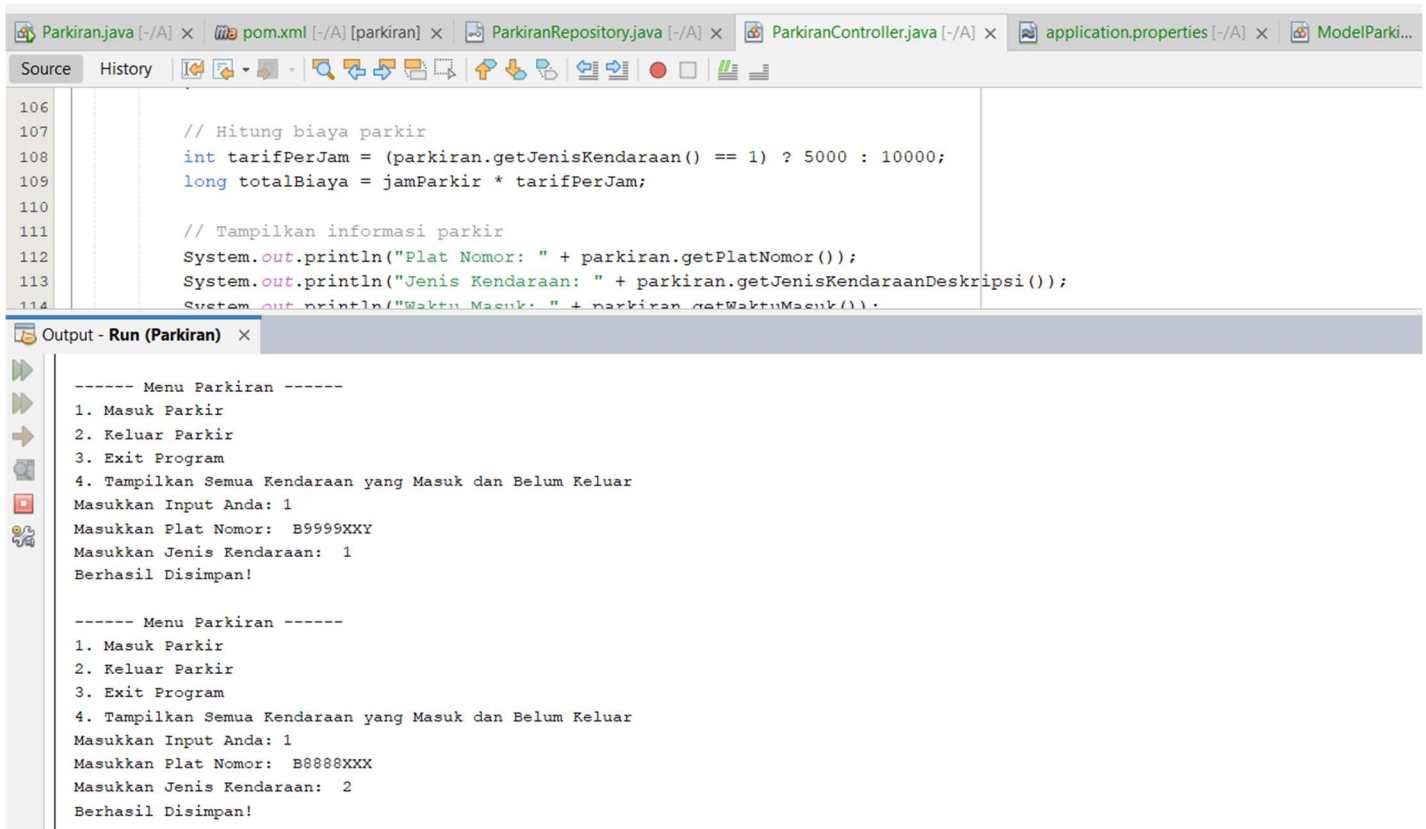
PROGRAM pada ModelParkiran



PROGRAM pada ParkiranController



OUTPUT kendaraan masuk parkir



The screenshot shows an IDE with several tabs: `Parkiran.java`, `pom.xml`, `ParkiranRepository.java`, `ParkiranController.java`, `application.properties`, and `ModelParki...`. The `ParkiranController.java` tab is active, showing the following code:

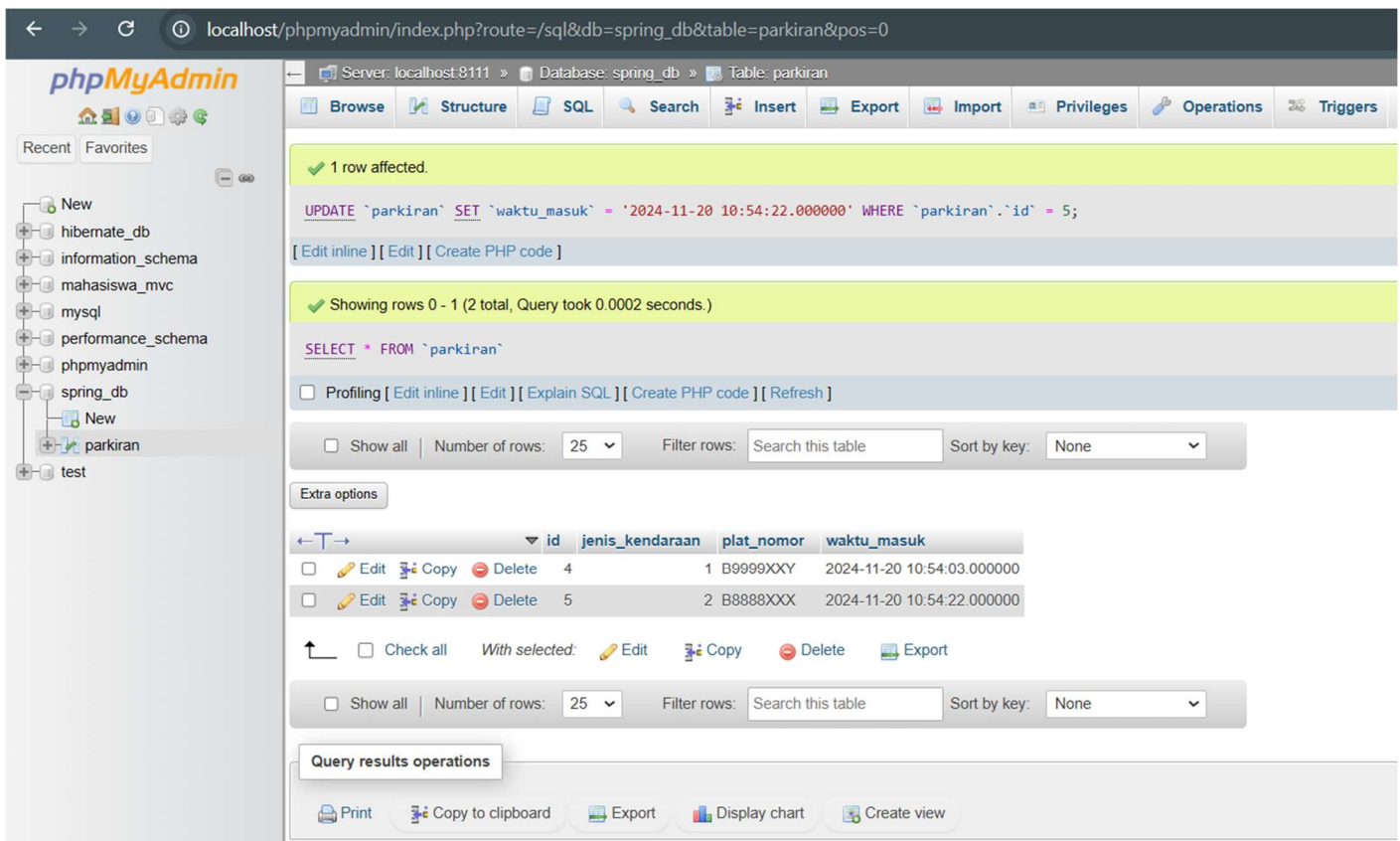
```
106
107 // Hitung biaya parkir
108 int tarifPerJam = (parkiran.getJenisKendaraan() == 1) ? 5000 : 10000;
109 long totalBiaya = jamParkir * tarifPerJam;
110
111 // Tampilkan informasi parkir
112 System.out.println("Plat Nomor: " + parkiran.getPlatNomor());
113 System.out.println("Jenis Kendaraan: " + parkiran.getJenisKendaraanDeskripsi());
114 System.out.println("Waktu Masuk: " + parkiran.getWaktuMasuk());
```

Below the code, the `Output - Run (Parkiran)` window shows the program's execution:

```
----- Menu Parkiran -----
1. Masuk Parkir
2. Keluar Parkir
3. Exit Program
4. Tampilkan Semua Kendaraan yang Masuk dan Belum Keluar
Masukkan Input Anda: 1
Masukkan Plat Nomor: B9999XXY
Masukkan Jenis Kendaraan: 1
Berhasil Disimpan!

----- Menu Parkiran -----
1. Masuk Parkir
2. Keluar Parkir
3. Exit Program
4. Tampilkan Semua Kendaraan yang Masuk dan Belum Keluar
Masukkan Input Anda: 1
Masukkan Plat Nomor: B8888XXX
Masukkan Jenis Kendaraan: 2
Berhasil Disimpan!
```

OUTPUT database pada data parkir kendaraan



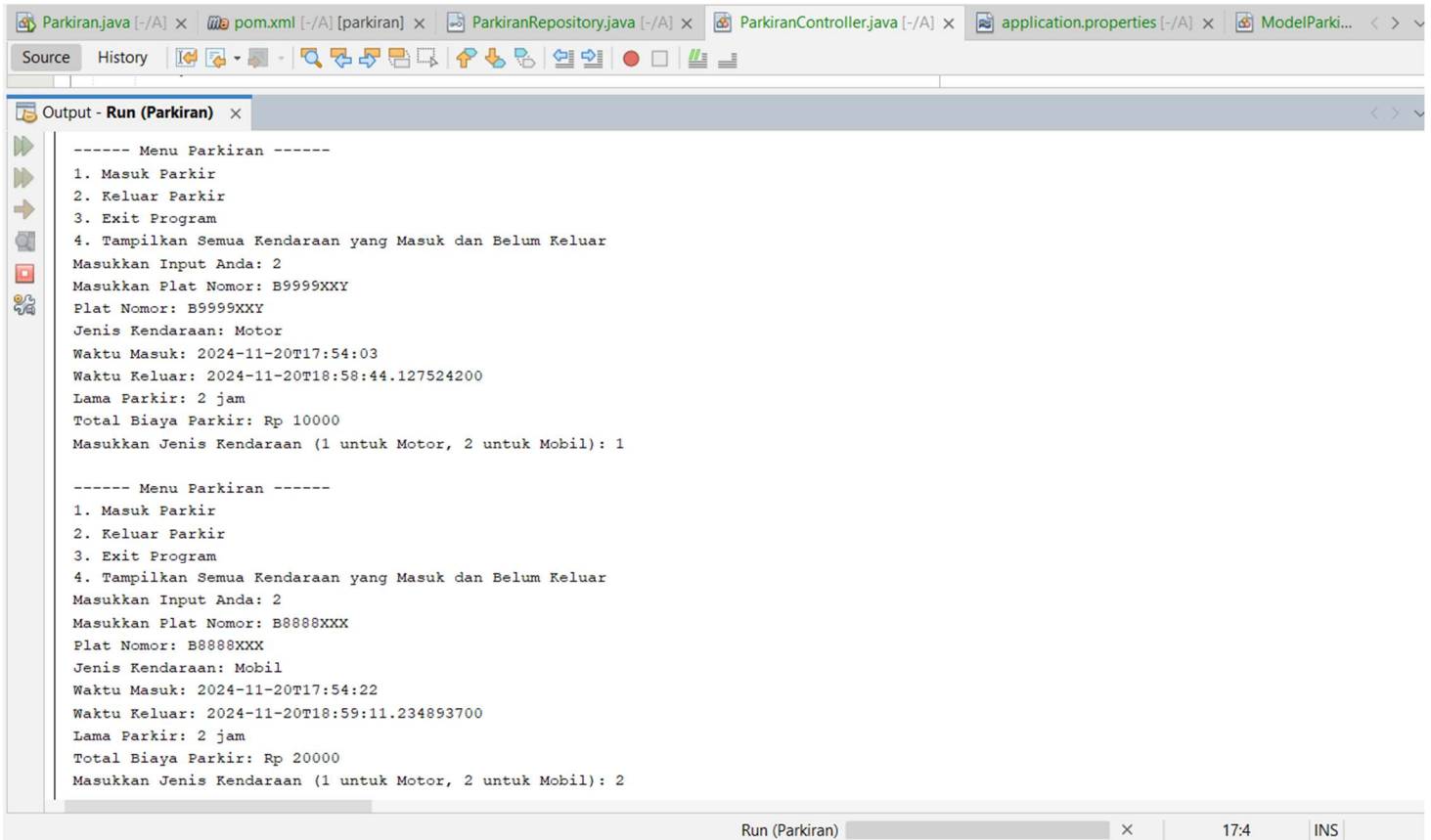
The screenshot shows the phpMyAdmin interface for a database named `spring_db`. The `parkiran` table is selected. The interface shows the following actions:

- Update:** A message "1 row affected." is displayed. The SQL query executed is: `UPDATE `parkiran` SET `waktu_masuk` = '2024-11-20 10:54:22.000000' WHERE `parkiran`.`id` = 5;`
- Query:** A message "Showing rows 0 - 1 (2 total, Query took 0.0002 seconds.)" is displayed. The SQL query executed is: `SELECT * FROM `parkiran``
- Table Data:** The table data is displayed in a grid. The columns are `id`, `jenis_kendaraan`, `plat_nomor`, and `waktu_masuk`. The data is as follows:

	id	jenis_kendaraan	plat_nomor	waktu_masuk
<input type="checkbox"/>	4	1	B9999XXY	2024-11-20 10:54:03.000000
<input type="checkbox"/>	5	2	B8888XXX	2024-11-20 10:54:22.000000

Below the table, there are options to "Check all", "With selected:", "Edit", "Copy", "Delete", and "Export". At the bottom, there are options to "Print", "Copy to clipboard", "Export", "Display chart", and "Create view".

OUTPUT kendaraan keluar parkir



```
----- Menu Parkiran -----
1. Masuk Parkir
2. Keluar Parkir
3. Exit Program
4. Tampilkan Semua Kendaraan yang Masuk dan Belum Keluar
Masukkan Input Anda: 2
Masukkan Plat Nomor: B9999XXY
Plat Nomor: B9999XXY
Jenis Kendaraan: Motor
Waktu Masuk: 2024-11-20T17:54:03
Waktu Keluar: 2024-11-20T18:58:44.127524200
Lama Parkir: 2 jam
Total Biaya Parkir: Rp 10000
Masukkan Jenis Kendaraan (1 untuk Motor, 2 untuk Mobil): 1

----- Menu Parkiran -----
1. Masuk Parkir
2. Keluar Parkir
3. Exit Program
4. Tampilkan Semua Kendaraan yang Masuk dan Belum Keluar
Masukkan Input Anda: 2
Masukkan Plat Nomor: B8888XXX
Plat Nomor: B8888XXX
Jenis Kendaraan: Mobil
Waktu Masuk: 2024-11-20T17:54:22
Waktu Keluar: 2024-11-20T18:59:11.234893700
Lama Parkir: 2 jam
Total Biaya Parkir: Rp 20000
Masukkan Jenis Kendaraan (1 untuk Motor, 2 untuk Mobil): 2
```

Run (Parkiran) × 17:4 INS