

Dokumentasi Project OOP

Anselmus Levin Lukman – 2802446352

Melvin Hansen Zulkarnain – 2802444561

Samuel Teguh – 2802444593

Deskripsi Singkat

Pada project ini, kami membuat sebuah program aplikasi game simulasi pabrik dan showroom mobil. Aplikasi ini dirancang untuk mensimulasikan proses produksi mobil, penyimpanan inventaris mobil, pencarian, penjualan, hingga penyimpanan data ke dalam file eksternal. Aplikasi ini dilengkapi dengan GUI menggunakan Javawing untuk memberikan tampilan yang simpel dan minimalis sehingga mempermudah user dalam mengoperasikannya. Aplikasi ini menggunakan konsep OOP encapsulation, inheritance, polymorphism, dan abstraction.

Fitur

1. Create Car

Memproduksi dan menambah mobil baru dengan memilih pilihan tipe mobil (Sedan, SUV, Hatchback) dan memberi nama mobil. Setiap mobil akan mendapat hp, warna, VIN, dan plat nomor yang ditentukan secara otomatis dan acak dengan fitur `math.random` di Java.

2. Show Car

Menampilkan seluruh stok mobil di gudang dalam bentuk tabel lengkap yang menampilkan semua informasi mobil yang telah disimpan.

3. Search Car

Mencari mobil spesifik berdasarkan nama.

4. Sell Car

Menghapus data mobil atau simulasi penjualan dengan memilih mobil yang ingin dijual.

5. Save Data

Menyimpan seluruh data inventaris ke file teks (`cars.txt`) agar data tidak hilang dan dapat dibaca.

Cara Menjalankan

Prasyarat:

- Pastikan komputer sudah terinstall Java Development Kit (JDK) minimal versi 8.
- Disarankan menggunakan IDE seperti Eclipse, IntelliJ IDEA, atau VS Code.

Langkah-Langkah:

1. Clone atau Download proyek ini.
2. Buka folder proyek di IDE pilihanmu.
3. Cari file utama di lokasi: src/showroom/ui/MainMenu.java.
4. Klik Run pada file MainMenu.java.

Output File:

Setelah melakukan penyimpanan data (tombol Save), file cars.txt akan otomatis muncul di folder root proyek.

Daftar Class dan Fungsinya

1. Package showroom.models (Data Logic)

Class:

Car.java: Abstract Class induk yang mendefinisikan atribut dasar mobil (nama, hp, vin, kapasitas, plat nomor, warna).

SUV.java, Sedan.java, Hatchback.java: Sub-class yang mewarisi Car dengan spesifikasi default masing-masing.

2. Package showroom.ui (Tampilan Utama)

Class:

MainMenu.java: Entry point aplikasi dan menampilkan dashboard utama berisi navigasi tombol menu.

3. Package showroom.ui.dialogs (Popup Jendela)

Class:

CreateCarDialog.java: Form input untuk membuat mobil baru dengan validasi dan custom toggle button.

ShowCarDialog.java: Menampilkan tabel daftar mobil yang ada di inventaris.

SearchCarDialog.java: Fitur pencarian mobil menggunakan Java Stream API (filter by name).

SellCarDialog.java: Menampilkan tabel untuk memilih dan menghapus mobil berdasarkan VIN.

4. Package showroom.ui.components (Desain Custom)

Class:

BackgroundPanel.java: Panel dengan tampilan Radial Gradient untuk background efek "Glow" tanpa gambar eksternal.

RoundedPanel.java: Panel transparan dengan sudut membulat sebagai wadah menu utama.

5. Package showroom.utils (Utilitas)

Class:

AppTheme.java: Pusat konfigurasi warna (Palet Dark Mode, Cyan Accent) dan Font agar desain konsisten.

FileManager.java: Menangani operasi File I/O dengan menyimpan ArrayList ke file .txt.

Penjelasan Konsep OOP

1. Encapsulation

Seluruh atribut data pada class Car bersifat private untuk melindungi data dari akses langsung yang tidak sah. Data hanya dapat diakses atau diubah melalui method public (Getter).

Contohnya private String vin; diakses melalui public String getVin().

2. Inheritance

Mencegah duplikasi kode dengan mewariskan sifat umum.

Implementasinya adalah pada Class SUV, Sedan, dan Hatchback menggunakan kata kunci extends Car. Mereka otomatis memiliki atribut nama, hp, dan vin tanpa perlu menuliskannya ulang.

3. Polymorphism

Kemampuan objek untuk memiliki banyak bentuk.

Implementasinya adalah menggunakan koleksi ArrayList<Car>. List ini bertipe induk (Car), namun dapat menyimpan berbagai bentuk objek anak (new SUV(), new Sedan()).

Saat method c.getType() dipanggil, hasilnya akan berbeda tergantung objek aslinya.

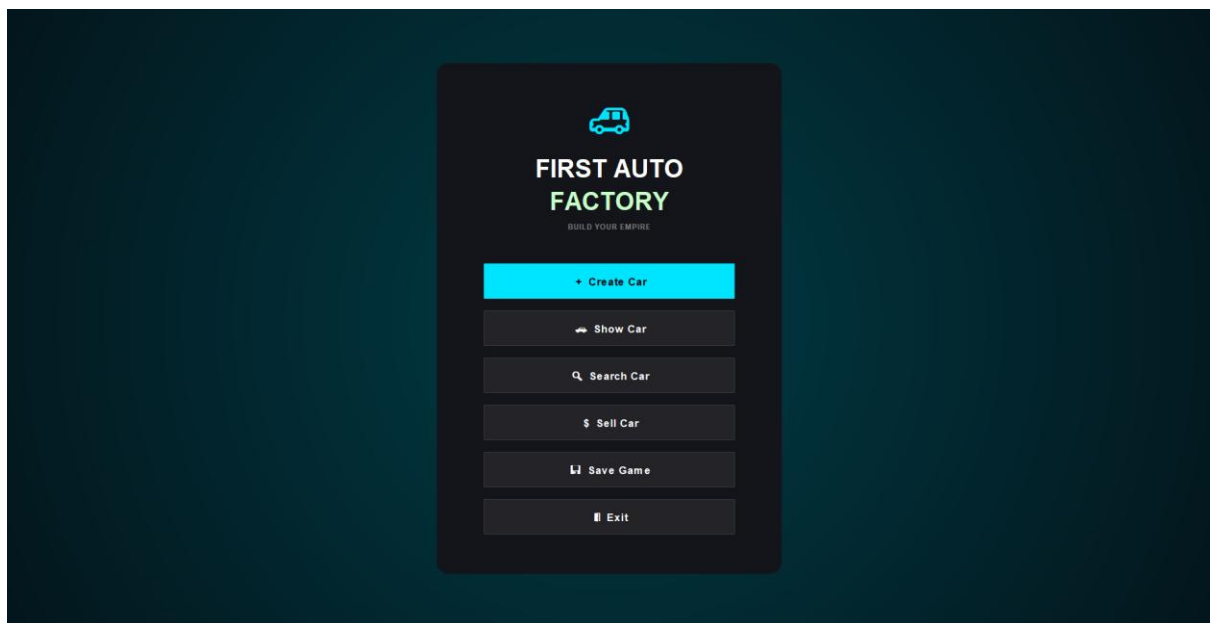
4. Abstraction

Menyembunyikan detail implementasi yang kompleks dan hanya menampilkan kerangka kerjanya.

Implementasinya adalah pada Class Car dideklarasikan sebagai public abstract class. Kita tidak bisa membuat objek new Car() secara langsung, melainkan harus membuat objek yang spesifik. Ini memastikan setiap mobil di pabrik pasti memiliki tipe yang jelas.

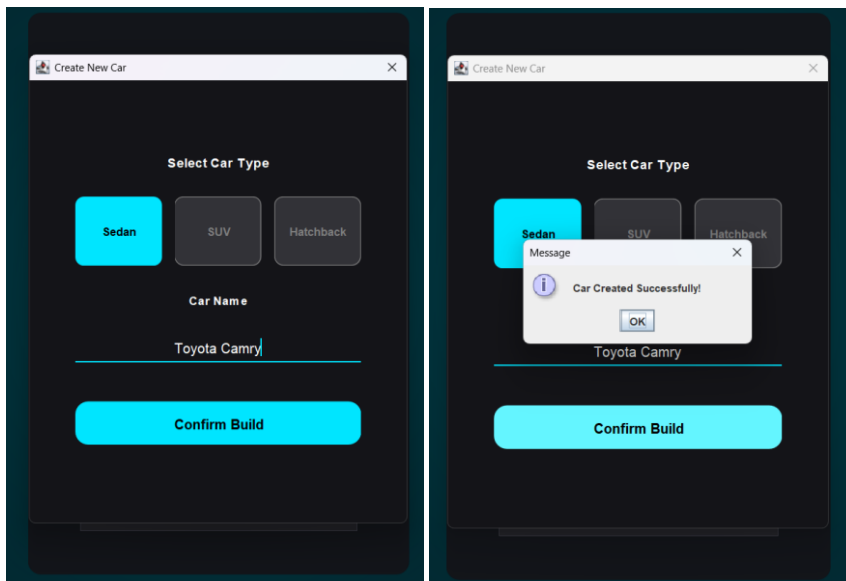
Screenshot Tampilan Aplikasi

Halaman utama aplikasi



Pada halaman utama berisi pilihan menu yang dapat dipilih oleh user.

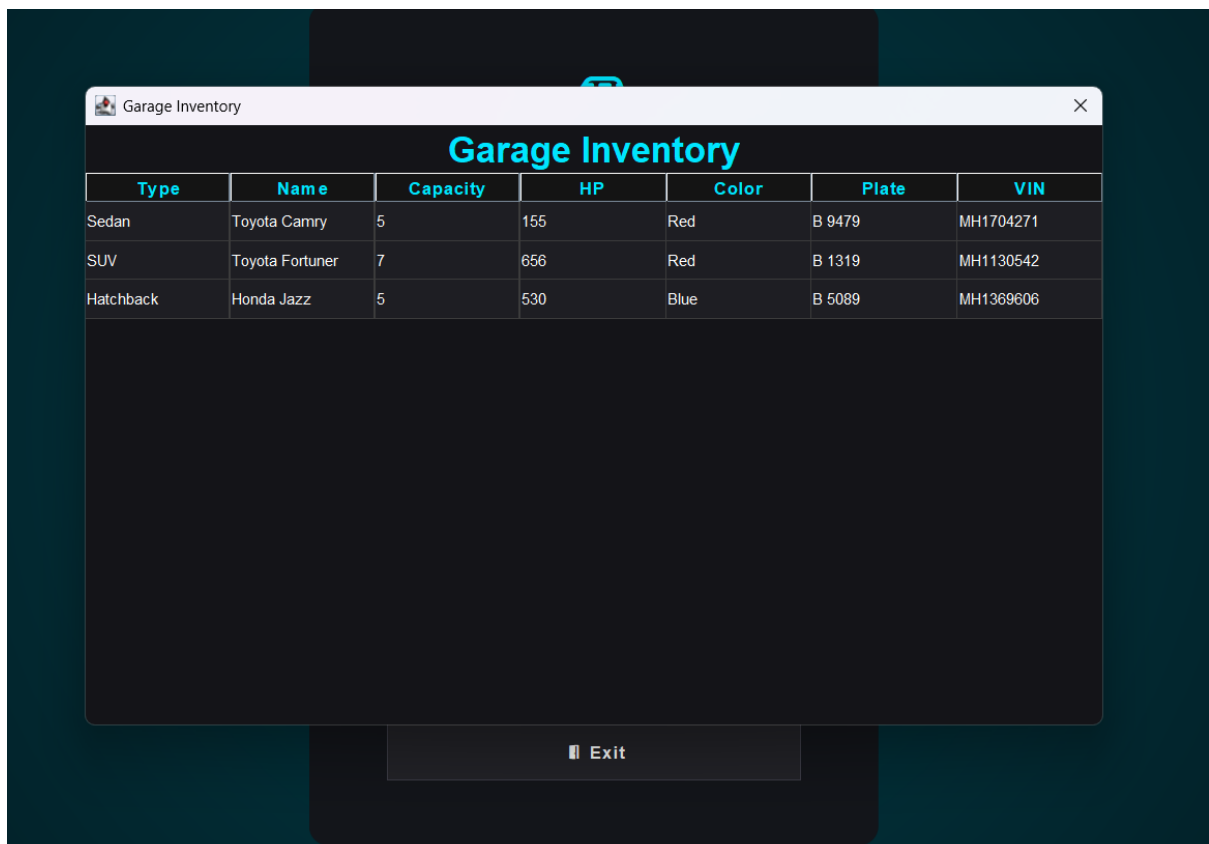
Halaman Create Car



The 'Create New Car' form consists of a title bar 'Create New Car' with a close button. Below the title bar is a section titled 'Select Car Type' with three buttons: 'Sedan' (highlighted in red), 'SUV', and 'Hatchback'. Below this is a 'Car Name' input field containing 'Toyota Camry'. At the bottom is a large red 'Confirm Build' button. In the second screenshot, a 'Message' dialog box is displayed over the form, showing an information icon, the text 'Car Created Successfully!', and an 'OK' button.

Pada halaman ini, user dapat membuat dan menambah mobil baru. User dapat memilih Car Type dan mengisi nama mobil. Setelah itu, user dapat menekan tombol Confirm Build dan muncul notifikasi bahwa mobil telah berhasil dibuat dan ditambahkan.

Halaman Show Car



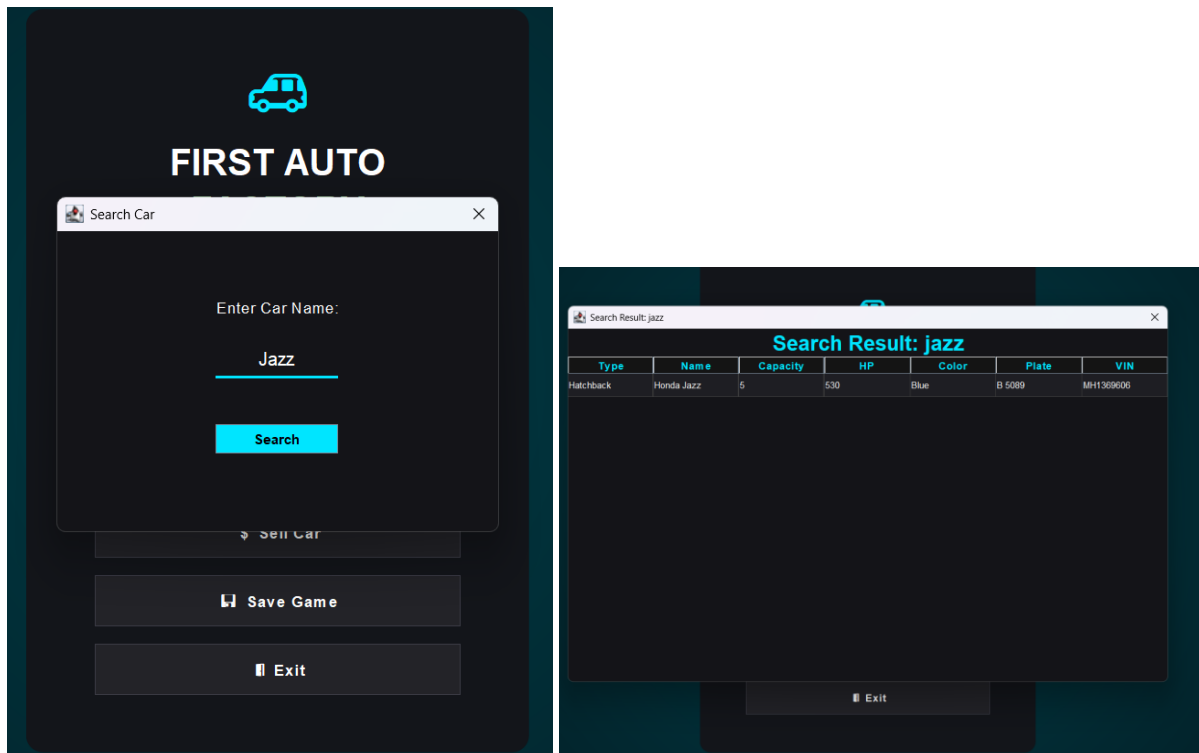
The 'Garage Inventory' window has a title bar 'Garage Inventory' with a close button. The main content area displays a table titled 'Garage Inventory' with the following data:

Type	Name	Capacity	HP	Color	Plate	VIN
Sedan	Toyota Camry	5	155	Red	B 9479	MH1704271
SUV	Toyota Fortuner	7	656	Red	B 1319	MH1130542
Hatchback	Honda Jazz	5	530	Blue	B 5089	MH1369606

Below the table is a large empty area and an 'Exit' button.

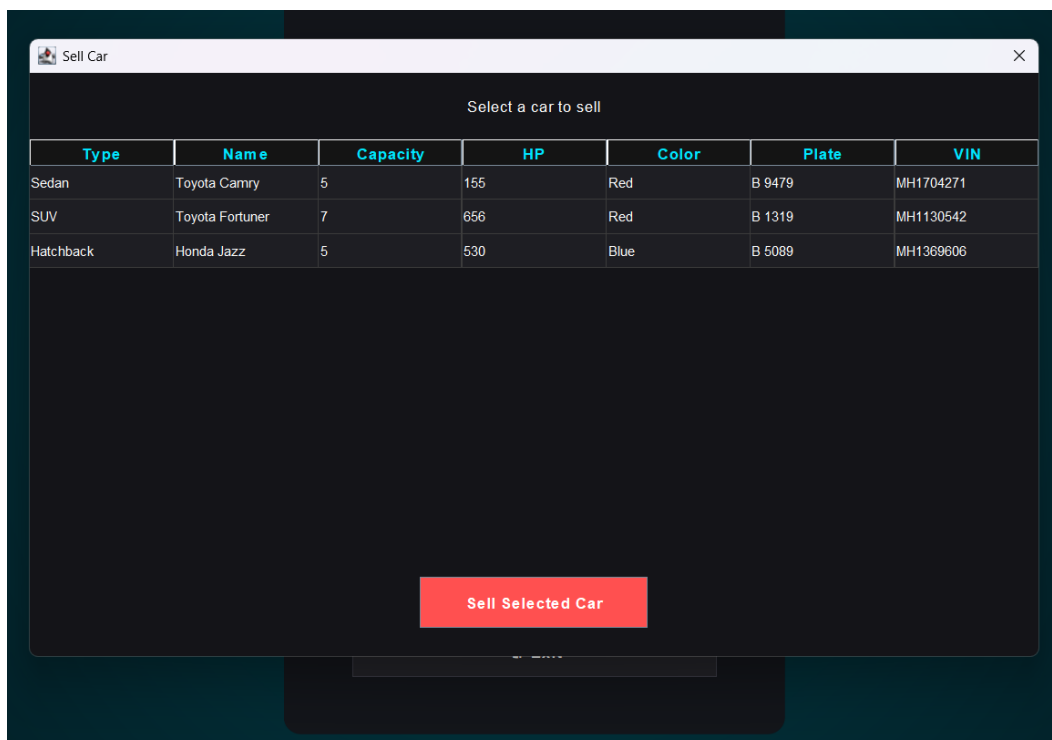
Pada halaman ini, user dapat melihat semua data mobil yang telah berhasil dibuat.

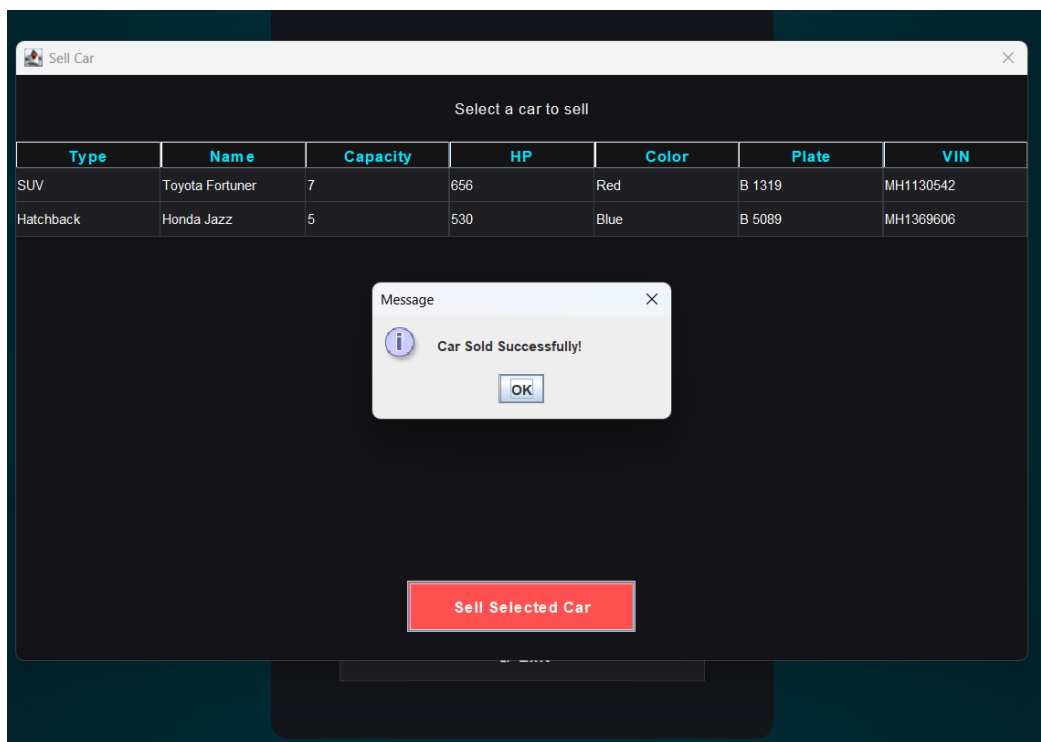
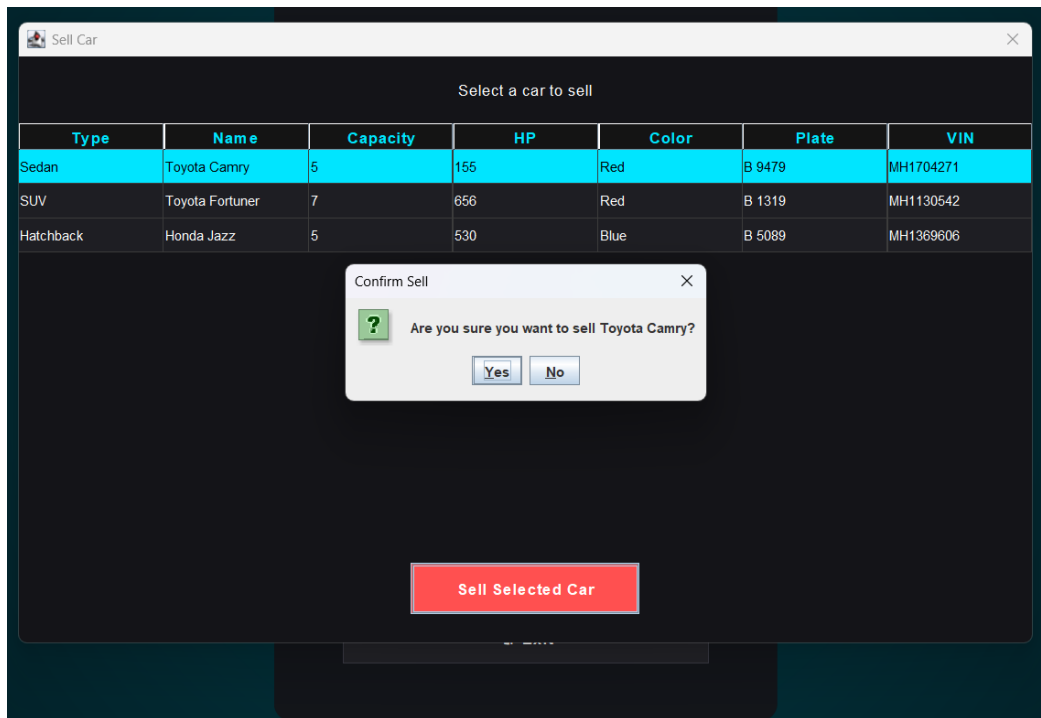
Halaman Search Car



Pada halaman ini, user dapat mencari mobil yang diinginkan dengan menuliskan nama mobil yang akan dicari. Setelah itu, user dapat menekan tombol search dan hasil pencarian akan otomatis ditampilkan.

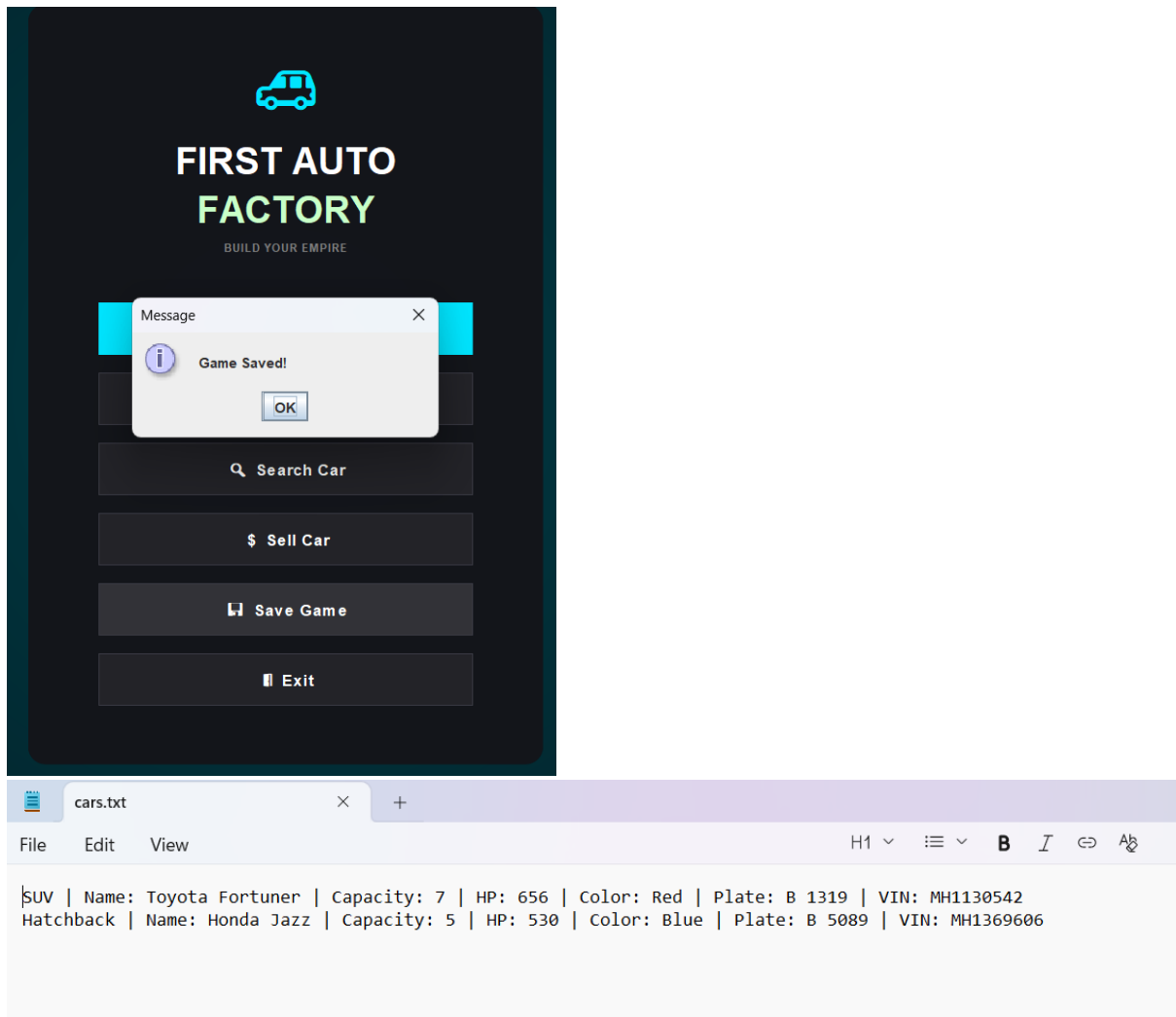
Halaman Sell Car





Pada halaman ini, user dapat memilih mobil yang ingin dijual dengan cara memilih mobil yang ingin dijual. Setelah itu, user dapat menekan tombol Sell Selected Car dan akan muncul pop up konfirmasi. Disini user dapat memutuskan untuk jual atau tidak jadi jual mobilnya. Setelah user pencet Yes, maka muncul pop-up bahwa mobil berhasil dijual dan dihapus dari inventaris.

Halaman Save Game



Jika user ingin menyimpan hasil inventaris, maka user dapat memilih Save Game. Setelah itu, akan muncul pop-up bahwa data telah berhasil disimpan dalam bentuk txt dalam file cars.txt.