PROYEK APLIKASI PENJUALAN KENDARAAN

```
import java.util.ArrayList;
import java.util.List;

class Vehicle {
    private String vin;
    private String make;
    private String model;
    private int year;
    private double price;

    public Vehicle(String vin, String make, String model, int year, double
price) {
        this.vin = vin;
        this.make = make;
        this.model = model;
        this.year = year;
        this.price = price;
    }

    public String getVin() {
        return vin;
    }

    public String displayInfo() {
        return year + " " + make + " " + model + " (VIN: " + vin + ") - $"
+ price;
    }
}

class Customer {
    private String name;
    private String contact;

    public Customer(String name, String contact) {
        this.name = name;
```

```
        this.contact = contact;
    }

    public String getName() {
        return name;
    }

    public String getContact() {
        return contact;
    }
}

class Sale {
    private Vehicle vehicle;
    private Customer customer;
    private String saleDate;
    private double salePrice;

    public Sale(Vehicle vehicle, Customer customer, String saleDate,
double salePrice) {
        this.vehicle = vehicle;
        this.customer = customer;
        this.saleDate = saleDate;
        this.salePrice = salePrice;
    }

    public String displaySaleInfo() {
        return "Sale Info:\nCustomer: " + customer.getName() + "\nContact:
" + customer.getContact() + "\nVehicle: " + vehicle.displayInfo() +
"\nDate: " + saleDate + "\nPrice: $" + salePrice;
    }
}

class Dealership {
    private List<Vehicle> inventory = new ArrayList<>();
    private List<Sale> sales = new ArrayList<>();

    public void addVehicle(Vehicle vehicle) {
        inventory.add(vehicle);
    }

    public Sale sellVehicle(String vin, Customer customer, String
saleDate, double salePrice) {
        for (Vehicle vehicle : inventory) {
            if (vehicle.getVin().equals(vin)) {
```

```
                inventory.remove(vehicle);
                Sale sale = new Sale(vehicle, customer, saleDate,
salePrice);
                sales.add(sale);
                return sale;
            }
        }
        return null;
    }

    public List<String> displayInventory() {
        List<String> inventoryList = new ArrayList<>();
        for (Vehicle vehicle : inventory) {
            inventoryList.add(vehicle.displayInfo());
        }
        return inventoryList;
    }

    public List<String> displaySales() {
        List<String> salesList = new ArrayList<>();
        for (Sale sale : sales) {
            salesList.add(sale.displaySaleInfo());
        }
        return salesList;
    }
}

public class Main {
    public static void main(String[] args) {
        Dealership dealership = new Dealership();

        // Menambahkan kendaraan ke dalam inventaris
        Vehicle vehicle1 = new Vehicle("1HGCM82633A004352", "Honda",
"Civic", 2023, 25000);
        Vehicle vehicle2 = new Vehicle("1FAFP404X1F158796", "Ford",
"Mustang", 2022, 30000);
        dealership.addVehicle(vehicle1);
        dealership.addVehicle(vehicle2);

        // Menampilkan inventaris
        System.out.println("Inventory:");
        for (String info : dealership.displayInventory()) {
            System.out.println(info);
        }
```

```
        // Membuat pelanggan
        Customer customer = new Customer("John Doe",
"john.doe@example.com");

        // Menjual kendaraan
        Sale sale = dealership.sellVehicle("1HGCM82633A004352", customer,
"2024-07-05", 24000);
        if (sale != null) {
            System.out.println("\nSale Successful!");
            System.out.println(sale.displaySaleInfo());
        } else {
            System.out.println("Vehicle not found in inventory.");
        }

        // Menampilkan penjualan
        System.out.println("\nSales:");
        for (String info : dealership.displaySales()) {
            System.out.println(info);
        }
    }
}
```

Penjelasan Kode

1. **Kelas Vehicle:**
    - Representasi dari kendaraan dengan atribut seperti VIN (Vehicle Identification Number), merek, model, tahun, dan harga.
    - Metode `displayInfo` untuk menampilkan informasi kendaraan.
2. **Kelas Customer:**
    - Representasi dari pelanggan dengan atribut nama dan kontak.
3. **Kelas Sale:**
    - Representasi dari transaksi penjualan dengan atribut kendaraan yang terjual, pelanggan, tanggal penjualan, dan harga penjualan.
    - Metode `displaySaleInfo` untuk menampilkan informasi penjualan.
4. **Kelas Dealership:**
    - Mengelola inventaris kendaraan dan daftar penjualan.
    - Metode `addVehicle` untuk menambahkan kendaraan ke inventaris.
    - Metode `sellVehicle` untuk menjual kendaraan berdasarkan VIN, serta mencatat penjualan dan menghapus kendaraan dari inventaris.
    - Metode `displayInventory` dan `displaySales` untuk menampilkan inventaris kendaraan dan daftar penjualan.
5. **Penggunaan Contoh:**
    - Menambahkan dua kendaraan ke dalam inventaris dealer.
    - Menampilkan inventaris.

- Membuat pelanggan baru.
- Menjual salah satu kendaraan ke pelanggan tersebut.
- Menampilkan informasi penjualan dan daftar penjualan setelah transaksi.