

LAPORAN PROJECT UAS
“SISTEM PARKIR MENGGUNAKAN GUI JAVA”

Diajukan untuk Memenuhi UAS Mata Kuliah Pemrograman Berorientasi Objek

Dosen Pengampu :

Taufik Ridwan, M.T.



Disusun oleh :

1. ADITYA TAZKIA AULIA M : 2310631250003
2. FREIDRICK ALBERT P : 2310631250058
3. MUHAMMAD RAKHA S : 2310631250024
4. RIZKY AZHARI PUTRA : 2310631250028

PROGRAM STUDI SISTEM INFORMASI
FAKULTAS ILMU KOMPUTER
UNIVERSITAS SINGAPERBANGSA KARAWANG
2025

KATA PENGANTAR

Puji syukur kami panjatkan ke hadirat Tuhan Yang Maha Esa atas limpahan rahmat dan karunia-Nya, sehingga kami dapat menyelesaikan tugas akhir semester ini dalam mata kuliah *Pemrograman Berorientasi Objek* dengan judul laporan "*Sistem Parkir Menggunakan GUI Java*". Laporan ini merupakan hasil dari kerja sama dan dedikasi kelompok kami dalam memenuhi salah satu tugas akademik pada semester IV di Program Studi Sistem Informasi, Fakultas Ilmu Komputer, Universitas Singaperbangsa Karawang.

Penyusunan laporan ini tidak terlepas dari bimbingan dan dukungan dari berbagai pihak. Untuk itu, kami mengucapkan terima kasih yang sebesar-besarnya kepada Bapak Taufik Ridwan, M.T., selaku dosen pengampu mata kuliah, yang telah memberikan arahan, pengetahuan, dan motivasi selama proses pembelajaran hingga tersusunnya laporan ini.

Laporan ini membahas tentang pengembangan sistem parkir berbasis Java GUI yang menerapkan konsep *Object-Oriented Programming* (OOP) seperti *encapsulation*, *inheritance*, dan *polymorphism*. Dalam proses pengembangannya, kami juga menggunakan pendekatan *Unified Modeling Language* (UML) untuk mendesain sistem secara terstruktur, serta menerapkan fitur-fitur penting seperti login multi-role, pencatatan kendaraan masuk dan keluar, perhitungan tarif otomatis, pengelolaan data parkir, hingga pencetakan struk. Semua data tersimpan dalam database MySQL yang terhubung dengan aplikasi melalui koneksi terpusat.

Kami menyadari bahwa laporan ini masih jauh dari sempurna karena keterbatasan pengalaman dan pengetahuan yang kami miliki. Oleh karena itu, kami sangat mengharapkan saran dan kritik yang membangun demi perbaikan dan pengembangan lebih lanjut. Harapan kami, laporan ini dapat memberikan manfaat bagi siapa saja yang membacanya dan menjadi referensi dalam pengembangan sistem parkir digital yang lebih baik.

Karawang, 9 Juni 2025

Penyusun

DAFTAR ISI

KATA PENGANTAR.....	ii
DAFTAR ISI	iii
BAB I : PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah.....	1
1.3 Tujuan	2
BAB II : PEMBAHASAN.....	3
2.1 Fitur – Fitur Aplikasi	3
2.2 Konsep OOP Pada Kode Program.....	4
2.3 Perancangan Menggunakan Unified Model Language (UML).....	10
2.4 Implementasi Kode Program.....	25
2.5 Pengujian.....	67
BAB III : PENUTUP	72
3.1 Kesimpulan	72
3.2 Saran	72

BAB I

PENDAHULUAN

1.1 Latar Belakang

Seiring dengan meningkatnya jumlah kendaraan bermotor setiap tahunnya, kebutuhan akan sistem parkir yang efisien dan terorganisir menjadi semakin penting, terutama di area perkantoran, kampus, pusat perbelanjaan, dan fasilitas umum lainnya. Sistem parkir konvensional yang masih menggunakan pencatatan manual seringkali menimbulkan berbagai permasalahan, seperti kesalahan pencatatan, keterlambatan dalam proses parkir, hingga kesulitan dalam melakukan pencarian data kendaraan.

Untuk mengatasi permasalahan tersebut, dibutuhkan sebuah sistem parkir yang dapat mengelola data kendaraan secara digital dan real-time. Penerapan Graphical User Interface (GUI) dalam sistem parkir dapat memberikan kemudahan dalam pengoperasian, baik bagi petugas parkir maupun pengguna. GUI memungkinkan pengguna berinteraksi dengan sistem melalui tampilan visual yang intuitif, sehingga proses input, update, dan pencarian data kendaraan dapat dilakukan dengan lebih cepat dan akurat.

Selain itu, sistem parkir berbasis GUI juga dapat diintegrasikan dengan konsep Object-Oriented Programming (OOP) seperti encapsulation, inheritance, dan polymorphism untuk meningkatkan modularitas dan fleksibilitas dalam pengembangan sistem. Dengan adanya fitur CRUD (Create, Read, Update, Delete), sistem ini memungkinkan pengelolaan data parkir yang terstruktur dan efisien.

Oleh karena itu, pengembangan sistem parkir dengan antarmuka GUI menjadi solusi yang tepat untuk meningkatkan efisiensi operasional dan pelayanan dalam manajemen parkir di berbagai lingkungan.

1.2 Rumusan Masalah

Berdasarkan latar belakang di atas, maka rumusan masalah dalam laporan ini dapat dirumuskan sebagai berikut:

1. Apa saja fitur-fitur yang dibutuhkan dalam sistem parkir untuk memudahkan proses pencatatan dan pengelolaan data kendaraan?

2. Bagaimana cara mengimplementasikan konsep Object-Oriented Programming (OOP) seperti encapsulation, inheritance, dan polymorphism dalam sistem parkir?
3. Bagaimana perancangan alur kerja dan kebutuhan sistem parkir berbasis GUI secara sistematis menggunakan pendekatan Unified Modeling Language (UML)?
4. Bagaimana mengimplementasikan rancangan UML ke dalam kode program menggunakan bahasa pemrograman Java dan konsep GUI?
5. Bagaimana melakukan pengujian terhadap fitur-fitur sistem parkir untuk memastikan bahwa sistem berjalan sesuai dengan kebutuhan?

1.3 Tujuan

Tujuan dari penulisan laporan ini adalah:

1. Mengidentifikasi dan merancang fitur-fitur utama yang dibutuhkan dalam sistem parkir untuk memudahkan proses pencatatan kendaraan masuk dan keluar serta pengelolaan data secara efisien.
2. Menerapkan konsep Object-Oriented Programming (OOP) seperti encapsulation, inheritance, dan polymorphism dalam membangun struktur kode program sistem parkir agar lebih modular, fleksibel, dan mudah dikembangkan.
3. Merancang alur kerja sistem parkir secara sistematis menggunakan pendekatan Unified Modeling Language (UML), seperti Use Case Diagram, Class Diagram, Activity Diagram, dan Sequence Diagram, untuk memvisualisasikan kebutuhan dan proses sistem.
4. Mengimplementasikan hasil perancangan UML ke dalam kode program menggunakan bahasa pemrograman Java dengan antarmuka berbasis Graphical User Interface (GUI) yang interaktif dan mudah digunakan.
5. Melakukan pengujian sistem parkir guna memastikan bahwa semua fitur yang dikembangkan berjalan sesuai dengan fungsinya, serta sistem dapat digunakan secara andal oleh admin maupun operator.

BAB II

PEMBAHASAN

2.1 Fitur – Fitur Aplikasi

1. Login Sistem
 - Pengguna harus login terlebih dahulu menggunakan username dan password.
 - Mendukung multi-role: admin dan operator.
 - Validasi login dilakukan ke database.
2. Dashboard Utama
 - Setelah login, pengguna diarahkan ke dashboard sesuai role.
 - Terdapat menu navigasi dan informasi pengguna yang sedang login.
3. Pencatatan Kendaraan Masuk
 - Form input untuk plat nomor dan jenis kendaraan (Motor/Mobil).
 - Waktu masuk dicatat otomatis (timestamp).
 - Data kendaraan masuk disimpan ke database dan ditampilkan di tabel.
4. Pencatatan Kendaraan Keluar
 - Form input untuk plat nomor kendaraan yang akan keluar.
 - Sistem mencari data kendaraan yang belum keluar.
 - Waktu keluar dicatat otomatis.
 - Sistem menghitung durasi parkir dan tarif parkir secara otomatis berdasarkan jenis kendaraan dan lama parkir.
 - Data keluar diupdate ke database dan tabel.
5. Tabel Data Parkir
 - Menampilkan seluruh data kendaraan yang masuk dan keluar.
 - Kolom: Plat Nomor, Jenis, Tanggal Masuk, Tanggal Keluar, Durasi Parkir, Tarif.
 - Data dapat diurutkan (A-Z/Z-A) dan dicari berdasarkan plat nomor.
6. Edit & Hapus Data Parkir (khusus admin)
 - Admin dapat mengedit data parkir (plat, jenis, tanggal, tarif, dll) melalui dialog edit.
 - Admin dapat menghapus data parkir.

7. Cetak Struk Parkir
 - Fitur untuk mencetak struk parkir kendaraan yang keluar.
 - Struk berisi detail kendaraan, waktu masuk, keluar, durasi, dan tarif.
8. Logout
 - Pengguna dapat logout dari aplikasi dengan aman.
9. Tampilan Modern & Responsive
 - GUI menggunakan Java Swing dengan desain modern (warna, font, panel rounded, dsb).
 - Terdapat panel jam digital yang selalu update.
10. Koneksi Database
 - Semua data parkir, user, dan transaksi tersimpan di database MySQL.
 - Koneksi database dikelola melalui kelas khusus (DBConnection).

2.2 Konsep OOP Pada Kode Program

Object-Oriented Programming (OOP) adalah metode pemrograman yang mendasarkan pengembangan perangkat lunak pada objek-objek yang saling berinteraksi. Dalam paradigma ini, kode program dibangun dengan mengelompokkan data dan fungsi ke dalam satu kesatuan bernama kelas (class). Class berfungsi sebagai cetakan, sedangkan objek adalah perwujudan nyata dari cetakan tersebut yang bisa digunakan dalam program.

OOP memberikan pendekatan modular dan terstruktur, memudahkan dalam pengelolaan program skala besar, serta mendukung fitur-fitur seperti enkapsulasi, pewarisan, dan polymorphism.

- **Class**

Class adalah struktur dasar dalam OOP yang berfungsi sebagai template atau blueprint untuk menciptakan object. Di dalam kelas, kita mendefinisikan atribut (data) dan method (perilaku) yang nantinya akan dimiliki oleh object.

```
public class ParkirAppSystem extends JFrame {  
    // ...  
}
```

Class ParkirAppSystem adalah class turunan dari JFrame dalam Java. Dalam OOP, ini menunjukkan konsep inheritance (pewarisan), di mana ParkirAppSystem mewarisi semua fitur dari JFrame. Class ini akan digunakan untuk membangun aplikasi parkir dengan antarmuka pengguna.

- **Object**

```
tfPlatMasuk = new JTextField();
cbJenisMasuk = new JComboBox<>(new String[]{"Motor", "Mobil"});
lblJamMasuk = new JLabel("Auto-generated");
table = new JTable(tableModel);
```

Pada potongan kode tersebut, terdapat beberapa object yang dibuat dari class bawaan Java Swing, yaitu JTextField, JComboBox, JLabel, dan JTable. Dalam OOP, object merupakan instance dari class yang memiliki atribut dan method untuk merepresentasikan perilaku dan data. Misalnya, tfPlatMasuk adalah object dari JTextField yang digunakan untuk input teks, cbJenisMasuk adalah object dari JComboBox untuk memilih jenis kendaraan, lblJamMasuk adalah label untuk menampilkan informasi, dan table adalah object dari JTable untuk menampilkan data dalam bentuk tabel. Pembuatan object dengan keyword new ini mencerminkan penerapan prinsip instansiasi dalam OOP, di mana setiap object memiliki peran dan fungsi sesuai class-nya dalam membentuk perilaku program.

- **Atribut**

Atribut adalah variabel yang dimiliki oleh sebuah objek (atau kelas) yang menyimpan data atau keadaan (state) dari objek tersebut. Atribut biasanya dideklarasikan di dalam kelas dan mewakili properti dari objek. Contoh: String, Int, Timer.

```
private JTextField tfPlatMasuk, tfPlatKeluar, tfSearch;
private JComboBox<String> cbJenisMasuk, cbSort;
private JLabel lblJamMasuk, lblJamKeluar, lbdurasi_menit, lblTarif, lblClock;
private Timer timer;
private JTable table;
private DefaultTableModel tableModel;
private String currentRole;
```

Atribut-atribut ini mencerminkan komponen GUI dan data yang digunakan oleh objek dalam program berbasis Java Swing.

- **Method**

Method adalah bagian penting yang merepresentasikan perilaku (behavior) dari sebuah objek. Method memungkinkan objek untuk melakukan suatu aksi atau memproses data internalnya.

```
private void startClock() { ... }
private void handleKendaraanMasuk() { ... }
private void handleKendaraanKeluar() { ... }
private void updateDatabase(String plat, String jamMasuk, String jamKeluar, long durasi, long tarif)
private Kendaraan createKendaraan(String jenis, String plat, String jamMasuk) { ... }
private long calculateDuration(String masuk, String keluar) { ... }
private void loadDataFromDatabase() { ... }
private void performSearch() { ... }
private void resetSearch() { ... }
private void searchByPlat(String plat, int sortIndex) { ... }
private void deleteSelectedRow() { ... }
private void showEditDialog() { ... }
private void showPrintDialog() { ... }
private void logout() { ... }
private void styleButton(AbstractButton button, Color bgColor) { ... }
private JLabel createLabel(String text, Font font) { ... }
private void showMessage(String message, String title, int messageType) { ... }
```

Setiap method memiliki nama yang merepresentasikan tugas spesifiknya. Misalnya, handleKendaraanMasuk() dan handleKendaraanKeluar() digunakan untuk menangani proses saat kendaraan masuk dan keluar. Method seperti calculateDuration(...) digunakan untuk menghitung lama parkir berdasarkan waktu masuk dan keluar, sedangkan updateDatabase(...) bertugas memperbarui data di penyimpanan. Ada juga method yang berfungsi untuk pencarian (performSearch(), searchByPlat(...)) dan pengelolaan antarmuka pengguna seperti showEditDialog() atau showMessage(...).

- **Enkapsulasi**

Enkapsulasi merupakan teknik dalam OOP untuk melindungi data internal objek agar tidak diakses secara langsung dari luar kelas. Data hanya bisa dimodifikasi melalui method khusus seperti getter dan setter.

```
public abstract class Kendaraan {  
    protected String plat;  
    protected String jamMasuk;
```

Pada kode tersebut, atribut plat dan jamMasuk menggunakan access modifier protected. Penggunaan protected menunjukkan bahwa atribut-atribut tersebut tidak dapat diakses secara langsung dari luar kelas, melainkan hanya dapat diakses dari dalam kelas itu sendiri, kelas turunannya (subclass), atau kelas lain yang berada dalam package yang sama.

```
// Getter methods  
public String getPlat() {  
    return plat;  
}  
  
public String getJamMasuk() {  
    return jamMasuk;  
}
```

Method getter dalam kode ini digunakan untuk mengambil nilai dari atribut yang bersifat protected. Contohnya, method getPlat() digunakan untuk mengakses nilai dari atribut plat, dan getJamMasuk() digunakan untuk mendapatkan nilai dari atribut jamMasuk. Kedua method ini bersifat public, sehingga memungkinkan bagian lain dari program (di luar kelas) untuk membaca nilai atribut tanpa mengaksesnya secara langsung.

```
// Setter methods  
public void setPlat(String plat) {  
    this.plat = plat;  
}  
  
public void setJamMasuk(String jamMasuk) {  
    this.jamMasuk = jamMasuk;  
}
```

Method setter untuk memungkinkan perubahan nilai atribut secara aman. Setter berfungsi untuk memberikan akses tidak langsung dalam mengubah nilai atribut yang bersifat protected, seperti plat dan jamMasuk. Method setPlat(String plat) dan setJamMasuk(String jamMasuk) digunakan

untuk mengatur ulang nilai dari atribut plat dan jamMasuk. Sama seperti getter, method setter ini bersifat public, sehingga bisa diakses dari luar kelas.

● Inheritance

```
public class Mobil extends Kendaraan {  
    public Mobil(String plat, String jamMasuk) {  
        super(plat, jamMasuk);  
    }  
}
```

```
public class Motor extends Kendaraan {  
    public Motor(String plat, String jamMasuk) {  
        super(plat, jamMasuk);  
    }  
}
```

Kode di atas merupakan contoh penerapan konsep inheritance (pewarisan) dalam pemrograman berorientasi objek (OOP). Inheritance memungkinkan sebuah kelas (subclass) mewarisi properti dan perilaku dari kelas lain (superclass). Dalam kode ini, kelas Kendaraan berperan sebagai superclass (kelas induk) yang mendefinisikan atribut umum

Kelas Mobil dan Motor masing-masing berperan sebagai subclass (kelas turunan) dari Kendaraan. Melalui keyword extends, kedua kelas ini mewarisi seluruh atribut dan method dari superclass-nya, sehingga tidak perlu menuliskan ulang atribut atau fungsi dasar yang sama.

● Polymorphism

```
package Uas_Pbo;  
  
public class Motor extends Kendaraan {  
    private static final long TARIF_PER_JAM = 2000;  
  
    public Motor(String plat, String jamMasuk) {  
        super(plat, jamMasuk);  
    }  
  
    @Override  
    public long hitungTarif(long durasiMenit, long jumlahHari) {  
        // Implementasi tarif motor  
        // ...  
        return 0;  
    }  
  
    // Method overloading (polymorphism)  
    public long hitungTarif(long durasiMenit) {  
        // Tarif tanpa memperhitungkan hari  
        return (durasiMenit / 60) * TARIF_PER_JAM;  
    }  
  
    public String getJenis() {  
        return "Motor";  
    }  
}
```

Dalam paradigma Object-Oriented Programming (OOP), konsep polymorphism memungkinkan suatu method memiliki banyak bentuk atau perilaku yang berbeda tergantung pada konteksnya. Dua bentuk utama dari polymorphism yang terlihat pada kode Anda adalah method overloading dan method overriding.

1. Polymorphism dengan Overloading

```
// Method overloading (polymorphism)
public long hitungTarif(long durasiMenit) {
    // Tarif tanpa memperhitungkan hari
    return (durasiMenit / 60) * TARIF_PER_JAM;
}
```

Overloading adalah polymorphism statis (compile-time), di mana dua atau lebih method memiliki nama yang sama tetapi parameter berbeda (jumlah atau tipe) dalam satu class. Contohnya pada class Motor, terdapat dua method hitungTarif dengan parameter yang berbeda. Ini memungkinkan method dipanggil sesuai kebutuhan tanpa membuat nama method baru.

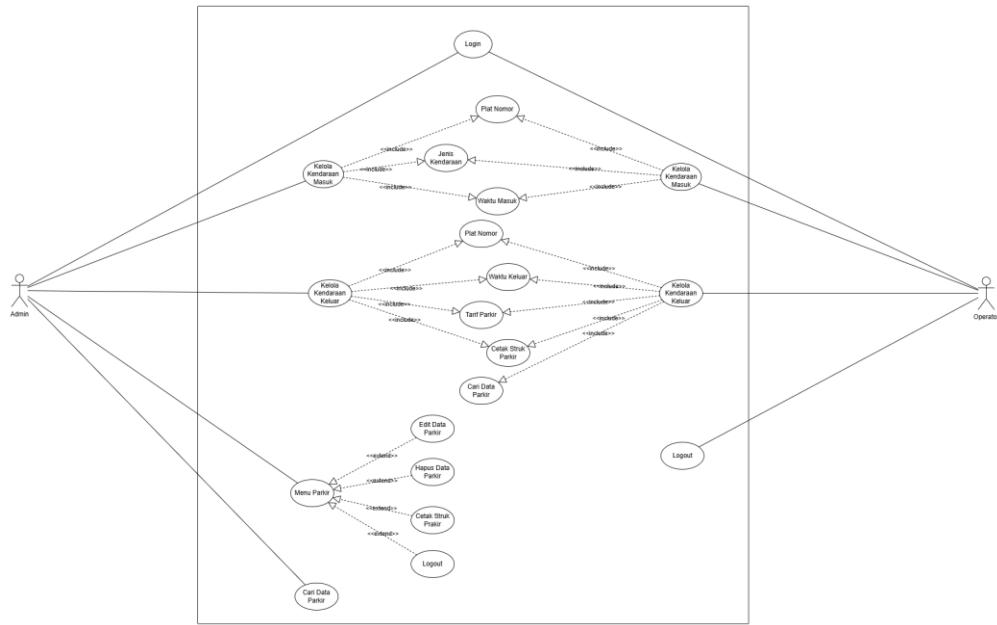
2. Polymorphism dengan Overriding

```
@Override
public long hitungTarif(long durasiMenit, long jumlahHari) {
    // Implementasi tarif motor
    // ...
    return 0;
}
```

Overriding adalah polymorphism dinamis (runtime), di mana subclass memberikan implementasi ulang terhadap method yang sudah ada di superclass dengan nama dan parameter yang sama. Dalam kode di atas, class Motor meng-*override* method hitungTarif dari class Kendaraan, sehingga saat objek bertipe Kendaraan sebenarnya adalah Motor, method Motor yang dijalankan.

2.3 Perancangan Menggunakan Unified Model Language (UML)

1. Use Case Diagram



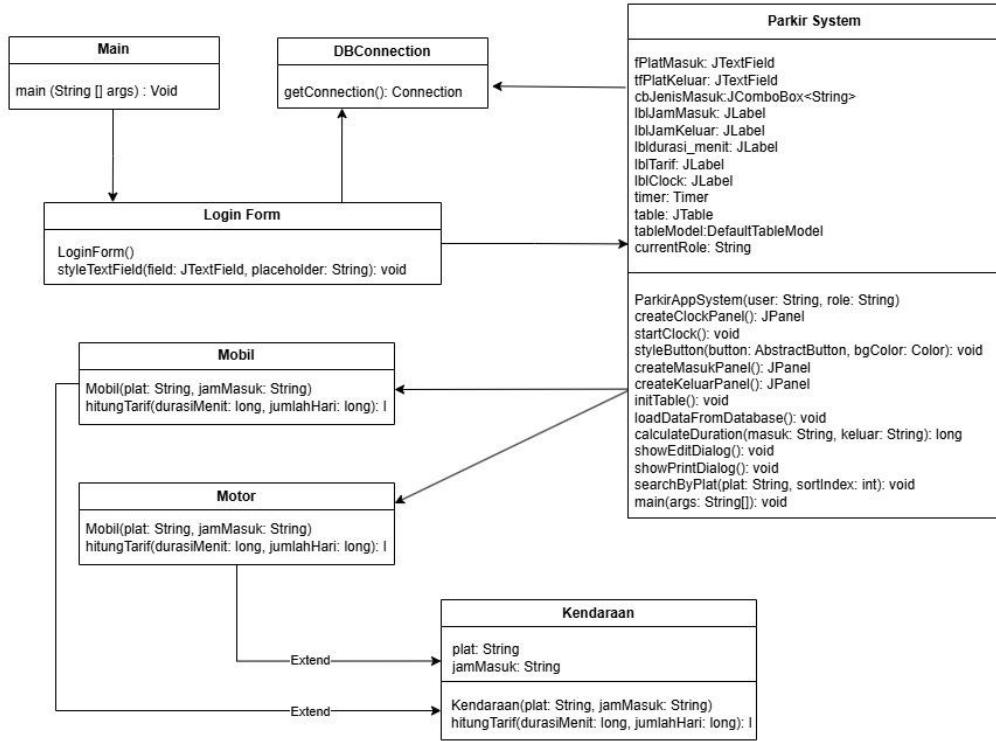
Dalam sistem ini terdapat dua aktor utama, yaitu Admin dan Operator, yang masing-masing memiliki peran dalam pengelolaan data kendaraan. Sebelum mengakses sistem, kedua aktor harus melakukan Login sebagai proses autentikasi awal.

Setelah login, Admin dan Operator dapat melakukan proses Kelola Kendaraan Masuk, yang mencakup pencatatan data kendaraan seperti plat nomor, jenis kendaraan, dan jam masuk. Sedangkan pada proses Kelola Kendaraan Keluar, sistem akan mencatat kembali plat nomor dan jam keluar, lalu menghitung tarif parkir berdasarkan lama waktu kendaraan berada di area parkir. Setelah itu, Operator dapat langsung melakukan Cetak Struk sebagai bukti transaksi kepada pengguna layanan parkir.

Sistem ini juga menyediakan fitur tambahan yang dapat diakses melalui Menu Parkir, seperti Edit Data Parkir, Hapus Data Parkir, Cetak Struk, Logout, serta Cari Data Parkir. Semua fitur tersebut ditampilkan sebagai bagian dari menu utama untuk memudahkan pengguna dalam mengelola data kendaraan. Dalam diagram ini, beberapa use case saling terhubung melalui relasi *include*,

yang menunjukkan bahwa sebuah aktivitas utama terdiri dari beberapa langkah yang wajib dilakukan.

2. Class Diagram



Kelas Main berfungsi sebagai titik awal program dengan metode `main(String[] args): Void`. Kelas ini akan menjalankan kelas Login Form, yang memiliki konstruktor `LoginForm()` serta metode `styleTextField(JTextField, placeholder: String): void` untuk mengatur tampilan field login.

Kelas DBConnection bertanggung jawab dalam koneksi ke basis data, ditunjukkan oleh metode `getConnection(): Connection`.

Setelah proses login, sistem akan diarahkan ke kelas Parkir System, yang memiliki berbagai atribut antarmuka pengguna seperti `tfPlatMasuk`, `tfPlatKeluar`, `cbJenisMasuk`, serta label dan komponen UI lainnya. Kelas ini memiliki banyak metode seperti `createClockPanel()`, `createMaskPanel()`, `loadDataFromDatabase()`, `search()`, dan lainnya yang digunakan untuk menangani fungsi utama sistem parkir, termasuk mencetak tiket, mencari data kendaraan, dan menampilkan informasi.

Pada sisi kendaraan, terdapat kelas Kendaraan sebagai kelas induk yang memiliki atribut plat dan jamMasuk, serta metode hitungTarif(durasiMent: long, jumlahHari: long): long. Dua kelas yaitu Mobil dan Motor merupakan turunan dari kelas Kendaraan, dan masing-masing mengimplementasikan metode hitungTarif() sesuai jenis kendaraan.

Secara keseluruhan, diagram ini menggambarkan sistem parkir terintegrasi yang menangani login pengguna, pengelolaan data kendaraan masuk dan keluar, serta perhitungan tarif parkir berdasarkan durasi dan jenis kendaraan.

3. Activity Diagram

a. Activity Diagram Login

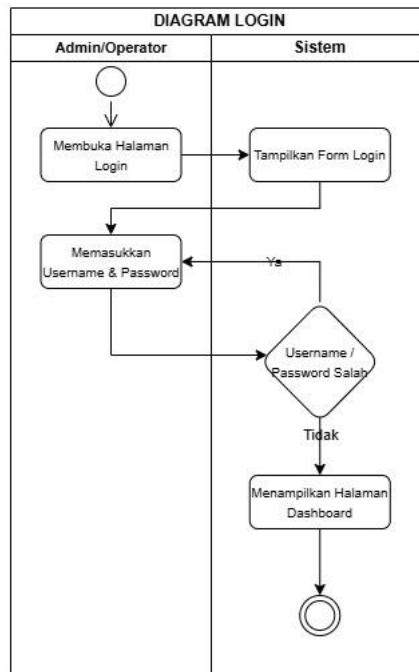


Diagram "Login" mengilustrasikan proses Admin/Operator masuk ke dalam sistem. Dimulai dengan Admin/Operator membuka halaman login dan memasukkan username serta password. Sistem akan menampilkan form login. Setelah username dan password dimasukkan, sistem melakukan validasi. Jika username atau password salah, Admin/Operator akan diminta kembali untuk memasukkan username dan password. Jika benar, sistem akan menampilkan halaman dashboard, dan proses login selesai

b. Activity Diagram Logout

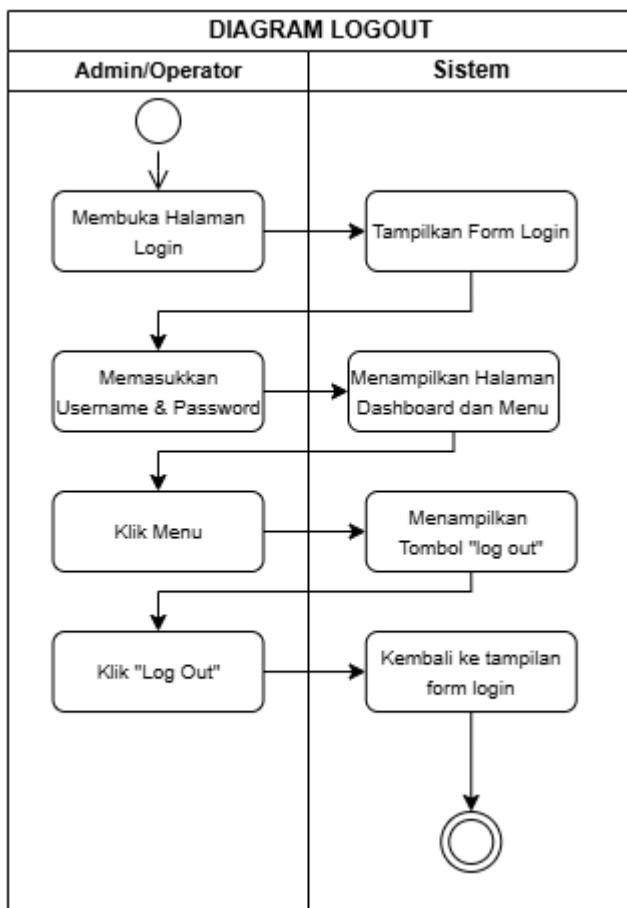


Diagram "Logout" menjelaskan bagaimana Admin/Operator keluar dari sistem. Proses dimulai ketika Admin/Operator membuka halaman login, kemudian memasukkan username dan password. Sistem akan menampilkan halaman dashboard dan menu. Admin/Operator selanjutnya mengklik menu dan memilih tombol "Log out". Sistem kemudian akan kembali ke tampilan form login, menandakan proses logout telah berhasil dan selesai.

c. Activity Diagram Edit

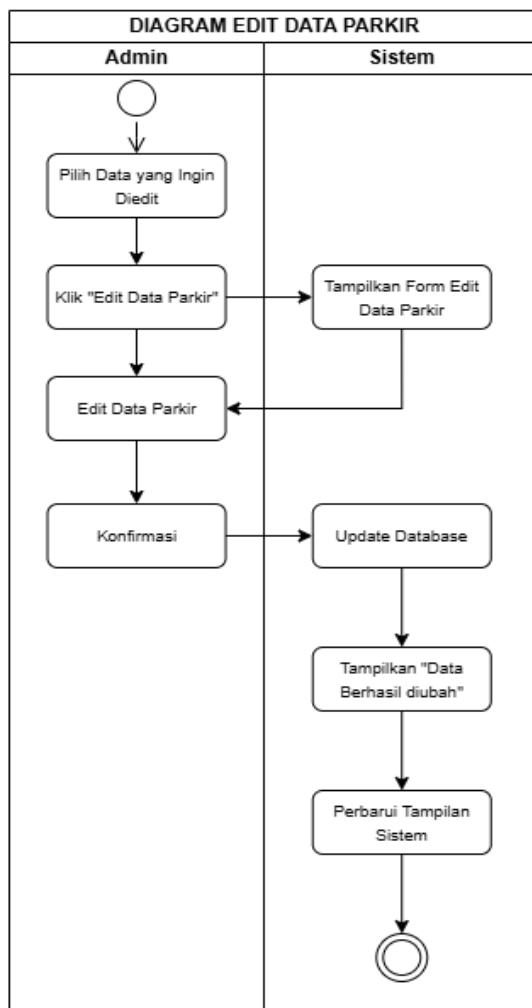


Diagram "Edit Data Parkir" menggambarkan alur kerja ketika Admin/Operator mengubah data parkir. Admin/Operator memilih data yang ingin diedit dan mengklik tombol "Edit Data Parkir". Sistem kemudian menampilkan form untuk mengedit data parkir. Admin/Operator melakukan pengeditan data dan memberikan konfirmasi. Setelah konfirmasi, sistem memperbarui data di database, menampilkan pesan bahwa data berhasil diubah, dan memperbarui tampilan sistem. Proses edit data parkir selesai.

d. Activity Diagram Hapus

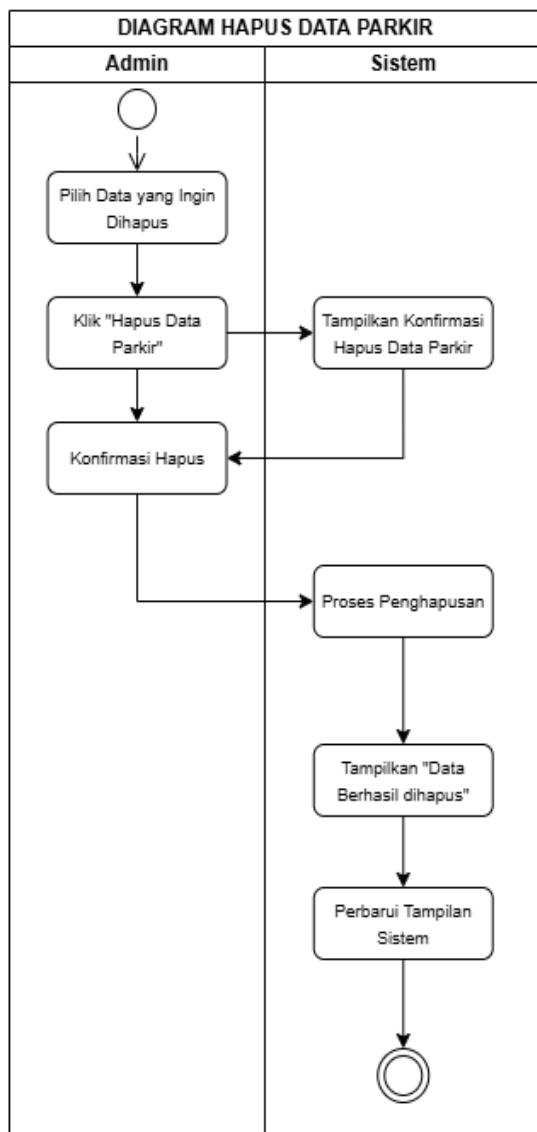


Diagram "Hapus Data Parkir" merinci langkah-langkah untuk menghapus data parkir. Admin/Operator memulai dengan memilih data yang ingin dihapus dan mengklik tombol "Hapus Data Parkir". Sistem akan menampilkan konfirmasi penghapusan data parkir. Admin/Operator kemudian memberikan konfirmasi untuk menghapus. Setelah konfirmasi diberikan, sistem memproses penghapusan data, menampilkan pesan bahwa data berhasil dihapus, dan memperbarui tampilan sistem. Proses pen

e. Activity Diagram Kelola Kendaraan Masuk

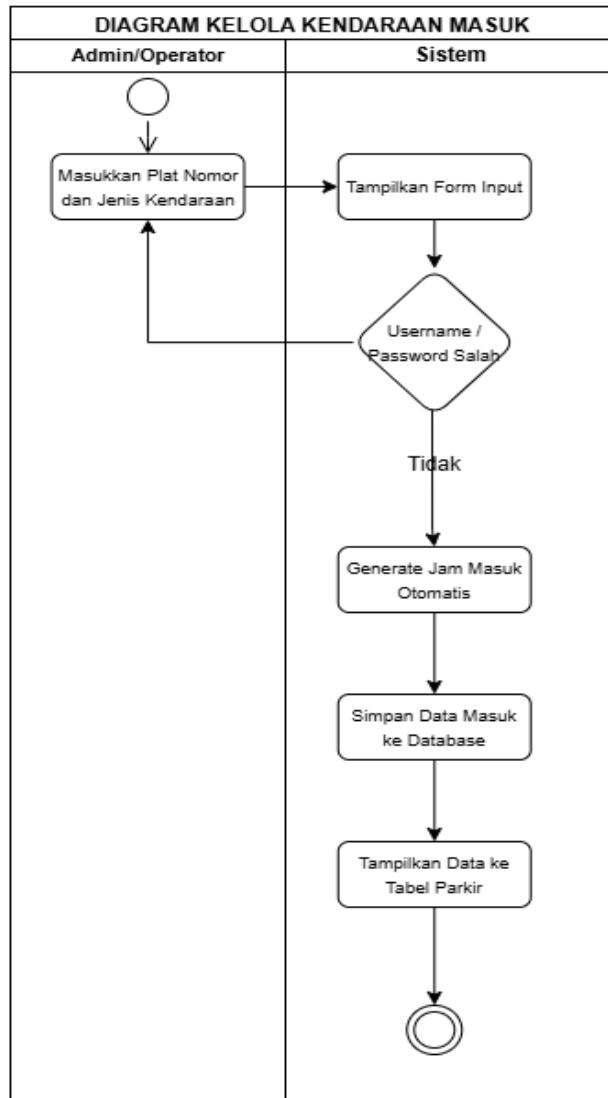


Diagram "Kelola Kendaraan Masuk" menunjukkan proses pengelolaan kendaraan yang masuk oleh Admin/Operator. Dimulai dengan Admin/Operator memasukkan plat nomor dan jenis kendaraan. Sistem kemudian menampilkan form input. Terdapat kondisi pengecekan username/password; jika salah, Admin/Operator diminta kembali memasukkan plat nomor dan jenis kendaraan. Jika benar (atau tidak ada validasi username/password di langkah ini, melainkan langsung ke proses selanjutnya), sistem akan menghasilkan jam masuk secara otomatis, menyimpan data masuk tersebut ke database, dan menampilkan data ke tabel parkir. Proses pengelolaan kendaraan masuk kemudian selesai.

f. Activity Diagram Kelola Kendaraan Keluar

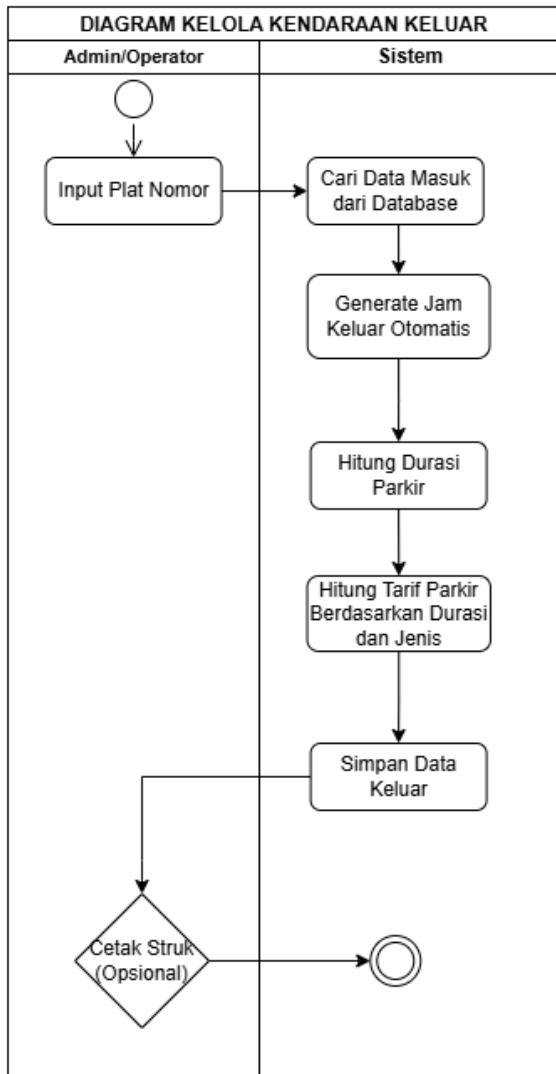


Diagram "Kelola Kendaraan Keluar" menjelaskan alur proses ketika Admin/Operator mengelola kendaraan yang keluar dari area parkir. Admin/Operator menginput nomor plat kendaraan. Sistem kemudian mencari data masuk kendaraan tersebut dari database, menghasilkan jam keluar secara otomatis, menghitung durasi parkir, dan menghitung tarif parkir berdasarkan durasi serta jenis kendaraan. Setelah itu, sistem menyimpan data keluar. Admin/Operator memiliki opsi untuk mencetak struk, dan setelah itu proses pengelolaan kendaraan keluar selesai.

g. Activity Diagram Cari Data Parkir

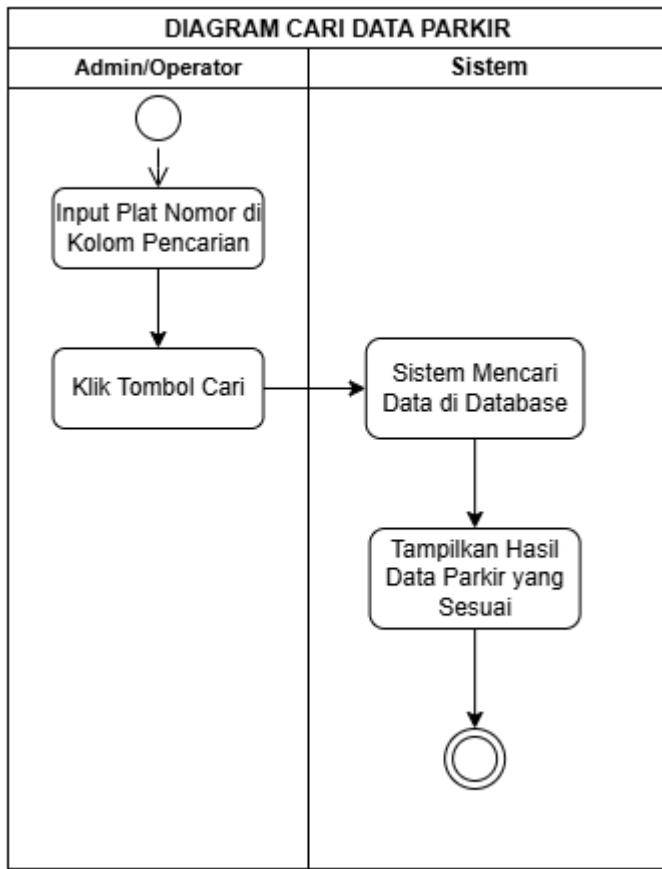


Diagram "Cari Data Parkir" menggambarkan proses pencarian data parkir oleh Admin/Operator. Proses dimulai ketika Admin/Operator menginput nomor plat kendaraan di kolom pencarian dan mengklik tombol "Cari". Sistem kemudian akan mencari data tersebut di dalam database. Setelah data ditemukan, sistem akan menampilkan hasil data parkir yang sesuai kepada Admin/Operator, dan proses berakhir.

h. Activity Diagram Cetak Struk

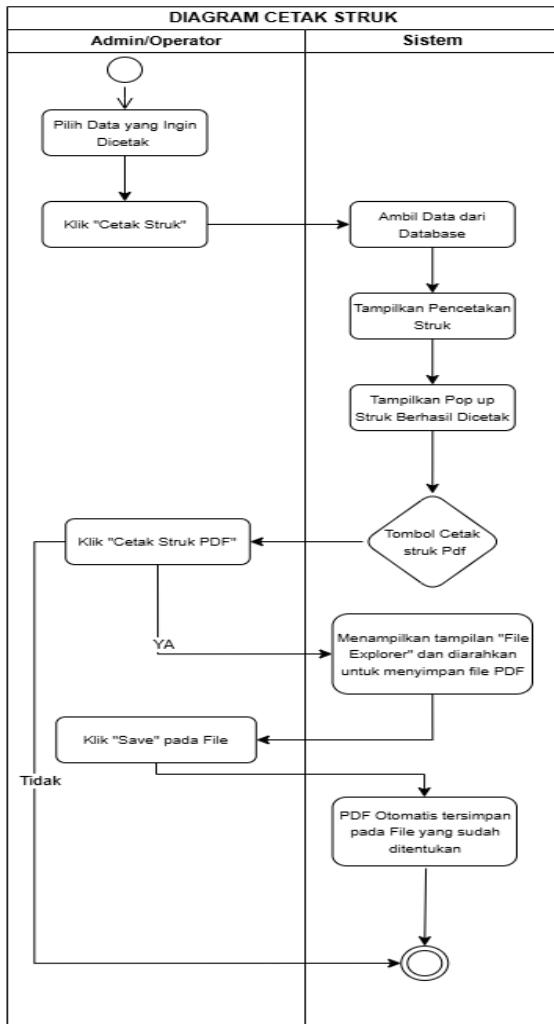


Diagram "Cetak Struk" ini mengilustrasikan proses pencetakan struk parkir oleh Admin/Operator dengan opsi penyimpanan sebagai PDF. Proses diawali saat Admin/Operator memilih data yang diinginkan dan mengklik tombol "Cetak Struk". Sistem kemudian mengambil data dari database, menampilkan pratinjau struk, serta memunculkan notifikasi pop-up bahwa struk berhasil dicetak. Setelah itu, alur berlanjut ke opsi pencetakan PDF. Jika Admin/Operator memilih untuk mengklik "Cetak Struk PDF", maka sistem akan menampilkan jendela "File Explorer" agar pengguna dapat memilih lokasi penyimpanan dan menyimpan file tersebut dengan mengklik "Save". Namun, jika opsi tersebut tidak dipilih (alur "Tidak"), sistem akan secara otomatis menyimpan file PDF ke lokasi yang sudah ditentukan sebelumnya, dan setelah itu proses pencetakan struk dianggap selesai.

4. Sequence Diagram

a. Sequence Diagram Login

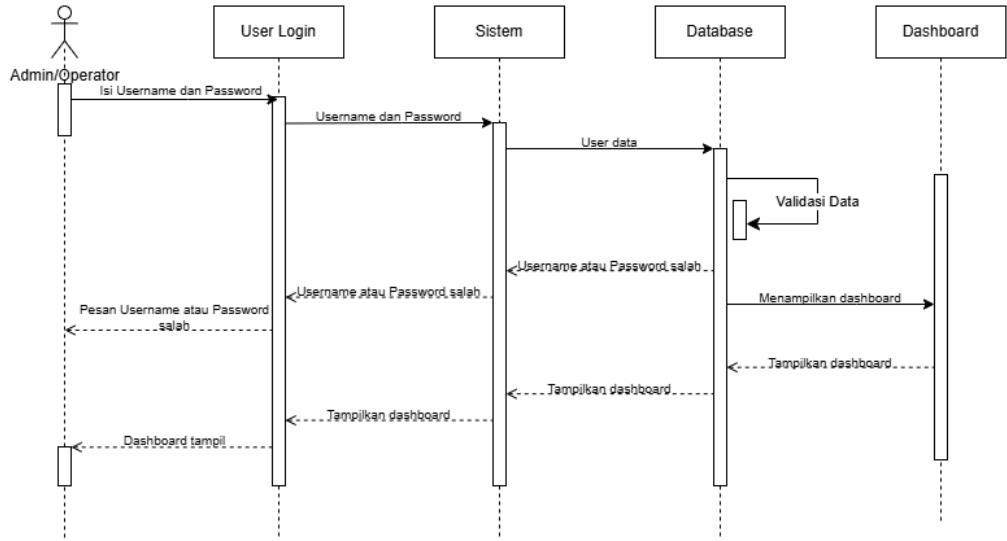


Diagram ini merinci interaksi saat login. Admin/Operator memasukkan username dan password pada antarmuka "User Login", yang kemudian dikirim ke Sistem. Sistem meneruskan data tersebut ke Database untuk divalidasi. Jika data tidak valid, menampilkan pesan error pada antarmuka "User Login" kepada Admin. Jika data valid, Database akan memberi sinyal untuk menampilkan "Dashboard", yang kemudian ditampilkan secara berurutan oleh Sistem dan "User Login" kepada Admin/Operator.

b. Sequence Diagram Logout

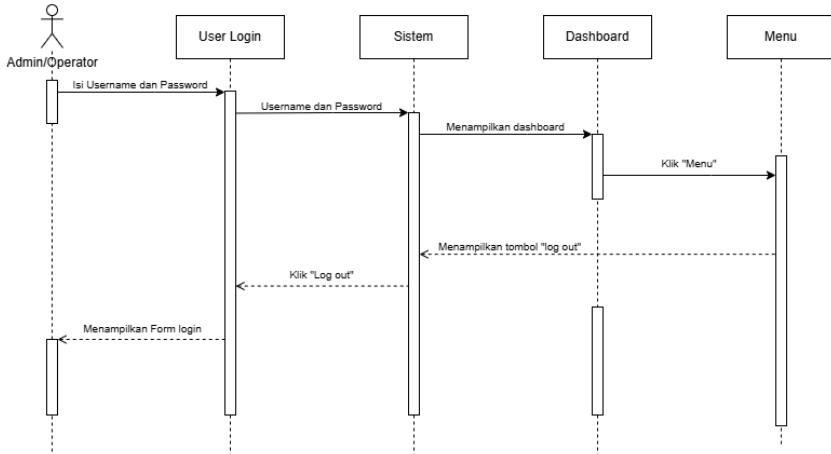


Diagram ini menunjukkan urutan proses logout. Admin/Operator memasukkan username dan password di halaman "User Login", yang diteruskan ke Sistem. Sistem kemudian menampilkan "Dashboard". Di Dashboard, Admin mengklik "Menu", yang memicu Sistem untuk menampilkan tombol "log out". Ketika Admin mengklik "Log out" di halaman login, Sistem merespons dengan menampilkan kembali "Form login", menandakan sesi telah berakhir.

c. Sequence Diagram Edit

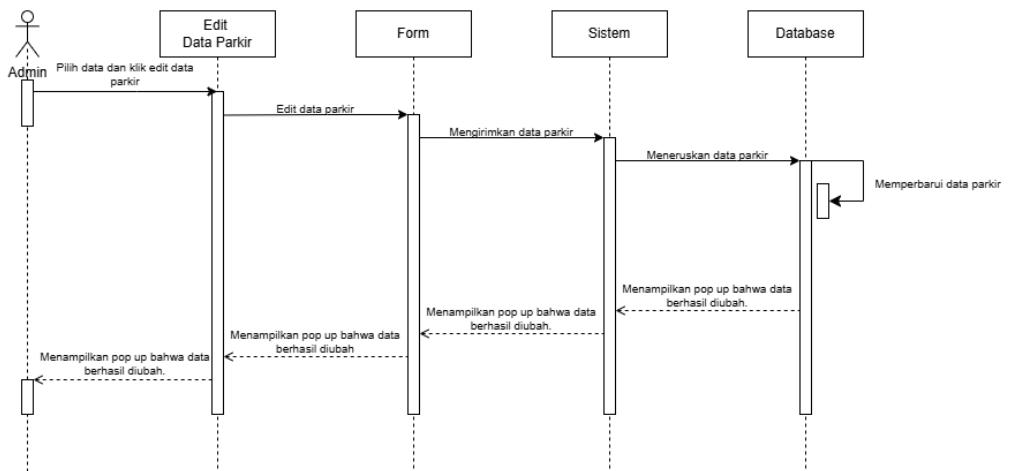


Diagram ini menunjukkan proses pengeditan data parkir. Admin memilih data dan mengklik tombol edit pada halaman "Edit Data Parkir". Perintah ini diteruskan ke Form, yang mengirimkan data parkir yang akan diubah ke Sistem. Sistem meneruskan data tersebut ke Database untuk diperbarui. Setelah pembaruan berhasil, Database mengirimkan konfirmasi ke Sistem, yang kemudian menampilkan pesan pop-up keberhasilan secara berantai melalui Form, halaman "Edit Data Parkir", hingga sampai ke Admin.

d. Sequence Diagram Hapus

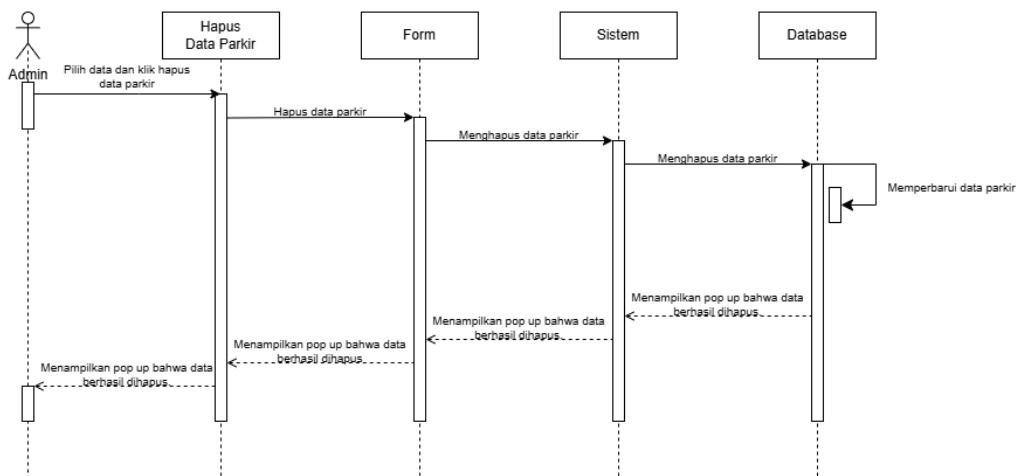


Diagram ini menggambarkan alur penghapusan data parkir. Admin memilih data dan mengklik tombol hapus pada halaman "Hapus Data Parkir". Perintah ini diteruskan melalui Form ke Sistem, yang kemudian memerintahkan Database untuk menghapus data tersebut. Database memperbarui datanya sendiri dan mengirimkan konfirmasi kembali ke Sistem. Sistem lalu secara berurutan menampilkan notifikasi pop-up "data berhasil dihapus" melalui Form, ke halaman "Hapus Data Parkir", dan akhirnya ditampilkan kepada Admin.

e. Sequence Diagram Cetak Struk

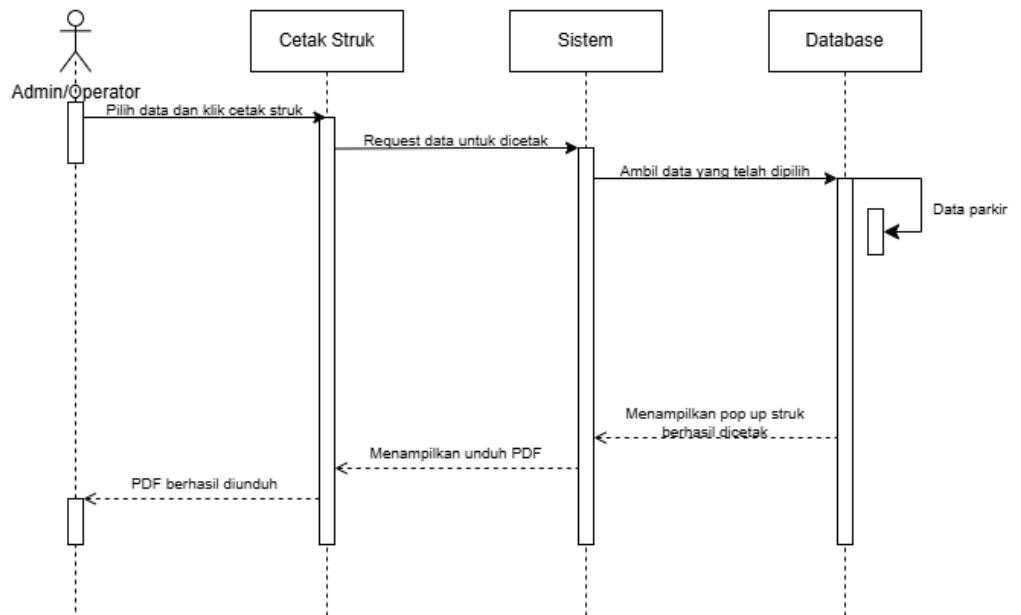


Diagram ini menjelaskan interaksi untuk mencetak struk. Admin/Operator memulai dengan memilih data dan mengklik cetak struk. Antarmuka "Cetak Struk" kemudian mengirimkan permintaan data ke Sistem. Sistem mengambil data yang telah dipilih dari Database. Setelah menerima data parkir, Sistem memberi perintah untuk menampilkan pop-up bahwa struk berhasil dicetak dan secara bersamaan mengirimkan perintah ke antarmuka "Cetak Struk" untuk menampilkan unduhan PDF. Akhirnya, Admin/Operator menerima notifikasi bahwa PDF berhasil diunduh.

f. Sequence Diagram Kelola Kendaraan Masuk

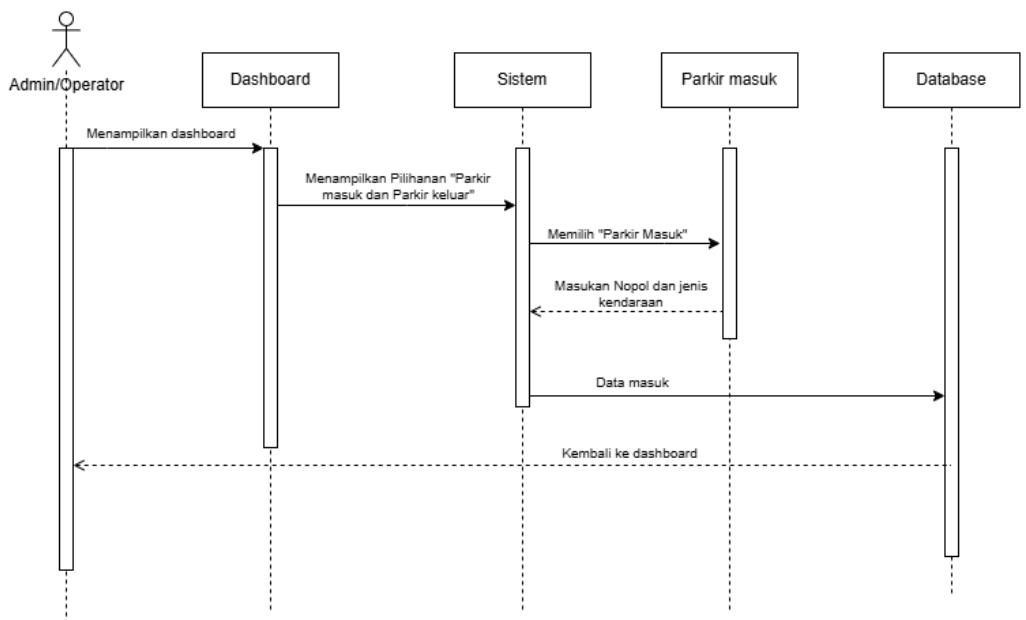


Diagram ini menjelaskan proses pencatatan kendaraan masuk. Dari Dashboard, Sistem menampilkan pilihan "Parkir masuk dan Parkir keluar". Admin/Operator memilih "Parkir Masuk". Sistem, melalui antarmuka "Parkir masuk", kemudian meminta Admin/Operator untuk memasukkan Nomor Polisi (Nopol) dan jenis kendaraan. Data masuk ini kemudian dikirim oleh Sistem untuk disimpan di dalam Database. Setelah proses selesai, alur kembali ke Dashboard.

g. Sequence Diagram Kelola Kendaraan Keluar

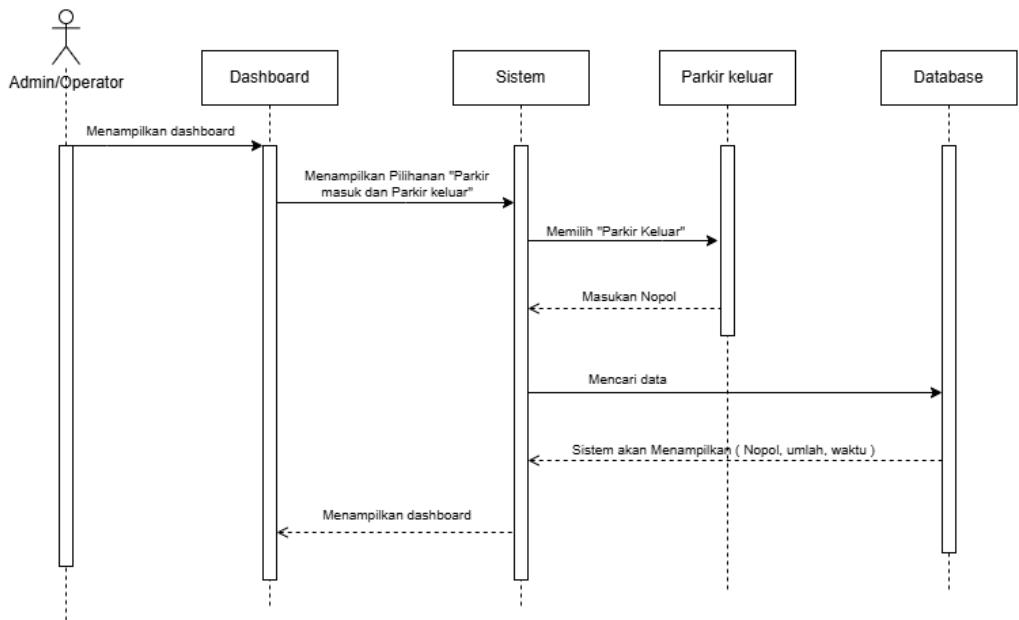


Diagram ini merinci proses kendaraan keluar. Admin/Operator yang berada di Dashboard diperlihatkan pilihan "Parkir masuk dan Parkir keluar" oleh Sistem. Ketika Admin memilih "Parkir Keluar", Sistem meminta masukan Nomor Polisi (Nopol) melalui antarmuka "Parkir keluar". Setelah Nopol dimasukkan, Sistem mencari data di Database. Database mengembalikan data (Nopol, jumlah, waktu) ke Sistem, yang kemudian akan menampilkannya kepada Admin/Operator dan kembali menampilkan Dashboard.

h. Sequence Diagram Cari Data Parkir

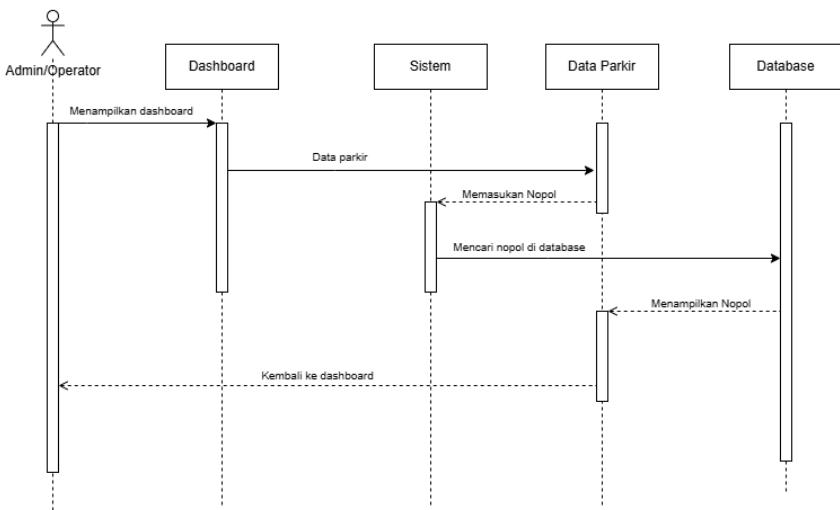


Diagram ini mengilustrasikan pencarian data parkir dari Dashboard. Admin/Operator mengakses fitur "Data parkir" dari Dashboard. Sistem kemudian meminta Admin/Operator untuk memasukkan Nomor Polisi (Nopol) melalui antarmuka "Data Parkir". Setelah Nopol dimasukkan, Sistem melakukan pencarian di Database. Database kemudian mengembalikan hasil pencarian Nopol ke antarmuka "Data Parkir" untuk ditampilkan, dan setelahnya alur kembali ke Dashboard.

2.4 Implementasi Kode Program

1. Login Form

Login Form adalah antarmuka login untuk aplikasi sistem parkir dengan tampilan modern (dengan background image/gradien), koneksi ke database, dan validasi berdasarkan role user.

a. Library

```
1 import javax.swing.*;
2 import java.awt.*;
3 import java.awt.image.BufferedImage;
4 import java.sql.Connection;
5 import java.sql.PreparedStatement;
6 import java.sql.ResultSet;
7 import javax.imageio.ImageIO;
```

Mengimpor pustaka yang dibutuhkan:

- javax.swing.*: Komponen GUI (seperti JFrame, JButton, JTextField, dll.)
- java.awt.*: Untuk layout dan grafik.
- java.awt.image.BufferedImage: Untuk gambar latar.
- java.sql.*: Untuk koneksi dan query ke database.
- javax.imageio.ImageIO: Untuk memuat gambar latar.

b. Konstruktor LoginForm()

```
1 public LoginForm() {  
2     initializeComponents();  
3     setupLayout();  
4     setupEventHandlers();  
5 }
```

Memanggil 3 metode utama:

- initializeComponents(): Inisialisasi komponen.
- setupLayout(): Menata layout.
- setupEventHandlers(): Menangani aksi (event handler).

c. Desain GUI

1) initializeComponents()

```
1 private void initializeComponents() {  
2     setTitle("Login Aplikasi Parkir");  
3     setExtendedState(JFrame.MAXIMIZED_BOTH);  
4     setDefaultCloseOperation(EXIT_ON_CLOSE);  
5     setLocationRelativeTo(null);  
6  
7     tfUser = new JTextField();  
8     pfPass = new JPasswordField();  
9     btnLogin = new JButton("LOGIN");  
10  
11     styleTextField(tfUser, "Username");  
12     styleTextField(pfPass, "Password");  
13     styleLoginButton();
```

Berfungsi Untuk :

- Menyetel ukuran jendela.
- Membuat field username dan password.
- Menambahkan style pada text field dan tombol login.

2) setupLayout()

```
1 private void setupLayout() {  
2     JPanel mainPanel = createMainPanel();  
3     JPanel loginPanel = createLoginPanel();  
4  
5     mainPanel.add(loginPanel);  
6     add(mainPanel);  
7 }
```

Berfungsi Untuk :

- Membuat mainPanel sebagai container utama.
- Menambahkan loginPanel ke dalam mainPanel.

3) createMainPanel()

```
1 private JPanel createMainPanel() {
2     return new JPanel(new GridBagLayout()) {
3         BufferedImage bgImage;
4         {
5             try {
6                 bgImage = ImageIO.read(getClass().getResource("/Uas_Pbo/img/parkir.png"));
7             } catch (Exception e) {
8                 bgImage = null;
9                 System.out.println("Background image tidak ditemukan, menggunakan warna default");
10            }
11        }
12
13        @Override
14        protected void paintComponent(Graphics g) {
15            super.paintComponent(g);
16            if (bgImage != null) {
17                g.drawImage(bgImage, 0, 0, getWidth(), getHeight(), this);
18            } else {
19                // Fallback gradient background
20                Graphics2D g2d = (Graphics2D) g;
21                GradientPaint gp = new GradientPaint(0, 0, new Color(25, 118, 210),
22                                                getWidth(), getHeight(), new Color(33, 150, 243));
23                g2d.setPaint(gp);
24                g2d.fillRect(0, 0, getWidth(), getHeight());
25            }
26        }
27    };
28}
```

Berfungsi Untuk :

Membuat panel dengan background image (parkir.png).

Jika gambar tidak ditemukan, digunakan gradien warna biru sebagai fallback.

4) createLoginPanel()

```
1 private JPanel createLoginPanel() {
2     JPanel loginPanel = new JPanel(new GridBagLayout()) {
3         @Override
4         protected void paintComponent(Graphics g) {
5             super.paintComponent(g);
6             Graphics2D g2d = (Graphics2D) g;
7             g2d.setRenderingHint(RenderingHints.KEY_ANTIALIASING, RenderingHints.VALUE_ANTIALIAS_ON);
8             g2d.setColor(Color.WHITE);
9             g2d.fillRoundRect(0, 0, getWidth(), getHeight(), 32, 32);
10            g2d.setColor(new Color(33, 150, 243, 40));
11            g2d.drawRoundRect(0, 0, getWidth()-1, getHeight()-1, 32, 32);
12        }
13    };
14}
```

Berfungsi Untuk :

Membuat pane login dengan tampilan rounded (sudut tumpul) dan warna putih semi-transparan.

5) addComponentsToLoginPage()

```
1  private void addComponentsToLoginPage(JPanel loginPanel) {
2      GridBagConstraints gbc = new GridBagConstraints();
3      gbc.insets = new Insets(16, 10, 16, 10);
4      gbc.fill = GridBagConstraints.HORIZONTAL;
5
6      // Title
7      JLabel lblTitle = createTitleLabel();
8      gbc.gridx = 0; gbc.gridy = 0; gbc.gridwidth = 2;
9      loginPanel.add(lblTitle, gbc);
10
11     // Subtitle
12     JLabel lblSubtitle = createSubtitleLabel();
13     gbc.gridy = 1;
14     loginPanel.add(lblSubtitle, gbc);
15
16     // Username panel
17     JPanel userPanel = createFieldPanel("Username:", tfUser);
18     gbc.insets = new Insets(24, 10, 8, 10);
19     gbc.gridy = 3;
20     loginPanel.add(userPanel, gbc);
21
22     // Password panel
23     JPanel passPanel = createFieldPanel("Password:", pfPass);
24     gbc.gridy = 4;
25     loginPanel.add(passPanel, gbc);
26
27     // Login button
28     gbc.insets = new Insets(32, 10, 10, 10);
29     gbc.gridy = 5;
30     loginPanel.add(btnLogin, gbc);
31
32     // Copyright
33     JLabel lblCopyright = createCopyrightLabel();
34     gbc.gridy = 6;
35     gbc.insets = new Insets(18, 10, 10, 10);
36     loginPanel.add(lblCopyright, gbc);
37 }
```

Berfungsi untuk menambahkan komponen ke dalam panel login menggunakan GridBagLayout:

- Label judul dan subjudul
- Panel field username dan password
- Tombol login
- Label copyright

6) Style Tambahan

```
1  private void styleTextField(JTextField field, String placeholder) {
2      field.setPreferredSize(new Dimension(320, 40));
3      field.setFont(new Font("Segoe UI", Font.PLAIN, 16));
4      field.setBorder(BorderFactory.createCompoundBorder(
5          BorderFactory.createLineBorder(new Color(33, 150, 243), 1),
6          BorderFactory.createEmptyBorder(5, 10, 5, 10)
7      ));
8      field.setBackground(new Color(245, 250, 255));
9      field.setToolTipText(placeholder);
10 }
```

```

1  private void styleLoginButton() {
2      btnLogin.setBackground(new Color(25, 118, 210));
3      btnLogin.setForeground(Color.WHITE);
4      btnLogin.setFocusPainted(false);
5      btnLogin.setBorderPainted(false);
6      btnLogin.setFont(new Font("Segoe UI", Font.BOLD, 18));
7      btnLogin.setPreferredSize(new Dimension(200, 45));
8      btnLogin.setCursor(new Cursor(Cursor.HAND_CURSOR));
9
10     // Hover effect
11     btnLogin.addMouseListener(new java.awt.event.MouseAdapter() {
12         public void mouseEntered(java.awt.event.MouseEvent evt) {
13             btnLogin.setBackground(new Color(33, 150, 243));
14         }
15         public void mouseExited(java.awt.event.MouseEvent evt) {
16             btnLogin.setBackground(new Color(25, 118, 210));
17         }
18     });
19 }

```

Berfungsi Untuk :

- styleTextField(): Desain untuk input username & password (warna latar, padding, border).
- styleLoginButton(): Desain tombol login (warna, hover effect, kursor).

d. Event Handling

setupEventHandlers ()

```

1  private void setupEventHandlers() {
2      btnLogin.addActionListener(e -> performLogin());
3
4      // Enter key pada password field
5      pfPass.addActionListener(e -> performLogin());
6
7      // Enter key pada username field
8      tfUser.addActionListener(e -> pfPass.requestFocus());
9  }

```

Berfungsi Untuk :

- Tombol login diklik → performLogin().
- Tekan Enter di field username → pindah ke password.
- Tekan Enter di field password → langsung login.

e. Proses Login (performLogin ())

```
1  private void performLogin() {
2      String username = tfUser.getText().trim();
3      String password = new String(pfPass.getPassword());
4
5      if (username.isEmpty() || password.isEmpty()) {
6          showMessage("Username dan password tidak boleh kosong!", "Peringatan", JOptionPane.WARNING_MESSAGE);
7          return;
8      }
9
10     try (Connection conn = DBConnection.getConnection()) {
11         String sql = "SELECT username, password, role FROM users WHERE username=? AND password=?";
12         PreparedStatement ps = conn.prepareStatement(sql);
13         ps.setString(1, username);
14         ps.setString(2, password);
15
16         ResultSet rs = ps.executeQuery();
17         if (rs.next()) {
18             String role = rs.getString("role");
19             dispose();
20             if ("operator".equalsIgnoreCase(role)) {
21                 new OptionOP(username, role).setVisible(true);
22             } else {
23                 new ParkirAppSystem(username, role).setVisible(true);
24             }
25         } else {
26             showMessage("Username atau password salah!", "Login Gagal", JOptionPane.ERROR_MESSAGE);
27             clearFields();
28         }
29     } catch (Exception ex) {
30         showMessage("Gagal koneksi ke database: " + ex.getMessage(), "Error", JOptionPane.ERROR_MESSAGE);
31     }
32 }
```

Urutan Proses :

- Ambil input username & password.
- Validasi input tidak kosong.
- Hubungkan ke database (DBConnection.getConnection()).
- Eksekusi query SQL untuk username dan password.
- Jika berhasil login: Ambil peran (role) user dan tampilkan window sesuai role: operator → OptionOP lainnya → ParkirAppSystem
- Jika gagal login: Tampilkan pesan kesalahan dan Kosongkan password.

2. Dashboard Admin

Dashboard admin adalah kelas dalam aplikasi parkir berbasis GUI (Swing) yang mengelola: Form kendaraan masuk & keluar, Tabel data parkir, Fitur pencarian, sortir, edit, hapus, cetak struk, dan Koneksi ke database (melalui DBConnection).

a. Library

```
1 import javax.swing.*;
2 import javax.swing.table.DefaultTableModel;
3 import java.awt.*;
4 import java.time.LocalDateTime;
5 import java.time.format.DateTimeFormatter;
6 import java.util.Vector;
7 import java.sql.*;
```

Penjelasan :

- javax.swing.* : Untuk komponen GUI (seperti tombol, tabel, frame).
- DefaultTableModel : Untuk model data untuk JTable.
- java.awt.* : Untuk desain antarmuka (warna, layout, font).
- LocalDateTime : Untuk mendapatkan tanggal dan waktu saat ini.
- DateTimeFormatter : Untuk format tampilan tanggal/waktu.
- Vector : Untuk struktur data dinamis (digunakan di tabel).
- java.sql.* : Untuk koneksi dan operasi database SQL.

b. Konstruktor

```
1 public ParkirAppSystem(String user, String role) {
2         this.currentRole = role;
3
4         initializeFrame();
5         createComponents();
6         setupLayout();
7         setupEventHandlers();
8         loadDataFromDatabase();
9         startClock();
10    }
```

Penjelasan :

- this.currentRole = role; → Menyimpan peran pengguna.
- initializeFrame(); → Menyiapkan jendela utama.
- createComponents(); → Membuat elemen GUI.
- setupLayout(); → Menata komponen di jendela.
- setupEventHandlers(); → Mengatur respon aksi pengguna.

- `loadDataFromDatabase();` → Mengambil data dari database.
- `startClock();` → Menjalankan jam/tampilan waktu.

c. Desain GUI

1) `initializeFrame()`

```

1  private void initializeFrame() {
2      setTitle("Aplikasi Parkir - Form Masuk & Keluar");
3      setDefaultCloseOperation(EXIT_ON_CLOSE);
4      setExtendedState(JFrame.MAXIMIZED_BOTH);
5      setLocationRelativeTo(null);
6  }

```

Berfungsi untuk :

- Mengatur judul jendela: "Aplikasi Parkir - Form Masuk & Keluar"
- Menutup aplikasi saat jendela ditutup (EXIT_ON_CLOSE)
- Membuka jendela dalam mode layar penuh (MAXIMIZED_BOTH)
- Menempatkan jendela di tengah layar (setLocationRelativeTo(null))
- Fungsi ini menyiapkan tampilan awal jendela utama aplikasi.

2) `createComponents()`

```

1  private void createComponents() {
2      // Text fields
3      tfPlatMasuk = new JTextField();
4      tfPlatKeluar = new JTextField();
5      tfSearch = new JTextField();
6
7      // Combo boxes
8      cbJenisMasuk = new JComboBox<>(new String[]{"Motor", "Mobil"});
9      cbSort = new JComboBox<>(new String[]{"A-Z", "Z-A"});
10
11     // Labels
12     lblJamMasuk = new JLabel("Auto-generated");
13     lblJamKeluar = new JLabel("Auto-generated");
14     lblDurasi_menit = new JLabel("-");
15     lblTarif = new JLabel("-");
16     lblClock = new JLabel();
17
18     // Setup styles
19     setupComponentStyles();
20 }

```

Berfungsi untuk :

- Text fields: Membuat 3 input teks, plat masuk, plat keluar, dan pencarian.

- Combo boxes: Dropdown untuk memilih jenis kendaraan (Motor/Mobil) dan urutan data (A-Z/Z-A).
- Labels: Label untuk menampilkan jam masuk, jam keluar, durasi, tarif, dan jam saat ini.
- setupComponentStyles(): Memanggil fungsi untuk mengatur tampilan komponen.
- Fungsi ini bertugas membuat semua elemen GUI yang dibutuhkan.

3) setupLayout()

```

1  private void setupLayout() {
2      JPanel mainPanel = new JPanel(new BorderLayout());
3      mainPanel.setBackground(new Color(232, 245, 253));
4
5      // Menu bar
6      setJMenuBar(createMenuBar());
7
8      // Clock panel
9      mainPanel.add(createClockPanel(), BorderLayout.NORTH);
10
11     // Tabbed pane for forms
12     mainPanel.add(createTabbedPane(), BorderLayout.CENTER);
13
14     // Table and search panel
15     mainPanel.add(createSouthPanel(), BorderLayout.SOUTH);
16
17     setContentPane(mainPanel);
18 }
```

Berfungsi untuk :

- Membuat mainPanel dengan layout BorderLayout dan latar belakang berwarna biru muda.
- Menambahkan: Menu bar (setJMenuBar), Panel jam di atas (NORTH), Tabbed form di tengah (CENTER), Tabel & pencarian di bawah (SOUTH)
- Menetapkan mainPanel sebagai konten utama jendela (setContentPane)
- Fungsi ini mengatur tata letak seluruh komponen GUI dalam jendela aplikasi.

4) createMenuBar ()

```
1  private JMenuBar createMenuBar() {
2      JMenuBar menuBar = new JMenuBar();
3      menuBar.setBackground(new Color(25, 118, 210));
4
5      JMenu menu = new JMenu(" Menu");
6      menu.setFont(new Font("Arial", Font.BOLD, 16));
7      menu.setForeground(Color.WHITE);
8      menu.setOpaque(true);
9      menu.setBackground(new Color(25, 118, 210));
10
11     // Print menu item
12     JMenuItem printItem = new JMenuItem("Cetak Struk");
13     styleMenuItem(printItem, new Color(76, 175, 80));
14     printItem.setBackground(new Color(76, 175, 80));
15     printItem.setForeground(Color.WHITE);
16     printItem.setOpaque(true);
17     printItem.addActionListener(e -> showPrintDialog());
18     menu.add(printItem);
19
20     // Delete menu item
21     JMenuItem deleteItem = new JMenuItem("Hapus Data");
22     styleMenuItem(deleteItem, new Color(33, 158, 243));
23     deleteItem.setBackground(new Color(33, 158, 243));
24     deleteItem.setForeground(Color.WHITE);
25     deleteItem.setOpaque(true);
26     deleteItem.addActionListener(e -> deleteSelectedRow());
27     menu.add(deleteItem);
28
29     JMenuItem editItem = new JMenuItem("Edit Data Parkir");
30     styleMenuItem(editItem, new Color(255, 152, 0));
31     editItem.setBackground(new Color(255, 152, 0));
32     editItem.setForeground(Color.WHITE);
33     editItem.setOpaque(true);
34     editItem.addActionListener(e -> showEditDialog());
35     menu.add(editItem);
36
37     menu.addSeparator();
38
39     // Logout menu item
40     JMenuItem logoutItem = new JMenuItem("Logout");
41     styleMenuItem(logoutItem, new Color(244, 67, 54));
42     logoutItem.setBackground(new Color(244, 67, 54));
43     logoutItem.setForeground(Color.WHITE);
44     logoutItem.setOpaque(true);
45     logoutItem.addActionListener(e -> logout());
46     menu.add(logoutItem);
47
48     // Set warna background menu bar
49     menuBar.setOpaque(true);
50     menuBar.setBackground(new Color(25, 118, 210));
51     menuBar.add(menu);
52
53 }
```

Berfungsi untuk :

- Membuat JMenuBar dengan menu utama bernama "Menu".
- Menambahkan Menu Cetak Struk untuk menjalankan showPrintDialog().
- Menambahkan Menu Hapus Data untuk menjalankan deleteSelectedRow().
- Menambahkan Menu Edit Data Parkir untuk menjalankan showEditDialog().
- Menambahkan Menu Logout untuk menjalankan logout().

- Setiap item menu diberi warna latar dan teks khusus serta dibuat transparan (setOpaque(true)).
- Menambahkan separator sebelum item Logout.
- Mengatur latar belakang menu bar dan mengembalikannya.

5) createClockPanel ()

```

1 private JPanel createClockPanel() {
2     JPanel panel = new JPanel(new BorderLayout());
3     panel.setBackground(new Color(25, 118, 210));
4     panel.setPreferredSize(new Dimension(1000, 50));
5
6     lblClock.setFont(new Font("Arial", Font.BOLD, 20));
7     lblClock.setForeground(Color.WHITE);
8     lblClock.setHorizontalAlignment(SwingConstants.RIGHT);
9     panel.add(lblClock, BorderLayout.EAST);
10
11    JLabel title = new JLabel(" SISTEM PARKIR", SwingConstants.LEFT);
12    title.setFont(new Font("Arial", Font.BOLD, 28));
13    title.setForeground(Color.WHITE);
14    panel.add(title, BorderLayout.WEST);
15
16    return panel;
17 }
```

Berfungsi untuk :

- Membuat panel dengan layout horizontal (BorderLayout) dan ukuran tetap.
- Menambahkan label jam (lblClock) di kanan (EAST), dengan font putih dan rata kanan.
- Menambahkan judul "SISTEM PARKIR" di kiri (WEST), dengan font besar dan putih.
- Mengatur warna latar belakang panel menjadi biru.

6) createTabbedPane ()

```

1 private JTabbedPane createTabbedPane() {
2     JTabbedPane tabbedPane = new JTabbedPane();
3     tabbedPane.setFont(new Font("Arial", Font.BOLD, 15));
4     tabbedPane.setBackground(new Color(197, 225, 165));
5
6     JPanel masukPanel = createMasukPanel();
7     JPanel keluarPanel = createKeluarPanel();
8
9     masukPanel.setPreferredSize(new Dimension(400, 300));
10    keluarPanel.setPreferredSize(new Dimension(400, 300));
11
12    tabbedPane.addTab("Kendaraan Masuk", masukPanel);
13    tabbedPane.addTab("Kendaraan Keluar", keluarPanel);
14
15    return tabbedPane;
16 }
```

Berfungsi untuk :

- Membuat JTabbedPane dengan font dan warna latar khusus.
- Menambahkan tab "Kendaraan Masuk" : panel untuk data masuk.
- Menambahkan tab "Kendaraan Keluar" : panel untuk data keluar.
- Masing-masing panel diatur ukurannya (400×300 piksel).

7) createMasukPanel ()

```
1  private JPanel createMasukPanel() {  
2      JPanel panel = new JPanel(new GridBagLayout());  
3      panel.setBackground(new Color(232, 245, 253));  
4  
5      GridBagConstraints gbc = new GridBagConstraints();  
6      gbc.insets = new Insets(0, 0, 0, 0);  
7      gbc.fill = GridBagConstraints.BOTH;  
8      gbc.weightx = 1.0;  
9      gbc.weighty = 1.0;  
10  
11      JPanel formPanel = createMasukFormPanel();  
12      panel.add(formPanel, gbc);  
13  
14      return panel;  
15  }
```

Berfungsi untuk :

- Membuat JPanel dengan GridBagLayout.
- Mengatur warna latar belakang panel (biru muda).
- Mengatur GridBagConstraints (isi penuh, tanpa margin).
- Membuat formPanel dari metode createMasukFormPanel().
- Menambahkan formPanel ke panel utama.
- Mengembalikan panel sebagai hasil metode.

8) createMasukFormPanel ()

```
1  private JPanel createMasukFormPanel() {
2      JPanel mainFormPanel = new JPanel(new GridBagLayout());
3      mainFormPanel.setBackground(new Color(232, 245, 253));
4
5      JPanel leftPanel = createMasukLeftPanel();
6      JPanel rightPanel = createMasukRightPanel();
7
8      GridBagConstraints gbc = new GridBagConstraints();
9      gbc.insets = new Insets(0, 0, 0, 0);
10     gbc.fill = GridBagConstraints.BOTH;
11     gbc.gridx = 0;
12
13     gbc.gridx = 0;
14     gbc.weightx = 0.8;
15     gbc.gridy = 1;
16     mainFormPanel.add(leftPanel, gbc);
17
18     gbc.gridx = 1;
19     gbc.weightx = 0.2;
20     mainFormPanel.add(rightPanel, gbc);
21
22     return mainFormPanel;
23 }
```

Berfungsi untuk :

- Membuat panel utama mainFormPanel dengan GridBagLayout.
- Mengatur warna latar belakang (biru muda).
- Membuat dua sub-panel: leftPanel dan rightPanel.
- Menyiapkan GridBagConstraints untuk pengaturan tata letak.
- Menambahkan leftPanel ke kolom kiri (80% lebar).
- Menambahkan rightPanel ke kolom kanan (20% lebar).
- Mengembalikan mainFormPanel sebagai hasil.

9) createMasukLeftPanel ()

```
1  private JPanel createMasukLeftPanel() {
2      JPanel leftPanel = new JPanel(new GridBagLayout());
3      leftPanel.setBackground(Color.WHITE);
4      leftPanel.setBorder(BorderFactory.createCompoundBorder(
5          BorderFactory.createLineBorder(new Color(33, 150, 243), 2),
6          BorderFactory.createEmptyBorder(18, 18, 18, 18)
7      ));
8
9      Font labelFont = new Font("Segoe UI", Font.BOLD, 15);
10
11     JLabel lblTitle = new JLabel("Form Kendaraan Masuk", JLabel.CENTER);
12     lblTitle.setFont(new Font("Segoe UI", Font.BOLD, 18));
13     lblTitle.setForeground(new Color(33, 150, 243));
14
15     GridBagConstraints gbc = new GridBagConstraints();
16     gbc.insets = new Insets(10, 10, 10, 10);
17     gbc.fill = GridBagConstraints.HORIZONTAL;
18
19     gbc.gridx = 0; gbc.gridy = 0; gbc.gridwidth = 2;
20     leftPanel.add(lblTitle, gbc);
21
22     gbc.gridwidth = 1;
23     gbc.gridx = 0; gbc.gridy = 1;
24     leftPanel.add(createLabel("Plat Nomor:", labelFont), gbc);
25     gbc.gridx = 1;
26     leftPanel.add(tfPlatMasuk, gbc);
27
28     gbc.gridx = 0; gbc.gridy = 2;
29     leftPanel.add(createLabel("Jenis Kendaraan:", labelFont), gbc);
30     gbc.gridx = 1;
31     leftPanel.add(cbJenisMasuk, gbc);
32
33     gbc.gridx = 0; gbc.gridy = 3;
34     leftPanel.add(createLabel("Tanggal Masuk:", labelFont), gbc);
35     gbc.gridx = 1;
36     leftPanel.add(lblJamMasuk, gbc);
37
38     return leftPanel;
39 }
```

Berfungsi untuk :

- Membuat panel leftPanel dengan GridBagLayout dan latar belakang putih.
- Menambahkan border garis dan padding pada panel.
- Mengatur font dan warna untuk judul form.
- Menambahkan label judul “Form Kendaraan Masuk”.
- Menambahkan label dan field untuk: Plat Nomor, Jenis Kendaraan, Tanggal Masuk
- Mengatur posisi setiap komponen menggunakan GridBagConstraints.
- Mengembalikan leftPanel sebagai hasil.

10) createMasukRightPanel ()

```
1 private JPanel createMasukRightPanel() {
2     JPanel rightPanel = new JPanel(new BorderLayout());
3     rightPanel.setBackground(Color.WHITE);
4     rightPanel.setBorder(BorderFactory.createCompoundBorder(
5         BorderFactory.createLineBorder(new Color(33, 150, 243), 2),
6         BorderFactory.createEmptyBorder(18, 18, 18, 18)
7     ));
8
9     JButton btnMasuk = new JButton("Simpan Masuk");
10    styleButton(btnMasuk, new Color(76, 175, 80));
11    btnMasuk.setFont(new Font("Segoe UI", Font.BOLD, 28));
12    btnMasuk.addActionListener(e -> handleKendaraanMasuk());
13
14    rightPanel.add(btnMasuk, BorderLayout.CENTER);
15
16    return rightPanel;
17 }
```

Berfungsi untuk :

- Membuat panel rightPanel dengan BorderLayout dan latar belakang putih.
- Menambahkan border garis dan padding pada panel.
- Membuat tombol "Simpan Masuk" dengan warna dan font khusus.
- Menambahkan aksi saat tombol ditekan (memanggil handleKendaraanMasuk()).
- Menempatkan tombol di tengah panel.
- Mengembalikan rightPanel sebagai hasil.

11) createKeluarPanel ()

```
1 private JPanel createKeluarPanel() {
2     JPanel panel = new JPanel(new GridBagLayout());
3     panel.setBackground(new Color(232, 245, 253));
4
5     GridBagConstraints gbc = new GridBagConstraints();
6     gbc.insets = new Insets(0, 0, 0, 0);
7     gbc.fill = GridBagConstraints.BOTH;
8     gbc.weightx = 1.0;
9     gbc.weighty = 1.0;
10
11    JPanel formPanel = createKeluarFormPanel();
12    panel.add(formPanel, gbc);
13
14    return panel;
15 }
```

Berfungsi untuk :

- Membuat panel utama dengan GridBagLayout.
- Mengatur warna latar belakang (biru muda).

- Menyiapkan GridBagConstraints untuk mengatur tata letak (isi penuh).
- Membuat panel form dengan memanggil createKeluarFormPanel().
- Menambahkan form panel ke panel utama.
- Mengembalikan panel utama sebagai hasil.

12) createKeluarFormPanel ()

```

1  private JPanel createKeluarFormPanel() {
2      JPanel mainFormPanel = new JPanel(new GridBagLayout());
3      mainFormPanel.setBackground(new Color(232, 245, 253));
4
5      JPanel leftPanel = createKeluarLeftPanel();
6      JPanel rightPanel = createKeluarRightPanel();
7
8      GridBagConstraints gbc = new GridBagConstraints();
9      gbc.insets = new Insets(0, 0, 0, 0);
10     gbc.fill = GridBagConstraints.BOTH;
11     gbc.gridx = 0;
12
13     gbc.gridx = 0;
14     gbc.weightx = 0.8;
15     gbc.weighty = 1.0;
16     mainFormPanel.add(leftPanel, gbc);
17
18     gbc.gridx = 1;
19     gbc.weightx = 0.2;
20     mainFormPanel.add(rightPanel, gbc);
21
22     return mainFormPanel;
23 }
```

Berfungsi untuk :

- Membuat panel utama mainFormPanel dengan GridBagLayout dan latar biru muda.
- Membuat dua sub-panel: leftPanel dan rightPanel.
- Mengatur GridBagConstraints agar komponen menyesuaikan ruang secara proporsional.
- Menambahkan leftPanel (80% lebar) dan rightPanel (20% lebar) ke dalam mainFormPanel.
- Mengembalikan mainFormPanel sebagai hasil.

13) createKeluarLeftPanel ()

```
1  private JPanel createKeluarLeftPanel() {  
2      JPanel leftPanel = new JPanel(new GridBagLayout());  
3      leftPanel.setBackground(Color.WHITE);  
4      leftPanel.setBorder(BorderFactory.createCompoundBorder(  
5          BorderFactory.createLineBorder(new Color(33, 150, 243), 2),  
6          BorderFactory.createEmptyBorder(18, 18, 18, 18)  
7      ));  
8  
9      Font labelFont = new Font("Segoe UI", Font.BOLD, 15);  
10  
11     JLabel lblTitle = new JLabel("Form Kendaraan Keluar", JLabel.CENTER);  
12     lblTitle.setFont(new Font("Segoe UI", Font.BOLD, 18));  
13     lblTitle.setForeground(new Color(33, 150, 243));  
14  
15     GridBagConstraints gbc = new GridBagConstraints();  
16     gbc.insets = new Insets(10, 10, 10, 10);  
17     gbc.fill = GridBagConstraints.HORIZONTAL;  
18  
19     gbc.gridx = 0; gbc.gridy = 0; gbc.gridwidth = 2;  
20     leftPanel.add(lblTitle, gbc);  
21  
22     gbc.gridwidth = 1;  
23     gbc.gridx = 0; gbc.gridy = 1;  
24     leftPanel.add(createLabel("Plat Nomor:", labelFont), gbc);  
25     gbc.gridx = 1;  
26     leftPanel.add(tfPlatKeluar, gbc);  
27  
28     gbc.gridx = 0; gbc.gridy = 2;  
29     leftPanel.add(createLabel("Tanggal Keluar:", labelFont), gbc);  
30     gbc.gridx = 1;  
31     leftPanel.add(lblJamKeluar, gbc);  
32  
33     gbc.gridx = 0; gbc.gridy = 3;  
34     leftPanel.add(createLabel("Durasi Parkir:", labelFont), gbc);  
35     gbc.gridx = 1;  
36     leftPanel.add(lbldurasi_menit, gbc);  
37  
38     gbc.gridx = 0; gbc.gridy = 4;  
39     leftPanel.add(createLabel("Tarif Parkir:", labelFont), gbc);  
40     gbc.gridx = 1;  
41     leftPanel.add(lblTarif, gbc);  
42  
43     return leftPanel;  
44 }
```

Berfungsi untuk :

- Membuat panel kiri (leftPanel) dengan GridBagLayout dan latar putih.
- Menambahkan border garis biru dan padding.
- Menyiapkan font dan menambahkan judul “Form Kendaraan Keluar”.
- Menambahkan label dan komponen input/output untuk: Plat Nomor, Tanggal Keluar, Durasi Parkir, Tarif Parkir
- Mengatur tata letak menggunakan GridBagConstraints.
- Mengembalikan leftPanel sebagai hasil.

14) createKeluarRightPanel ()

```
1  private JPanel createKeluarRightPanel() {
2      JPanel rightPanel = new JPanel(new BorderLayout());
3      rightPanel.setBackground(Color.WHITE);
4      rightPanel.setBorder(BorderFactory.createCompoundBorder(
5          BorderFactory.createLineBorder(new Color(33, 150, 243), 2),
6          BorderFactory.createEmptyBorder(18, 18, 18, 18)
7      ));
8
9      JButton btnKeluar = new JButton("Simpan Keluar");
10     styleButton(btnKeluar, new Color(255, 152, 0));
11     btnKeluar.setFont(new Font("Segoe UI", Font.BOLD, 28));
12     btnKeluar.addActionListener(e -> handleKendaraanKeluar());
13
14     rightPanel.add(btnKeluar, BorderLayout.CENTER);
15
16     return rightPanel;
17 }
```

Berfungsi untuk :

- Membuat panel kanan (rightPanel) dengan BorderLayout dan latar putih.
- Menambahkan border garis biru dan padding.
- Membuat tombol "Simpan Keluar" dengan warna merah muda dan font besar.
- Menambahkan aksi klik tombol yang memanggil fungsi handleKendaraanKeluar().
- Menempatkan tombol di tengah panel.
- Mengembalikan rightPanel sebagai hasil.

15) createSeacrhPanel ()

```
1  private JPanel createSouthPanel() {
2      JPanel southPanel = new JPanel(new BorderLayout());
3      southPanel.setBackground(new Color(232, 245, 253));
4
5      southPanel.add(createSearchPanel(), BorderLayout.NORTH);
6      southPanel.add(createTablePanel(), BorderLayout.CENTER);
7
8      return southPanel;
9  }
10
11 private JPanel createSearchPanel() {
12     JPanel searchPanel = new JPanel(new GridBagLayout());
13     searchPanel.setBackground(new Color(232, 245, 253));
14
15     GridBagConstraints gbc = new GridBagConstraints();
16     gbc.insets = new Insets(8, 8, 8, 8);
17     gbc.fill = GridBagConstraints.HORIZONTAL;
18
19     JLabel lblSearch = new JLabel("Cari/Filter Plat Nomor: ");
20     lblSearch.setFont(new Font("Arial", Font.BOLD, 15));
21
22     JButton btnSearch = new JButton("Cari");
23     styleButton(btnSearch, new Color(33, 150, 243));
24     btnSearch.setFont(new Font("Arial", Font.BOLD, 14));
25     btnSearch.setPreferredSize(new Dimension(100, 32));
26
27     JButton btnReset = new JButton("Reset");
28     styleButton(btnReset, new Color(76, 175, 80));
29     btnReset.setFont(new Font("Arial", Font.BOLD, 14));
30     btnReset.setPreferredSize(new Dimension(100, 32));
31
32     JLabel lblSort = new JLabel("Urutkan: ");
33     lblSort.setFont(new Font("Arial", Font.BOLD, 15));
34
35     gbc.gridx = 0; gbc.gridy = 0; gbc.weightx = 0;
36     searchPanel.add(lblSearch, gbc);
37     gbc.gridx = 1; gbc.weightx = 1;
38     searchPanel.add(tfSearch, gbc);
39     gbc.gridx = 2; gbc.weightx = 0;
40     searchPanel.add(btnSearch, gbc);
41     gbc.gridx = 3;
42     searchPanel.add(btnReset, gbc);
43     gbc.gridx = 4;
44     searchPanel.add(Box.createHorizontalStrut(20), gbc);
45     gbc.gridx = 5;
46     searchPanel.add(lblSort, gbc);
47     gbc.gridx = 6;
48     searchPanel.add(cbSort, gbc);
49
50     // Event handlers untuk search
51     btnSearch.addActionListener(e -> performSearch());
52     tfSearch.addActionListener(e -> performSearch());
53     btnReset.addActionListener(e -> resetSearch());
54     cbSort.addActionListener(e -> performSearch());
55
56     return searchPanel;
57 }
```

Berfungsi untuk :

- Membuat panel bawah (southPanel) dengan BorderLayout dan latar biru muda.
- Menambahkan dua panel: panel pencarian (searchPanel) di atas dan panel tabel di tengah.

- Menyusun searchPanel dengan GridBagLayout.
- Menambahkan komponen: label, text field, tombol "Cari" & "Reset", dan combo box urutan.
- Memberi warna latar dan mengatur ukuran serta font komponen.
- Menambahkan aksi klik pada tombol dan input untuk pencarian dan reset.
- Mengembalikan southPanel sebagai hasil.

16) createTablePanel ()

```

1  private JPanel createTablePanel() {
2      initTable();
3      JScrollPane scrollPane = new JScrollPane(table);
4      scrollPane.setPreferredSize(new Dimension(1000, 400));
5      scrollPane.setBorder(BorderFactory.createTitledBorder(
6          BorderFactory.createLineBorder(new Color(33, 150, 243), 2),
7          "Data Parkir",
8          0, 0,
9          new Font("Arial", Font.BOLD, 16),
10         new Color(33, 150, 243)
11     ));
12
13     JPanel panel = new JPanel(new BorderLayout());
14     panel.add(scrollPane, BorderLayout.CENTER);
15
16     return panel;
17 }
```

Berfungsi untuk :

- Membuat panel tabel (tablePanel) dengan BorderLayout.
- Menginisialisasi tabel melalui initTable().
- Membungkus tabel dalam JScrollPane berukuran 1000×400 piksel.
- Menambahkan border dengan judul "Data Parkir", garis biru, dan font tebal.
- Menempatkan scrollPane di tengah panel.
- Mengembalikan panel sebagai hasil.

17) loadDataFromDatabase()

```
1  private void loadDataFromDatabase() {
2      tableView.setModel(tableModel);
3      try (Connection conn = DBConnection.getConnection()) {
4          String sql = "SELECT plat, jenis, tanggal_masuk, IFNULL(tanggal_keluar, '') as tanggal_keluar, " +
5                      "IFNULL(durasi_menit, '') as durasi_menit, IFNULL(tarif, '') as tarif FROM parkir";
6          Statement stmt = conn.createStatement();
7          ResultSet rs = stmt.executeQuery(sql);
8
9          while (rs.next()) {
10              Vector<String> row = new Vector<>();
11              row.add(rs.getString("plat"));
12              row.add(rs.getString("jenis"));
13              // Format tanggal masuk dan keluar ke dd/MM/yyyy HH:mm:ss jika ada
14              String tglMasuk = rs.getString("tanggal_masuk");
15              String tglKeluar = rs.getString("tanggal_keluar");
16              if (tglMasuk != null && !tglMasuk.isEmpty()) {
17                  try {
18                      LocalDateTime ldt = LocalDateTime.parse(tglMasuk, DateTimeFormatter.ofPattern("yyyy-MM-dd HH:mm:ss"));
19                      tglMasuk = ldt.format(DateTimeFormatter.ofPattern("dd/MM/yyyy HH:mm:ss"));
20                  } catch (Exception e) {}
21              }
22              if (tglKeluar != null && !tglKeluar.isEmpty()) {
23                  try {
24                      LocalDateTime ldt = LocalDateTime.parse(tglKeluar, DateTimeFormatter.ofPattern("yyyy-MM-dd HH:mm:ss"));
25                      tglKeluar = ldt.format(DateTimeFormatter.ofPattern("dd/MM/yyyy HH:mm:ss"));
26                  } catch (Exception e) {}
27              }
28              row.add(tglMasuk);
29              row.add(tglKeluar);
30              row.add(rs.getString("durasi_menit"));
31              row.add(rs.getString("tarif"));
32              tableModel.addRow(row);
33          }
34      } catch (Exception ex) {
35          showMessage("Gagal load data dari database: " + ex.getMessage(), "Error", JOptionPane.ERROR_MESSAGE);
36      }
37  }
```

Berfungsi untuk :

- Mengambil dan menampilkan data parkir dari database ke tabel.
- Menjalankan query SQL untuk mengambil plat, jenis, tanggal masuk/keluar, durasi, dan tarif.
- Memformat tanggal masuk dan keluar ke format dd/MM/yyyy HH:mm:ss.
- Menambahkan setiap baris data ke tableModel.
- Menangani error dengan menampilkan pesan jika gagal memuat data.

18) startClock()

```
1  private void startClock() {
2      timer = new Timer(1000, e -> {
3          DateTimeFormatter formatter = DateTimeFormatter.ofPattern("dd-MM-yyyy HH:mm:ss");
4          lblClock.setText("Waktu Sekarang: " + LocalDateTime.now().format(formatter));
5      });
6      timer.start();
7  }
```

Berfungsi untuk :

- Menampilkan jam real-time pada label lblClock.
- Mengatur format waktu menjadi dd-MM-yyyy HH:mm:ss.
- Memperbarui teks setiap detik menggunakan Timer.
- Menampilkan teks: "Waktu Sekarang: [waktu saat ini]".

3. Dashboard Operator

Dashboard Operator adalah kelas Java Swing yang menampilkan dashboard utama untuk operator parkir setelah login. Di sini, user bisa memilih menu Kendaraan Masuk atau Kendaraan Keluar, melihat jam digital, dan melakukan logout.

a. Library

```
1 import javax.swing.*;  
2 import java.awt.*;  
3 import java.awt.event.*;  
4 import java.time.LocalDateTime;  
5 import java.time.format.DateTimeFormatter;
```

Penjelasan :

- javax.swing.* : Untuk menyediakan komponen-komponen untuk membangun antarmuka grafis (GUI), seperti JFrame (jendela), JButton (tombol), dan JLabel (teks).
- java.awt.* : Untuk menyediakan fungsionalitas dasar GUI, seperti kelas untuk Layout, Color (warna), dan Font.
- java.awt.event.* : Untuk menangani berbagai macam *event* atau aksi dari pengguna, seperti klik tombol, gerakan mouse, atau input keyboard.
- java.time.LocalDateTime : Untuk mendapatkan dan memanipulasi informasi tanggal dan waktu (tanpa zona waktu).
- java.time.format.DateTimeFormatter : Untuk mengubah format penampilan tanggal dan waktu agar sesuai dengan yang diinginkan.

b. Konstruktor

```
1  public OptionOP(String username, String role) {
2      this.username = username;
3      this.role = role;
4
5      setTitle("Parkir Management System - Dashboard");
6      setDefaultCloseOperation(EXIT_ON_CLOSE);
7      setExtendedState(JFrame.MAXIMIZED_BOTH);
8      setLocationRelativeTo(null);
9
10     JPanel mainPanel = new JPanel(new GridBagLayout()) {
11         Image bgImage;
12         {
13             try {
14                 bgImage = javax.imageio.ImageIO.read(getClass().getResource("/Uas_Pbo/img/parkir2.png"));
15             } catch (Exception e) {
16                 bgImage = null;
17             }
18         }
19         @Override
20         protected void paintComponent(Graphics g) {
21             super.paintComponent(g);
22             if (bgImage != null) {
23                 g.drawImage(bgImage, 0, 0, getWidth(), getHeight(), this);
24             } else {
25                 // fallback biru
26                 Graphics2D g2d = (Graphics2D) g;
27                 g2d.setColor(new Color(25, 118, 210));
28                 g2d.fillRect(0, 0, getWidth(), getHeight());
29             }
30         }
31     };
32     mainPanel.setLayout(new GridBagLayout());
33
34     JPanel card = createMainCard();
35     mainPanel.add(card);
36     setContentPane(mainPanel);
37
38     startClock();
39 }
```

Berfungsi untuk :

- Constructor OptionOP(...): Dieksekusi saat objek dibuat; menyimpan username dan role.
- setTitle(...): Menetapkan judul jendela.
- setDefaultCloseOperation(EXIT_ON_CLOSE): Keluar dari aplikasi saat ditutup.
- setExtendedState(JFrame.MAXIMIZED_BOTH): Jendela terbuka penuh layar.
- setLocationRelativeTo(null): Posisikan jendela di tengah layar.
- setContentPane(mainPanel): Atur mainPanel sebagai isi utama jendela.
- startClock(): Jalankan jam digital di UI.
- Dibuat lewat createMainCard() (berisi komponen utama).
- Ditambahkan ke mainPanel.
- mainPanel (JPanel khusus):

c. Desain GUI

1) createMainCard

```
1 private JPanel createMainCard() {
2     JPanel card = new JPanel() {
3         @Override
4         protected void paintComponent(Graphics g) {
5             super.paintComponent(g);
6             Graphics2D g2d = (Graphics2D) g;
7             g2d.setRenderingHint(RenderingHints.KEY_ANTIALIASING, RenderingHints.VALUE_ANTIALIAS_ON);
8
9             g2d.setColor(Color.WHITE);
10            g2d.fillRoundRect(0, 0, getWidth()-1, getHeight()-1, 24, 24);
11            g2d.setColor(new Color(33, 150, 243, 180));
12            g2d.setStroke(new BasicStroke(3));
13            g2d.drawRoundRect(0, 0, getWidth()-1, getHeight()-1, 24, 24);
14        }
15    };
}
```

Berfungsi untuk :

- Membuat Panel Khusus : Kode membuat sebuah JPanel dengan perilaku yang dimodifikasi melalui override metode paintComponent. Metode ini berfungsi menggambar tampilan visual panel secara manual sesuai desain yang diinginkan.
- Menghaluskan Grafis (Anti-Aliasing) : Sebelum menggambar, diaktifkan fitur anti-aliasing melalui objek Graphics2D. Ini bertujuan agar garis, lengkungan, dan bentuk tampak lebih halus, tidak bergerigi, sehingga tampilan lebih estetis.
- Menggambar Latar Belakang: Panel menggambar sebuah persegi panjang bersudut tumpul (RoundRectangle2D) yang diisi dengan warna putih. Bentuk ini menjadi latar utama dari panel, menyerupai kartu atau frame.
- Menggambar Bingkai (Border): Setelah latar selesai, bentuk yang sama digambar ulang namun hanya sebagai garis tepi (tanpa isi) menggunakan metode draw. Bingkai ini dibuat dengan ketebalan 3 piksel dan menggunakan warna biru transparan untuk memberi kesan modern dan profesional.

2) createHeaderPanel ()

```
1  private JPanel createHeaderPanel() {
2      JPanel headerPanel = new JPanel();
3      headerPanel.setLayout(new BoxLayout(headerPanel, BoxLayout.Y_AXIS));
4      headerPanel.setOpaque(false);
5
6      // Welcome section
7      JLabel welcomeLabel = new JLabel("Selamat Datang!");
8      welcomeLabel.setFont(new Font("Segoe UI", Font.BOLD, 28));
9      welcomeLabel.setForeground(Color.BLACK);
10     welcomeLabel.setAlignmentX(Component.CENTER_ALIGNMENT);
11
12     // Clock
13     JLabel clockLabel = new JLabel();
14     clockLabel.setFont(new Font("Segoe UI", Font.BOLD, 16));
15     clockLabel.setForeground(Color.BLACK);
16     clockLabel.setAlignmentX(Component.CENTER_ALIGNMENT);
17
18     headerPanel.add(welcomeLabel);
19     headerPanel.add(Box.createVerticalStrut(10));
20     headerPanel.add(clockLabel);
21     headerPanel.add(Box.createVerticalStrut(10));
22
23     return headerPanel;
24 }
```

Berfungsi untuk :

- Membuat JPanel baru bernama headerPanel.
- Mengatur layout ke BoxLayout vertikal (Y_AXIS) agar komponen tersusun dari atas ke bawah.
- setOpaque(false) membuat panel transparan.
- Membuat JLabel dengan teks "Selamat Datang!".
- Mengatur font dan ukuran tulisan.
- Warna teks diatur menjadi hitam.
- Label diatur agar berada di tengah horizontal (CENTER_ALIGNMENT).
- Membuat label kosong untuk jam (kemungkinan nanti akan diisi oleh timer).
- Font lebih kecil dari welcome label.
- Penyesuaian warna dan posisi alignment sama seperti welcomeLabel.
- Menambahkan label ke dalam panel.
- Box.createVerticalStrut(10) menambahkan spasi vertikal 10 piksel antar komponen.

3) createMenuPanel ()

```
1  private JPanel createMenuPanel() {
2      JPanel menuPanel = new JPanel();
3      menuPanel.setLayout(new BoxLayout(menuPanel, BoxLayout.Y_AXIS));
4      menuPanel.setOpaque(false);
5
6      // Menu title
7      JLabel menuTitle = new JLabel("Pilih Menu Operasi");
8      menuTitle.setFont(new Font("Segoe UI", Font.BOLD, 22));
9      menuTitle.setForeground(new Color(55, 65, 81));
10     menuTitle.setAlignmentX(Component.CENTER_ALIGNMENT);
11
12    menuPanel.add(menuTitle);
13    menuPanel.add(Box.createVerticalStrut(30));
14
15    // Create menu cards
16    JPanel cardsContainer = new JPanel();
17    cardsContainer.setLayout(new BoxLayout(cardsContainer, BoxLayout.Y_AXIS));
18    cardsContainer.setOpaque(false);
19
20    // Card Kendaraan Masuk
21    JPanel entryCard = createMenuCard(
22        "Kendaraan Masuk",
23        "Registrasi kendaraan yang masuk area parkir",
24        null, null,
25        e -> {
26            new Masuk(username, role).setVisible(true);
27            dispose();
28        }
29    );
30
31    // Card Kendaraan Keluar
32    JPanel exitCard = createMenuCard(
33        "Kendaraan Keluar",
34        "Proses kendaraan yang akan keluar parkir",
35        null, null,
36        e -> {
37            new Keluar(username, role).setVisible(true);
38            dispose();
39        }
40    );
41
42    cardsContainer.add(entryCard);
43    cardsContainer.add(Box.createVerticalStrut(20));
44    cardsContainer.add(exitCard);
45
46    menuPanel.add(cardsContainer);
47
48    return menuPanel;
49 }
```

Berfungsi untuk :

- Menampilkan panel menu utama dalam antarmuka aplikasi.
- Menambahkan judul menu “Pilih Menu Operasi” dan mengatur gaya tampilannya (font, warna, dan perataan).
- Kendaraan Masuk: membuka form registrasi kendaraan masuk.
- Kendaraan Keluar: membuka form proses kendaraan keluar.

- Menangani aksi klik tombol untuk masing-masing menu dengan membuka form yang sesuai dan menutup menu utama.
- Mengatur tata letak panel dan komponen secara vertikal menggunakan BoxLayout untuk tampilan yang rapi.

4) createMenuCard ()

```

1 private JPanel createMenuCard(String title, String description, Color primaryColor, Color hoverColor, ActionListener action) {
2     final Color blue = new Color(33, 150, 243);
3     final Color blueDark = new Color(25, 118, 210);
4     final boolean[] isHovered = {false};
5     JPanel card = new JPanel() {
6         @Override
7         protected void paintComponent(Graphics g) {
8             super.paintComponent(g);
9             Graphics2D g2d = (Graphics2D) g;
10            g2d.setRenderingHint(RenderingHints.KEY_ANTIALIASING, RenderingHints.VALUE_ANTIALIAS_ON);
11
12            GradientPaint gradient = new GradientPaint(
13                0, 0, isHovered[0] ? blueDark : blue,
14                getWidth(), getHeight(), isHovered[0] ? blueDark.darker() : blue.darker()
15            );
16            g2d.setPaint(gradient);
17            g2d.fillRoundRect(0, 0, getWidth(), getHeight(), 18, 18);
18
19            g2d.setColor(new Color(255,255,255,40));
20            g2d.fillRoundRect(0, 0, getWidth(), getHeight()/2, 18, 18);
21        }
22    };
}

```

Berfungsi untuk :

- Membuat komponen kartu menu (JPanel) yang dapat digunakan sebagai item menu visual dalam antarmuka pengguna.
 - Mengatur warna latar belakang panel menggunakan gradasi warna antara dua warna: biru terang dan biru gelap, tergantung apakah mouse sedang hover atau tidak.
 - Mengaktifkan anti-aliasing agar rendering elemen grafik lebih halus dan tajam.
 - Menggambar dua kotak bulat (rounded rectangle), satu untuk latar belakang utama, dan satu lagi sebagai efek bayangan transparan (opacity 40).
 - Menyesuaikan tampilan komponen berdasarkan status isHovered menggunakan logika pewarnaan GradientPaint.
- Mempersiapkan komponen untuk menanggapi event seperti klik melalui parameter ActionListener, meskipun belum diimplementasikan dalam cuplikan ini.

5) createFooterPanel ()

```
1  private JPanel createFooterPanel() {
2      JPanel footerPanel = new JPanel();
3      footerPanel.setLayout(new BoxLayout(footerPanel, BoxLayout.Y_AXIS));
4      footerPanel.setOpaque(false);
5
6      JButton logoutButton = createStyledButton(
7          "Logout",
8          new Color(239, 68, 68),
9          new Color(220, 38, 38)
10     );
11
12     logoutButton.addActionListener(e -> {
13         int result = showStyledConfirmDialog(
14             "Apakah Anda yakin ingin keluar dari sistem?",
15             "Konfirmasi Logout"
16         );
17
18         if (result == JOptionPane.YES_OPTION) {
19             if (clockTimer != null) {
20                 clockTimer.stop();
21             }
22             dispose();
23             new LoginForm().setVisible(true);
24         }
25     });
26
27     JLabel infoLabel = new JLabel("© 2025 Sistem Parkir - Kelompok 5 PBO");
28     infoLabel.setFont(new Font("Segoe UI", Font.ITALIC, 12));
29     infoLabel.setForeground(new Color(107, 114, 128));
30     infoLabel.setAlignmentX(Component.CENTER_ALIGNMENT);
31
32     footerPanel.add(logoutButton);
33     footerPanel.add(Box.createVerticalStrut(20));
34     footerPanel.add(infoLabel);
35
36     return footerPanel;
37 }
```

Berfungsi untuk :

- Membuat panel bagian bawah (footer) dari antarmuka pengguna.
- Menambahkan tombol Logout dengan warna khusus dan tampilan bergaya.
- Menangani aksi tombol Logout dengan menampilkan dialog konfirmasi keluar.
- Jika pengguna memilih “Ya”, maka sistem:
 - Menghentikan timer jika sedang berjalan.
 - Menutup jendela saat ini.
 - Menampilkan kembali form login.
- Menambahkan label informasi berupa copyright aplikasi (© 2025 Sistem Parkir - Kelompok 5 PBO) ke bagian bawah panel.
- Menata komponen dalam panel dengan orientasi vertikal menggunakan BoxLayout.

6) startClock ()

```
1  private void startClock() {
2      clockTimer = new Timer(1000, e -> {
3          String currentTime = LocalDateTime.now().format(
4              DateTimeFormatter.ofPattern("dd-MM-yyyy HH:mm:ss"))
5          );
6          clockLabel.setText("" + currentTime);
7      });
8      clockTimer.start();
9
10     // Set initial time
11     String currentTime = LocalDateTime.now().format(
12         DateTimeFormatter.ofPattern("dd-MM-yyyy HH:mm:ss"))
13     );
14     clockLabel.setText("" + currentTime);
15 }
```

Berfungsi untuk :

- Menampilkan jam digital real-time pada aplikasi.
- Menginisialisasi timer (clockTimer) yang berjalan setiap 1 detik (1000 milidetik).
- Mengambil waktu saat ini menggunakan LocalDateTime.now() dan memformatnya ke pola : dd-MM-yyyy HH:mm:ss, misalnya: 08-06-2025 09:15:30.
- Mengatur teks dari clockLabel agar menampilkan waktu yang sudah diformat.
- Menjalankan clockTimer.start() agar pembaruan waktu terus berjalan otomatis setiap detik.
- Mengatur waktu awal (initial time) saat pertama kali fungsi dipanggil, agar label langsung menampilkan waktu tanpa menunggu 1 detik pertama.

4. Form Masuk (Operator)

Form Masuk adalah form GUI untuk registrasi kendaraan masuk ke area parkir. User/operator mengisi plat nomor, memilih jenis kendaraan, dan waktu masuk otomatis terisi serta terus update. Data akan disimpan ke database saat tombol "Simpan Data" ditekan.

a. Library

```
1 import javax.swing.*;
2 import java.awt.*;
3 import java.awt.event.*;
4 import java.time.LocalDateTime;
5 import java.time.format.DateTimeFormatter;
6 import java.sql.Connection;
7 import java.sql.PreparedStatement;
```

Penjelasan :

- javax.swing.* → Komponen GUI (seperti JButton, JPanel, JFrame)
- java.awt.* dan java.awt.event.* → Tata letak dan event handling (aksi tombol, mouse, dll)
- java.time.LocalDateTime dan DateTimeFormatter → Untuk pengolahan dan format tanggal/waktu
- java.sql.Connection → Mengimpor kelas Connection dari JDBC (java.sql) untuk membuka koneksi ke database
- java.sql.PreparedStatement → Mengimpor kelas PreparedStatement, digunakan untuk menjalankan query SQL yang telah diparameterisasi (aman dari SQL Injection).

b. Konstraktor

```
1 public Masuk(String username, String role) {
2     this.username = username;
3     this.role = role;
4
5     setTitle("Parkir Management - Kendaraan Masuk");
6     setDefaultCloseOperation(EXIT_ON_CLOSE);
7     setExtendedState(JFrame.MAXIMIZED_BOTH);
8     setLocationRelativeTo(null);
9
10    JPanel mainPanel = new JPanel(new GridBagLayout()) {
11        Image bgImage;
12        {
13            try {
14                bgImage = javax.imageio.ImageIO.read(getClass().getResource("/Uas_Pbo/img/parkir2.png"));
15            } catch (Exception e) {
16                bgImage = null;
17            }
18        }
19        @Override
20        protected void paintComponent(Graphics g) {
21            super.paintComponent(g);
22            if (bgImage != null) {
23                g.drawImage(bgImage, 0, 0, getWidth(), getHeight(), this);
24            } else {
25                Graphics2D g2d = (Graphics2D) g;
26                g2d.setColor(new Color(25, 118, 210));
27                g2d.fillRect(0, 0, getWidth(), getHeight());
28            }
29        }
30    };
}
```

Berfungsi untuk :

- Menyimpan Data Pengguna
- Menyimpan informasi pengguna yang sedang login
- Mengatur Tampilan Jendela
- Menentukan bahwa aplikasi akan tertutup saat jendela ditutup.
- Membuat jendela tampil dalam mode fullscreen.
- Membuat Panel Utama dengan Latar Belakang Gambar
- Membuat panel utama (mainPanel) dengan layout GridBagLayout.
- Jika gambar gagal dimuat, latar belakang diwarnai biru sebagai fallback.

c. Desain GUI

1) createMainCard ()

```
1 private JPanel createMainCard() {
2     JPanel card = new JPanel() {
3         @Override
4         protected void paintComponent(Graphics g) {
5             super.paintComponent(g);
6             Graphics2D g2d = (Graphics2D) g;
7             g2d.setRenderingHint(RenderingHints.KEY_ANTIALIASING, RenderingHints.VALUE_ANTIALIAS_ON);
8
9             g2d.setColor(Color.WHITE);
10            g2d.fillRoundRect(0, 0, getWidth()-1, getHeight()-1, 24, 24);
11            g2d.setColor(new Color(33, 150, 243, 180));
12            g2d.setStroke(new BasicStroke(3));
13            g2d.drawRoundRect(0, 0, getWidth()-1, getHeight()-1, 24, 24);
14        }
15    };
}
```

Berfungsi untuk :

- Membuat panel baru dan langsung override method paintComponent agar tampilannya bisa dikustom.
- Menggambar ulang panel secara manual dengan tampilan khusus menggunakan Graphics2D.
- Mengaktifkan anti-aliasing agar gambar terlihat lebih halus (tidak pecah/bergerigi).
- Menentukan warna isi panel (putih).
- Menentukan warna garis tepi panel — warna biru transparan.
- Mengatur ketebalan garis border menjadi 3 piksel.
- Menggambar garis tepi dari panel dengan sudut yang juga membulat.

2) createHeaderPanel ()

```
1  private JPanel createHeaderPanel() {
2      JPanel headerPanel = new JPanel();
3      headerPanel.setLayout(new BoxLayout(headerPanel, BoxLayout.Y_AXIS));
4      headerPanel.setOpaque(false);
5
6      // Icon
7      JLabel iconLabel = new JLabel("🚗");
8      iconLabel.setFont(new Font("Segoe UI Emoji", Font.PLAIN, 48));
9      iconLabel.setAlignmentX(Component.CENTER_ALIGNMENT);
10
11     // Title
12     JLabel titleLabel = new JLabel("Kendaraan Masuk");
13     titleLabel.setFont(new Font("Segoe UI", Font.BOLD, 32));
14     titleLabel.setForeground(new Color(25, 118, 210));
15     titleLabel.setAlignmentX(Component.CENTER_ALIGNMENT);
16
17     // Subtitle
18     JLabel subtitleLabel = new JLabel("Registrasi kendaraan baru");
19     subtitleLabel.setFont(new Font("Segoe UI", Font.PLAIN, 16));
20     subtitleLabel.setForeground(new Color(33, 150, 243));
21     subtitleLabel.setAlignmentX(Component.CENTER_ALIGNMENT);
22
23     headerPanel.add(iconLabel);
24     headerPanel.add(Box.createVerticalStrut(10));
25     headerPanel.add(titleLabel);
26     headerPanel.add(Box.createVerticalStrut(5));
27     headerPanel.add(subtitleLabel);
28
29     return headerPanel;
30 }
```

Berfungsi untuk :

- Membuat panel kontainer vertikal dengan BoxLayout.Y_AXIS agar komponen disusun ke bawah.
- Label yang berisi ikon emoji mobil 🚗, ditampilkan dengan font besar (Segoe UI Emoji, ukuran 48).
- Label teks untuk judul "Kendaraan Masuk", menggunakan font tebal ukuran besar dan warna biru cerah (Color(25, 118, 210)).
- Label teks untuk subjudul "Registrasi kendaraan baru", menggunakan font ukuran lebih kecil dan warna biru muda (Color(33, 150, 243)).
- Digunakan untuk memberi jarak vertikal antar elemen, membuat tata letak lebih rapi.

3) createFormPanel ()

```
1  private JPanel createFormPanel() {
2      JPanel formPanel = new JPanel();
3      formPanel.setLayout(new BoxLayout(formPanel, BoxLayout.Y_AXIS));
4      formPanel.setOpaque(false);
5      formPanel.setAlignmentX(Component.CENTER_ALIGNMENT);
6      formPanel.setMaximumSize(new Dimension(400, 350));
7
8      // Plat field
9      JPanel platPanel = createInputField("Plat Nomor");
10     platPanel.setAlignmentX(Component.CENTER_ALIGNMENT);
11     tfPlat = createStyledTextField();
12     tfPlat.setAlignmentX(Component.CENTER_ALIGNMENT);
13     platPanel.add(Box.createVerticalStrut(8));
14     platPanel.add(tfPlat);
15
16      // Jenis field
17      JPanel jenisPanel = createInputField("Jenis Kendaraan");
18      jenisPanel.setAlignmentX(Component.CENTER_ALIGNMENT);
19      cbJenis = createStyledComboBox();
20      cbJenis.setAlignmentX(Component.CENTER_ALIGNMENT);
21      jenisPanel.add(Box.createVerticalStrut(8));
22      jenisPanel.add(cbJenis);
23
24      // Time field
25      JPanel timePanel = createInputField("Waktu Masuk");
26      timePanel.setAlignmentX(Component.CENTER_ALIGNMENT);
27      lblJamMasuk = createTimeLabel();
28      lblJamMasuk.setAlignmentX(Component.CENTER_ALIGNMENT);
29      timePanel.add(Box.createVerticalStrut(8));
30      timePanel.add(lblJamMasuk);
31
32      formPanel.add(platPanel);
33      formPanel.add(Box.createVerticalStrut(20));
34      formPanel.add(jenisPanel);
35      formPanel.add(Box.createVerticalStrut(20));
36      formPanel.add(timePanel);
37
38      return formPanel;
39  }
```

Berfungsi untuk :

- Membuat panel utama formPanel yang akan berisi seluruh komponen input.
- Menetapkan ukuran maksimum panel formPanel sebesar 400x350 piksel.
- Menambahkan TextField ke panel platPanel.
- Membuat panel dengan label "Jenis Kendaraan".
- Membuat ComboBox dengan gaya khusus untuk memilih jenis kendaraan.
- Menambahkan jenisPanel ke dalam formPanel.
- Menambahkan label waktu ke dalam timePanel.
- Menambahkan semua panel ke formPanel diselingi jarak vertikal.

4) createInputField ()

```
1  private JPanel createInputField(String labelText) {  
2      JPanel panel = new JPanel();  
3      panel.setLayout(new BoxLayout(panel, BoxLayout.Y_AXIS));  
4      panel.setOpaque(false);  
5      panel.setAlignmentX(Component.CENTER_ALIGNMENT);  
6  
7      JLabel label = new JLabel(labelText);  
8      label.setFont(new Font("Segoe UI", Font.BOLD, 16));  
9      label.setForeground(new Color(25, 118, 210)); // biru  
10     label.setAlignmentX(Component.CENTER_ALIGNMENT);  
11  
12     panel.add(label);  
13  
14 }
```

Berfungsi untuk :

- Method createInputField membuat JPanel yang berisi satu JLabel, dengan teks label yang diterima sebagai parameter (labelText).
- Membuat panel baru sebagai wadah dari label.
- Menyusun komponen dalam panel secara vertikal (atas ke bawah).
- Membuat panel transparan, tidak menampilkan latar belakang.
- Menyetel agar panel diratakan di tengah secara horizontal.
- Membuat JLabel dengan teks sesuai input dari parameter labelText.
- Mengubah warna teks label menjadi biru.
- Menengahkan label dalam panel secara horizontal.
- Menambahkan label ke dalam panel.

5) createStyledComboBox ()

```
1 private JComboBox<String> createStyledComboBox() {
2     String[] options = {"Motor", "Mobil"};
3     JComboBox<String> comboBox = new JComboBox<>(options);
4     comboBox.setFont(new Font("Segoe UI", Font.PLAIN, 18));
5     comboBox.setMaximumSize(new Dimension(Integer.MAX_VALUE, 45));
6     comboBox.setPreferredSize(new Dimension(350, 45));
7     comboBox.setBackground(new Color(245, 250, 255));
8     comboBox.setForeground(new Color(25, 118, 210));
9     comboBox.setBorder(BorderFactory.createCompoundBorder(
10         BorderFactory.createLineBorder(new Color(33, 150, 243), 1, true),
11         BorderFactory.createEmptyBorder(8, 12, 8, 12)
12     ));
13
14     return comboBox;
15 }
```

Berfungsi untuk :

- Membuat JComboBox: Inisialisasi sebuah JComboBox yang akan menampilkan opsi "Motor" dan "Mobil".
- Mengatur Font: Mengatur font teks di dalam JComboBox
- Mengatur Ukuran Maksimum dan Preferensi: Menentukan ukuran maksimum dan ukuran preferensi untuk kotak kombo.
- Mengatur Warna Latar Belakang: Mengatur warna latar belakang kotak kombo menjadi warna biru muda.
- Mengatur Warna Teks (Foreground): Mengatur warna teks di dalam kotak kombo menjadi warna biru.
- Mengatur Border (Batas): Membuat batas kustom untuk kotak kombo.

6) createTimeLabel ()

```
1 private JLabel createTimeLabel() {
2     JLabel timeLabel = new JLabel();
3     timeLabel.setFont(new Font("Segoe UI", Font.BOLD, 18));
4     timeLabel.setForeground(new Color(25, 118, 210));
5     timeLabel.setOpaque(true);
6     timeLabel.setBackground(new Color(232, 245, 253));
7     timeLabel.setBorder(BorderFactory.createCompoundBorder(
8         BorderFactory.createLineBorder(new Color(33, 150, 243), 1, true),
9         BorderFactory.createEmptyBorder(12, 16, 12, 16)
10    ));
11    timeLabel.setMaximumSize(new Dimension(Integer.MAX_VALUE, 45));
12    timeLabel.setPreferredSize(new Dimension(350, 45));
13
14    return timeLabel;
15 }
```

Berfungsi untuk :

- Membuat label (JLabel) dengan font tebal "Segoe UI" ukuran 18.
- Warna teks: biru muda, latar belakang: biru sangat terang.
- Menampilkan latar belakang (setOpaque(true)).
- Memberi border kombinasi garis biru dan padding.
- Menentukan ukuran maksimal dan preferensi label.
- Mengembalikan label tersebut.

7) createButtonPanel ()

```
1 private JPanel createButtonPanel() {
2     JPanel buttonPanel = new JPanel();
3     buttonPanel.setLayout(new BoxLayout(buttonPanel, BoxLayout.Y_AXIS));
4     buttonPanel.setOpaque(false);
5     buttonPanel.setAlignmentX(Component.CENTER_ALIGNMENT);
6     buttonPanel.setMaximumSize(new Dimension(400, 120));
7
8     // Save button
9     JButton btnSimpan = createStyledButton("Simpan Data", new Color(16, 185, 129), new Color(5, 150, 105));
10    btnSimpan.setAlignmentX(Component.CENTER_ALIGNMENT);
11    btnSimpan.addActionListener(e -> simpanMasuk());
12
13    // Back button
14    JButton btnBack = createStyledButton("Kembali", new Color(107, 114, 128), new Color(75, 85, 99));
15    btnBack.setAlignmentX(Component.CENTER_ALIGNMENT);
16    btnBack.addActionListener(e -> {
17        timeUpdater.stop();
18        new OptionOP(username, role).setVisible(true);
19        dispose();
20    });
21
22    buttonPanel.add(btnSimpan);
23    buttonPanel.add(Box.createVerticalStrut(15));
24    buttonPanel.add(btnBack);
25
26    return buttonPanel;
27 }
```

Berfungsi untuk :

- Potongan kode ini membuat panel tombol (JPanel) secara vertikal dan menambahkan dua tombol:
- Tombol "Simpan Data", warna tombol dikustomisasi. dan akan menjalankan aksi simpanWaktu() saat diklik.
- Tombol "Kembali", warna tombol dikustomisasi dan saat diklik, menghentikan timer, membuka opsi baru (new Option(...)), dan menutup jendela saat ini (dispose()).

5. Form Keluar (Operator)

Form Keluar adalah form GUI untuk memproses kendaraan yang akan keluar dari area parkir. User/operator dapat mencari kendaraan berdasarkan plat nomor, melihat info waktu keluar, durasi, dan tarif, lalu memproses keluar (update database & cetak struk).

a. Library

```
1 import javax.swing.*;
2 import java.awt.*;
3 import java.awt.event.*;
4 import java.time.LocalDateTime;
5 import java.time.format.DateTimeFormatter;
6 import java.sql.*;
7 import javax.swing.table.DefaultTableModel;
```

Untuk mengimpor pustaka-pustaka Java yang dibutuhkan:

- javax.swing.* → Komponen GUI (seperti JButton, JPanel, JFrame)
- java.awt.* dan java.awt.event.* → Tata letak dan event handling (aksi tombol, mouse, dll)
- java.time.LocalDateTime dan DateTimeFormatter → Untuk pengolahan dan format tanggal/waktu
- java.sql.* → Koneksi dan pengolahan database
- javax.swing.table.DefaultTableModel → Model data untuk tabel di Swing

b. Konstruktor

```
1 public Keluar(String username, String role) {
2     this.username = username;
3     this.role = role;
4
5     setTitle("Parkir Management - Kendaraan Keluar");
6     setDefaultCloseOperation(EXIT_ON_CLOSE);
7     setExtendedState(JFrame.MAXIMIZED_BOTH);
8     setLocationRelativeTo(null);
9
10    JPanel mainPanel = new JPanel(new GridBagLayout()) {
11        Image bgImage;
12        {
13            try {
14                bgImage = javax.imageio.ImageIO.read(getClass().getResource("/Uas_Pbo/img/parkir2.png"));
15            } catch (Exception e) {
16                bgImage = null;
17            }
18        }
19        @Override
20        protected void paintComponent(Graphics g) {
21            super.paintComponent(g);
22            if (bgImage != null) {
23                g.drawImage(bgImage, 0, 0, getWidth(), getHeight(), this);
24            } else {
25                // fallback biru
26                Graphics2D g2d = (Graphics2D) g;
27                g2d.setColor(new Color(25, 118, 210));
28                g2d.fillRect(0, 0, getWidth(), getHeight());
29            }
30        }
31    };
32}
```

Berfungsi untuk :

- Mengatur tampilan awal frame "Parkir Management - Kendaraan Keluar".
- Menerima parameter username dan role.
- Mengatur frame agar fullscreen dan menutup saat diklik close.
- Membuat panel utama (mainPanel) dengan background gambar (parkir2.png).
- Jika gambar gagal dimuat, digunakan warna biru sebagai fallback.
- Menggunakan paintComponent untuk menggambar background ke panel.

c. Desain GUI

1) createMainCard ()

```
1 private JPanel createMainCard() {
2     JPanel card = new JPanel() {
3         @Override
4         protected void paintComponent(Graphics g) {
5             super.paintComponent(g);
6             Graphics2D g2d = (Graphics2D) g;
7             g2d.setRenderingHint(RenderingHints.KEY_ANTIALIASING, RenderingHints.VALUE_ANTIALIAS_ON);
8             g2d.setColor(Color.WHITE);
9             g2d.fillRoundRect(0, 0, getWidth()-1, getHeight()-1, 24, 24);
10            g2d.setColor(new Color(33, 150, 243, 180));
11            g2d.setStroke(new BasicStroke(3));
12            g2d.drawRoundRect(0, 0, getWidth()-1, getHeight()-1, 24, 24);
13        }
14    };
15}
```

Berfungsi untuk :

- Membuat panel utama (card) dengan tampilan custom.
- Menggambar latar putih berbentuk kotak rounded (sudut melengkung).
- Menambahkan border luar berwarna biru (RGB 33, 150, 243) dengan ketebalan 3 piksel.
- Mengaktifkan anti-aliasing agar gambar tampak halus.
- Mengembalikan panel card sebagai hasil.

2) createHeaderPanel ()

```
1 private JPanel createHeaderPanel() {  
2     JPanel headerPanel = new JPanel();  
3     headerPanel.setLayout(new BoxLayout(headerPanel, BoxLayout.Y_AXIS));  
4     headerPanel.setOpaque(false);  
5  
6     // Icon  
7     JLabel iconLabel = new JLabel("🚗");  
8     iconLabel.setFont(new Font("Segoe UI Emoji", Font.PLAIN, 48));  
9     iconLabel.setAlignmentX(Component.CENTER_ALIGNMENT);  
10  
11    // Title  
12    JLabel titleLabel = new JLabel("Kendaraan Keluar");  
13    titleLabel.setFont(new Font("Segoe UI", Font.BOLD, 32));  
14    titleLabel.setForeground(new Color(25, 118, 210));  
15    titleLabel.setAlignmentX(Component.CENTER_ALIGNMENT);  
16  
17    // Subtitle  
18    JLabel subtitleLabel = new JLabel("Proses kendaraan yang akan keluar");  
19    subtitleLabel.setFont(new Font("Segoe UI", Font.PLAIN, 16));  
20    subtitleLabel.setForeground(new Color(33, 150, 243));  
21    subtitleLabel.setAlignmentX(Component.CENTER_ALIGNMENT);  
22  
23    headerPanel.add(iconLabel);  
24    headerPanel.add(Box.createVerticalStrut(10));  
25    headerPanel.add(titleLabel);  
26    headerPanel.add(Box.createVerticalStrut(5));  
27    headerPanel.add(subtitleLabel);  
28  
29    return headerPanel;  
30}
```

Berfungsi untuk :

- Membuat panel header vertikal (atas) dengan tata letak BoxLayout.
- Menampilkan icon emoji, judul “Kendaraan Keluar”, dan subjudul proses kendaraan.
- Mengatur font, warna, dan perataan tengah (CENTER_ALIGNMENT).
- Menambahkan jarak antar elemen menggunakan Box.createVerticalStrut().
- Mengembalikan panel header sebagai hasil.

3) createSearchPanel ()

```
1  private JPanel createSearchPanel() {
2      JPanel searchPanel = new JPanel();
3      searchPanel.setLayout(new BoxLayout(searchPanel, BoxLayout.Y_AXIS));
4      searchPanel.setOpaque(false);
5
6      // Search label
7      JLabel searchLabel = new JLabel("Masukkan Plat Nomor");
8      searchLabel.setFont(new Font("Segoe UI", Font.BOLD, 16));
9      searchLabel.setForeground(new Color(55, 65, 81));
10     searchLabel.setAlignmentX(Component.LEFT_ALIGNMENT);
11
12     // Search field with button
13     JPanel searchFieldPanel = new JPanel(new BorderLayout(10, 0));
14     searchFieldPanel.setOpaque(false);
15     searchFieldPanel.setMaximumSize(new Dimension(Integer.MAX_VALUE, 45));
16
17     tfPlat = createStyledTextField();
18     JButton btnCari = createSearchButton();
19
20     searchFieldPanel.add(tfPlat, BorderLayout.CENTER);
21     searchFieldPanel.add(btnCari, BorderLayout.EAST);
22
23     searchPanel.add(searchLabel);
24     searchPanel.add(Box.createVerticalStrut(8));
25     searchPanel.add(searchFieldPanel);
26
27
28     tfPlat.addActionListener(e -> cariKendaraan());
29
30     return searchPanel;
31 }
```

Berfungsi untuk :

- Membuat panel pencarian plat nomor dengan tata letak vertikal (BoxLayout).
- Menampilkan label instruksi “Masukkan Plat Nomor”.
- Membuat panel input (text field) dan tombol cari di satu baris (BorderLayout).
- Mengatur gaya font, warna teks, dan ukuran panel input.
- Menambahkan aksi pencarian saat Enter ditekan di text field.
- Mengembalikan searchPanel sebagai hasil.

4) createSearchButton ()

```
1  private JButton createSearchButton() {
2      JButton button = new JButton("Cari") {
3          @Override
4          protected void paintComponent(Graphics g) {
5              Graphics2D gd = (Graphics2D) g;
6              gd.setRenderingHint(RenderingHints.KEY_ANTIALIASING, RenderingHints.VALUE_ANTIALIAS_ON);
7
8              Color bgColor = getModel().isPressed() ? new Color(33, 150, 243) : new Color(25, 118, 210);
9              gd.setColor(bgColor);
10             gd.fillRoundRect(0, 0, getWidth(), getHeight(), 12, 12);
11
12             super.paintComponent(g);
13         }
14     };
15 }
```

Berfungsi untuk :

- Membuat tombol “Cari” dengan tampilan khusus.
- Menggunakan Graphics2D untuk menggambar tombol dengan pinggiran membulat (rounded rectangle).
- Mengatur warna tombol: biru muda saat ditekan, biru gelap saat normal.
- Mengaktifkan anti-aliasing agar tampilan lebih halus.
- Mengembalikan JButton sebagai hasil.

5) createInfoPanel ()

```
1 private JPanel createInfoPanel() {  
2     JPanel panel = new JPanel();  
3     panel.setLayout(new BoxLayout(panel, BoxLayout.Y_AXIS));  
4     panel.setOpaque(false);  
5     panel.setBorder(BorderFactory.createTitledBorder(  
6         BorderFactory.createLineBorder(new Color(33, 150, 243), 2, true),  
7         "Informasi Parkir",  
8         0, 0,  
9         new Font("Segoe UI", Font.BOLD, 16),  
10        new Color(25, 118, 210)  
11    ));  
12    panel.setMaximumSize(new Dimension(Integer.MAX_VALUE, 250));  
13  
14    // Initialize labels  
15    lblJamKeluar = new JLabel("-");  
16    lblJamKeluar.setFont(new Font("Segoe UI", Font.BOLD, 16));  
17    lblJamKeluar.setForeground(new Color(25, 118, 210));  
18    lblJamKeluar.setAlignmentX(Component.LEFT_ALIGNMENT);  
19  
20    lblDurasi = new JLabel("-");  
21    lblDurasi.setFont(new Font("Segoe UI", Font.BOLD, 16));  
22    lblDurasi.setForeground(new Color(25, 118, 210));  
23    lblDurasi.setAlignmentX(Component.LEFT_ALIGNMENT);  
24  
25    lblTarif = new JLabel("-");  
26    lblTarif.setFont(new Font("Segoe UI", Font.BOLD, 16));  
27    lblTarif.setForeground(new Color(25, 118, 210));  
28    lblTarif.setAlignmentX(Component.LEFT_ALIGNMENT);  
29  
30    // Label judul  
31    JLabel lblJamKeluarTitle = new JLabel("Waktu Keluar");  
32    lblJamKeluarTitle.setFont(new Font("Segoe UI", Font.BOLD, 14));  
33    lblJamKeluarTitle.setForeground(new Color(33, 150, 243));  
34    JLabel lblDurasiTitle = new JLabel("Durasi Parkir");  
35    lblDurasiTitle.setFont(new Font("Segoe UI", Font.BOLD, 14));  
36    lblDurasiTitle.setForeground(new Color(33, 150, 243));  
37    JLabel lblTariffTitle = new JLabel("Total Tarif");  
38    lblTariffTitle.setFont(new Font("Segoe UI", Font.BOLD, 14));  
39    lblTariffTitle.setForeground(new Color(33, 150, 243));  
40  
41    // Panel baris info  
42    JPanel rowJamKeluar = new JPanel(new BorderLayout());  
43    rowJamKeluar.setOpaque(false);  
44    rowJamKeluar.add(lblJamKeluarTitle, BorderLayout.WEST);  
45    rowJamKeluar.add(lblJamKeluar, BorderLayout.EAST);  
46  
47    JPanel rowDurasi = new JPanel(new BorderLayout());  
48    rowDurasi.setOpaque(false);  
49    rowDurasi.add(lblDurasiTitle, BorderLayout.WEST);  
50    rowDurasi.add(lblDurasi, BorderLayout.EAST);  
51  
52    JPanel rowTarif = new JPanel(new BorderLayout());  
53    rowTarif.setOpaque(false);  
54    rowTarif.add(lblTariffTitle, BorderLayout.WEST);  
55    rowTarif.add(lblTarif, BorderLayout.EAST);  
56  
57    panel.add(Box.createVerticalStrut(15));  
58    panel.add(rowJamKeluar);  
59    panel.add(Box.createVerticalStrut(15));  
60    panel.add(rowDurasi);  
61    panel.add(Box.createVerticalStrut(15));  
62    panel.add(rowTarif);  
63    panel.add(Box.createVerticalStrut(15));  
64  
65    return panel;  
66}
```

Berfungsi untuk :

- Membuat panel informasi kendaraan yang berisi label-label seperti: Jam keluar, Durasi parkir, Tarif parkir.
- Menggunakan BoxLayout untuk tata letak vertikal.
- Mengatur tampilan judul panel dengan border dan warna.
- Membuat label untuk menampilkan informasi parkir.
- Membagi panel menjadi baris-baris informasi (rowJamKeluar, rowDurasi, rowTarif) menggunakan BorderLayout.
- Menambahkan jarak antar elemen menggunakan Box.createVerticalStrut.

6) createButtonPanel ()

```
1 private JPanel createButtonPanel() {  
2     JPanel buttonPanel = new JPanel();  
3     buttonPanel.setLayout(new BoxLayout(buttonPanel, BoxLayout.Y_AXIS));  
4     buttonPanel.setOpaque(false);  
5  
6     // Proses button  
7     JButton btnSimpan = createStyledButton("Proses Keluar", new Color(220, 38, 38), new Color(185, 28, 28));  
8     btnSimpan.setMaximumSize(new Dimension(Integer.MAX_VALUE, 50));  
9     btnSimpan.setPreferredSize(new Dimension(350, 50));  
10    btnSimpan.setAlignmentX(Component.CENTER_ALIGNMENT);  
11    btnSimpan.addActionListener(e -> simpanKeluar());  
12  
13    // Back button  
14    JButton btnBack = createStyledButton("Kembali", new Color(107, 114, 128), new Color(75, 85, 99));  
15    btnBack.setMaximumSize(new Dimension(Integer.MAX_VALUE, 50));  
16    btnBack.setPreferredSize(new Dimension(350, 50));  
17    btnBack.setAlignmentX(Component.CENTER_ALIGNMENT);  
18    btnBack.addActionListener(e -> {  
19        new OptionOP(username, role).setVisible(true);  
20        dispose();  
21    });  
22  
23    buttonPanel.add(btnSimpan);  
24    buttonPanel.add(Box.createVerticalStrut(15));  
25    buttonPanel.add(btnBack);  
26  
27    return buttonPanel;  
28}
```

Berfungsi untuk :

- Membuat panel berisi dua tombol: btnSimpan: Tombol "Proses Keluar" untuk menyimpan data parkir keluar. dan btnBack: Tombol "Kembali" untuk kembali ke halaman sebelumnya.
- Menggunakan BoxLayout vertikal untuk menyusun tombol secara atas-bawah.
- btnSimpan diberi ukuran tetap, rata tengah, dan aksi saat diklik akan menjalankan simpanKeluar().
- btnBack juga dirancang serupa, tetapi akan membuka jendela baru (new Option()) dan menutup jendela sekarang (dispose()).
- Menambahkan jarak vertikal di antara kedua tombol.

7) createTablePanel ()

```
1  private JScrollPane createTablePanel() {  
2      // Kolom: Plat, Jenis, Tanggal Masuk  
3      tableModel = new DefaultTableModel(  
4          new String[][]{"Plat Nomor", "Jenis", "Tanggal Masuk"}, 0  
5      ) {  
6          @Override  
7          public boolean isCellEditable(int row, int column) {  
8              return false;  
9          }  
10     };
```

Berfungsi untuk :

- Membuat tabel data kendaraan masuk dengan tiga kolom: Plat Nomor, Jenis, Tanggal Masuk.
- Menggunakan DefaultTableModel untuk mendefinisikan struktur kolom.
- Override method isCellEditable() agar seluruh sel tabel tidak bisa diedit oleh pengguna.
- (Gambar belum menampilkan JTable atau JScrollPane sepenuhnya, tapi kemungkinan tabel ini akan dibungkus JScrollPane untuk scroll otomatis.)

2.5 Pengujian

Output/Tampilan dari kode Program :

1. Login Form



2. Dashboard Admin

- Form Kendaraan Masuk

Form Kendaraan Masuk

Plat Nomor:

Jenis Kendaraan: Motor

Tanggal Masuk: Auto-generated

SIMPAN MASUK

Plat Nomor	Jenis	Tanggal Masuk	Tanggal Keluar	Durasi Parkir	Tarif
B 1234 POI	Motor	22/05/2025 23:39:09	22/05/2025 23:44:56	5	165000
F 6754 KLU	Motor	22/05/2025 23:39:21	22/05/2025 23:58:55	19	627
B 1234 GHU	Motor	22/05/2025 23:45:35	23/05/2025 00:05:06	19	627
G 4567 LKJ	Mobil	22/05/2025 23:45:51	23/05/2025 22:56:19	1390	115370
Z 3987 SDF	Mobil	22/05/2025 23:46:04	23/05/2025 23:02:38	1396	80000
H 8765 MNB	Motor	22/05/2025 23:50:13	24/05/2025 10:20:34	2070	60000
B 6589 LKU	Mobil	22/05/2025 23:57:45	24/05/2025 14:29:24	2311	120000
A 4729 NGY	Motor	23/05/2025 00:15:49	24/05/2025 14:32:24	2296	40000
F 3421 FFF	Mobil	23/05/2025 00:16:02	24/05/2025 16:57:04	2441	80000
Q 3756 HHH	Mobil	23/05/2025 00:16:13	24/05/2025 17:55:14	2499	80000
A 7222 UVII	Motor	22/05/2025 23:46:27	23/05/2025 21:26:20	4150	60000

- Form Kendaraan Keluar

Form Kendaraan Keluar

Plat Nomor:

Tanggal Keluar: Auto-generated

Durasi Parkir: -

Tarif Parkir: -

SIMPAN KELUAR

Plat Nomor	Jenis	Tanggal Masuk	Tanggal Keluar	Durasi Parkir	Tarif
B 1234 POI	Motor	22/05/2025 23:39:09	22/05/2025 23:44:56	5	165000
F 6754 KLU	Motor	22/05/2025 23:39:21	22/05/2025 23:58:55	19	627
B 1234 GHU	Motor	22/05/2025 23:45:35	23/05/2025 00:05:06	19	627
G 4567 LKJ	Mobil	22/05/2025 23:45:51	23/05/2025 22:56:19	1390	115370
Z 3987 SDF	Mobil	22/05/2025 23:46:04	23/05/2025 23:02:38	1396	80000
H 8765 MNB	Motor	22/05/2025 23:50:13	24/05/2025 10:20:34	2070	60000
B 6589 LKU	Mobil	22/05/2025 23:57:45	24/05/2025 14:29:24	2311	120000
A 4729 NGY	Motor	23/05/2025 00:15:49	24/05/2025 14:32:24	2296	40000
F 3421 FFF	Mobil	23/05/2025 00:16:02	24/05/2025 16:57:04	2441	80000
Q 3756 HHH	Mobil	23/05/2025 00:16:13	24/05/2025 17:55:14	2499	80000
A 7222 UVII	Motor	22/05/2025 23:46:27	23/05/2025 21:26:20	4150	60000

- Menu

Menu

Cetak Struk

Hapus Data

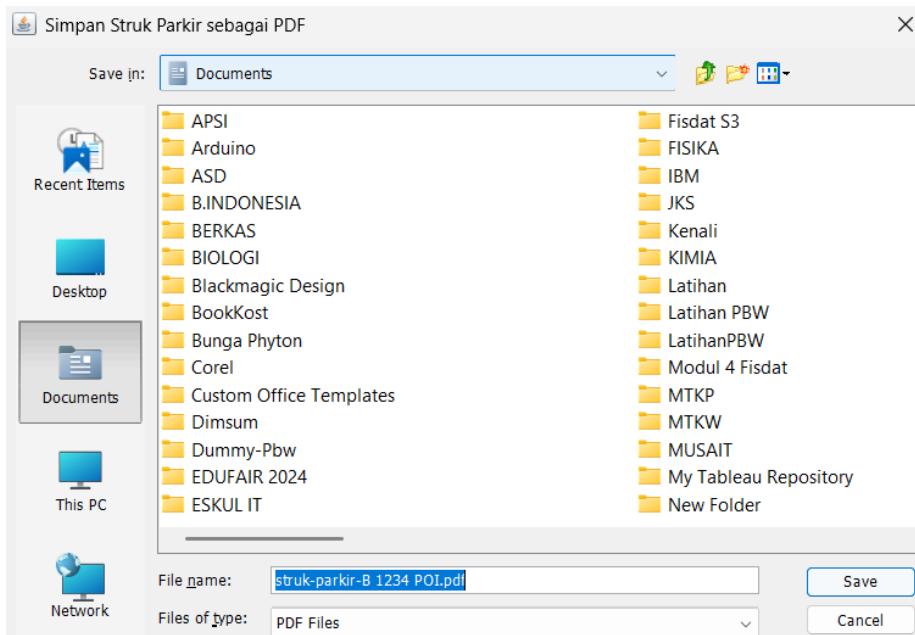
Edit Data Parkir

Logout

- Cetak Struk



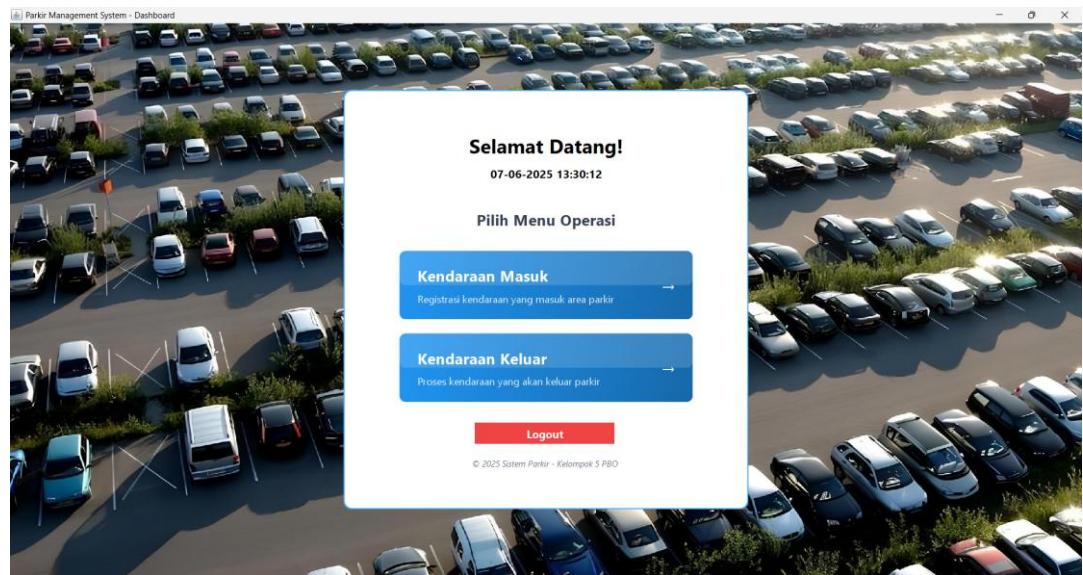
- Tampilan Pilih Folder Untuk Menyimpan PDF



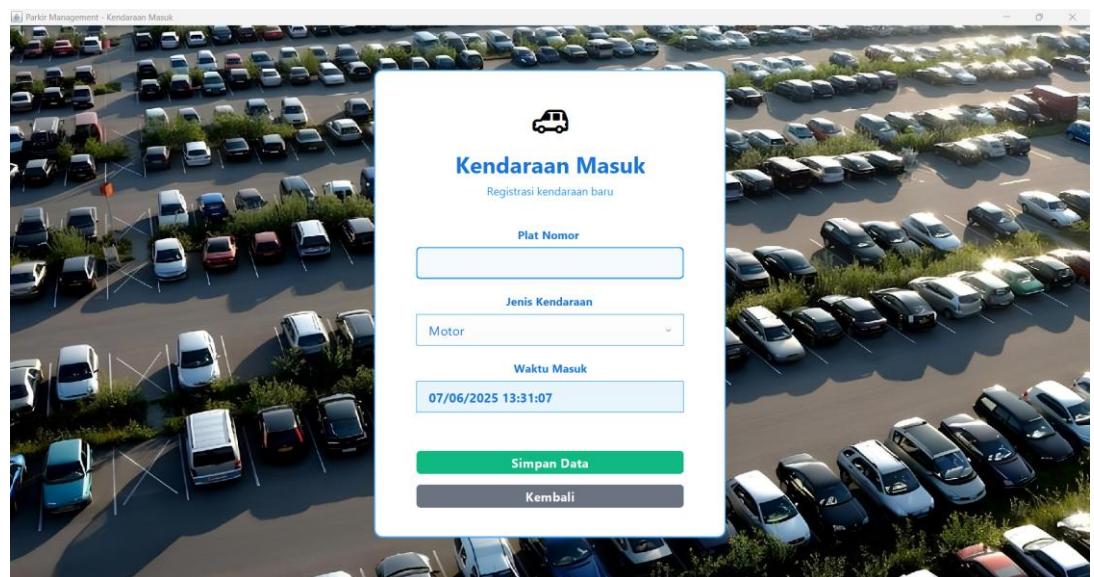
- Tampilan Struk Parkir dalam Format PDF

[Tampilan-PDF-Struk-Parkir.pdf](#)

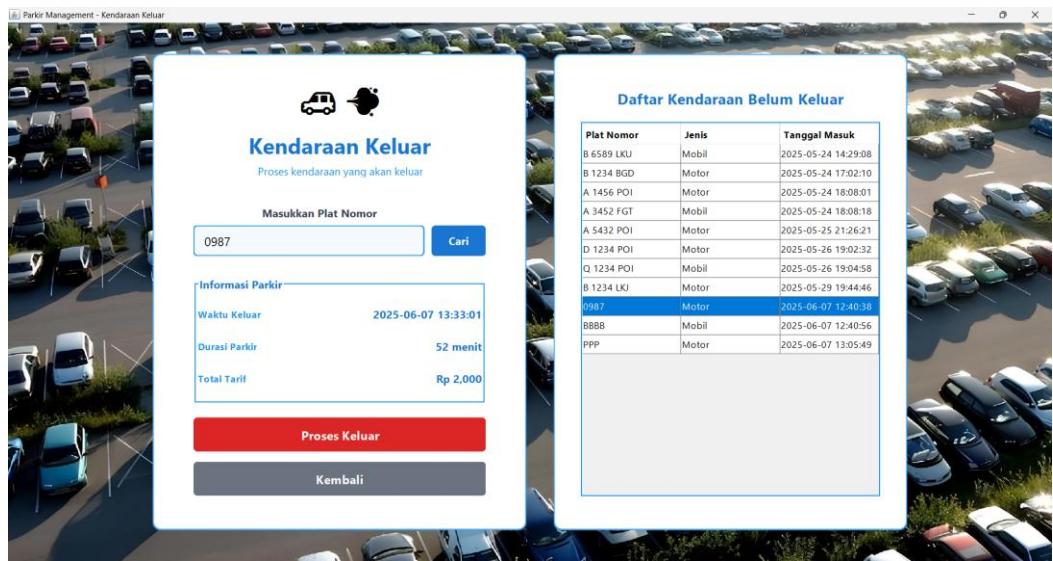
3. Dashboard Operator



4. Form Kendaraan Masuk (Operator)



5. Form Kendaraan Keluar (Operator)



BAB III

PENUTUP

3.1 Kesimpulan

Dari hasil penggeraan dan pengujian aplikasi sistem parkir ini, dapat kami simpulkan bahwa program yang kami buat sudah berjalan sesuai dengan rencana awal. Aplikasi ini dibuat menggunakan bahasa Java dengan tampilan GUI, dan memiliki berbagai fitur yang membantu pencatatan kendaraan, seperti login dengan dua peran (admin dan operator), pencatatan kendaraan masuk dan keluar, hitung tarif otomatis, sampai cetak struk parkir. Selama proses pengembangan, kami juga menerapkan konsep dasar OOP seperti enkapsulasi, inheritance, dan polymorphism, yang membuat struktur program jadi lebih rapi dan mudah dikembangkan. Selain itu, kami juga menggunakan diagram UML seperti use case, class diagram, activity, dan sequence untuk merancang sistem secara terstruktur. Setelah diuji, aplikasi ini berjalan dengan baik dan sesuai dengan kebutuhan yang telah ditentukan, serta bisa digunakan untuk mempermudah pengelolaan data parkir secara digital.

3.2 Saran

Walaupun aplikasi ini sudah cukup lengkap dan bisa digunakan, kami merasa masih ada beberapa hal yang bisa ditingkatkan. Misalnya, sistem ini masih bersifat lokal, jadi akan lebih baik jika bisa disimpan secara online agar lebih aman dan bisa diakses dari tempat lain. Kami juga menyarankan untuk menambahkan fitur seperti scan barcode atau QR code supaya proses masuk dan keluar kendaraan lebih cepat. Selain itu, tampilan antarmuka juga bisa dibuat lebih modern dan responsif agar nyaman dipakai di berbagai perangkat. Ke depannya, fitur tambahan seperti laporan data parkir dan statistik harian juga akan sangat membantu pihak pengelola. Kami berharap aplikasi ini bisa terus dikembangkan dan jadi solusi nyata dalam sistem parkir di lingkungan kampus atau tempat umum lainnya.