



TUGAS PRAKTIKUM

Soal Praktikum

Buatlah sebuah program sederhana untuk menginput data menggunakan inputan dinamis. Program harus memiliki atribut / method static, trapkan juga Naming Convnetion dengan benar.

Jawaban

Ketik jawaban disini ...

Source Code

```
import java.util.ArrayList;
import java.util.Scanner;

class User {
    private String nama;
    private int umur;

    public User(String nama, int umur) {
        this.nama = nama;
        this.umur = umur;
    }

    public String getNama() {
        return nama;
    }

    public int getUmur() {
        return umur;
    }
}

public class UserData {
    private static ArrayList<User> users = new ArrayList<>();

    public static void addUser(String nama, int umur) {
        users.add(new User(nama, umur));
    }

    public static void displayUsers() {
        if (users.isEmpty()) {
            System.out.println("Data User Tidak Ada");
            return;
        }
    }
}
```



TUGAS PRAKTIKUM

```
        System.out.println("User Data:");
        for (int i = 0; i < users.size(); i++) {
            User user = users.get(i);
            System.out.println((i + 1) + ". Nama: " +
user.getName() + ", umur: " + user.getUmur());
        }
    }

    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        while (true) {
            System.out.println("\nMenu:");
            System.out.println("1. Tambah User");
            System.out.println("2. Display User");
            System.out.println("3. Exit");
            System.out.print("Pilih 1-3: ");
            int choice = scanner.nextInt();
            scanner.nextLine();

            switch (choice) {
                case 1:
                    System.out.print("Masukkan Nama: ");
                    String name = scanner.nextLine();
                    System.out.print("Masukkan Umur: ");
                    int age = scanner.nextInt();
                    scanner.nextLine();
                    addUser(name, age);
                    System.out.println("Menambah User
Sukses!");
                    break;
                case 2:
                    displayUsers();
                    break;
                case 3:
                    System.out.println("Keluar");
                    scanner.close();
                    return;
                default:
                    System.out.println("Pilih yg Benar");
            }
        }
    }
}
```

Penjelasan

Tulis Penjelasan disini ...

Output



TUGAS PRAKTIKUM

Menu:

1. Tambah User
2. Display User
3. Exit

Pilih 1-3: 1

Masukkan Nama: Adam

Masukkan Umur: 17

Menambah User Sukses!

Menu:

1. Tambah User
2. Display User
3. Exit

Pilih 1-3: 2

User Data:

1. Nama: Adam, umur: 17

Menu:

1. Tambah User
2. Display User
3. Exit

Pilih 1-3: 3

Keluar



TUGAS PRAKTIKUM

Soal Praktikum

Buatlah sebuah sistem informasi sederhana untuk sebuah bengkel dengan fitur:

- Menampilkan, Menambahkan, Menghapus, dan Mengedit data kendaraan, mekanik, atau layanan.

- Setiap kendaraan harus memiliki mekanik yang menangani, dan setiap kendaraan mendapatkan layanan tertentu.

Jawaban

```
package Soal2;

import java.util.ArrayList;
import java.util.Scanner;

public class main {
    private static ArrayList<Kendaraan> daftarKendaraan = new
    ArrayList<>();
    private static ArrayList<Mekanik> daftarMekanik = new
    ArrayList<>();
    private static ArrayList<Layanan> daftarLayanan = new
    ArrayList<>();

    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        inisialisasiData();

        while (true) {
            tampilkanMenuUtama();
            int pilihan = scanner.nextInt();
            scanner.nextLine();

            switch (pilihan) {
                case 1 :
                    kelolaKendaraan(scanner);
                case 2 :
                    kelolaMekanik(scanner);
                case 3 :
                    kelolaLayanan(scanner);
                case 4 :
                    {
                        System.out.println("Keluar dari sistem.
Terima kasih!");
                        scanner.close();
                        return;
                    }
                default : System.out.println("Pilihan tidak
valid!");
            }
        }
    }
}
```



TUGAS PRAKTIKUM

```
    }  
    }  
}  
  
private static void inisialisasiData() {  
    daftarMekanik.add(new Mekanik("Adam"));  
    daftarMekanik.add(new Mekanik("Risky"));  
    daftarLayanan.add(new Layanan("Servis Body"));  
    daftarLayanan.add(new Layanan("Ganti Velg"));  
}  
  
private static void tampilkanMenuUtama() {  
    System.out.println("\n== Sistem Informasi Bengkel  
==");  
    System.out.println("1. Kelola Kendaraan");  
    System.out.println("2. Kelola Mekanik");  
    System.out.println("3. Kelola Layanan");  
    System.out.println("4. Keluar");  
    System.out.print("Pilih menu: ");  
}  
  
private static void kelolaKendaraan(Scanner scanner) {  
    while (true) {  
        System.out.println("\n== Kelola Kendaraan ==");  
        System.out.println("1. Tambah Kendaraan");  
        System.out.println("2. Tampilkan Kendaraan");  
        System.out.println("3. Hapus Kendaraan");  
        System.out.println("4. Kembali");  
        System.out.print("Pilih menu: ");  
        int pilihan = scanner.nextInt();  
        scanner.nextLine();  
  
        switch (pilihan) {  
            case 1 ->  
                tambahKendaraan(scanner);  
            case 2 ->  
                tampilkanKendaraan();  
            case 3 ->  
                hapusKendaraan(scanner);  
            case 4 -> {  
                return;  
            }  
            default -> System.out.println("Pilihan tidak  
valid!");  
        }  
    }  
}  
  
private static void kelolaMekanik(Scanner scanner) {  
    System.out.println("Fitur kelola mekanik dapat  
ditambahkan sesuai kebutuhan.");  
}  
  
private static void kelolaLayanan(Scanner scanner) {  
    System.out.println("Fitur kelola layanan dapat
```



TUGAS PRAKTIKUM

```
ditambahkan sesuai kebutuhan.");
    }

    private static void tambahKendaraan(Scanner scanner) {
        System.out.print("Masukkan nama kendaraan: ");
        String namaKendaraan = scanner.nextLine();

        System.out.println("Pilih mekanik:");
        tampilkanMekanik();
        System.out.print("Masukkan ID mekanik: ");
        int idMekanik = scanner.nextInt();
        scanner.nextLine();
        Mekanik mekanik = cariMekanikBerdasarkanId(idMekanik);

        System.out.println("Pilih layanan:");
        tampilkanLayanan();
        System.out.print("Masukkan ID layanan: ");
        int idLayanan = scanner.nextInt();
        scanner.nextLine();
        Layanan layanan = cariLayananBerdasarkanId(idLayanan);

        if (mekanik != null && layanan != null) {
            Kendaraan kendaraan = new Kendaraan(namaKendaraan,
mekanik, layanan);
            daftarKendaraan.add(kendaraan);
            System.out.println("Kendaraan berhasil
ditambahkan.");
        } else {
            System.out.println("Mekanik atau layanan tidak
ditemukan.");
        }
    }

    private static void tampilkanKendaraan() {
        if (daftarKendaraan.isEmpty()) {
            System.out.println("Tidak ada kendaraan.");
        } else {
            System.out.println("\n=== Daftar Kendaraan ===");
            for (int i = 0; i < daftarKendaraan.size(); i++) {
                Kendaraan k = daftarKendaraan.get(i);
                System.out.printf("%d. %s - Mekanik: %s -
Layanan: %s\n",
                                i + 1, k.getNamaKendaraan(),
k.getMekanik().getNamaMekanik(),
k.getLayanan().getNamaLayanan());
            }
        }
    }

    private static void hapusKendaraan(Scanner scanner) {
        tampilkanKendaraan();
        System.out.print("Masukkan nomor kendaraan yang akan
dihapus: ");
        int index = scanner.nextInt() - 1;
```



TUGAS PRAKTIKUM

```
        if (index >= 0 && index < daftarKendaraan.size()) {
            daftarKendaraan.remove(index);
            System.out.println("Kendaraan berhasil dihapus.");
        } else {
            System.out.println("Nomor kendaraan tidak
valid.");
        }
    }

    private static void tampilkanMekanik() {
        if (daftarMekanik.isEmpty()) {
            System.out.println("Tidak ada mekanik.");
        } else {
            for (Mekanik m : daftarMekanik) {
                System.out.printf("%d. %s\n",
m.getIdMekanik(), m.getNamaMekanik());
            }
        }
    }

    private static void tampilkanLayanan() {
        if (daftarLayanan.isEmpty()) {
            System.out.println("Tidak ada layanan.");
        } else {
            for (Layanan l : daftarLayanan) {
                System.out.printf("%d. %s\n",
l.getIdLayanan(), l.getNamaLayanan());
            }
        }
    }

    private static Mekanik cariMekanikBerdasarkanId(int id) {
        for (Mekanik m : daftarMekanik) {
            if (m.getIdMekanik() == id) {
                return m;
            }
        }
        return null;
    }

    private static Layanan cariLayananBerdasarkanId(int id) {
        for (Layanan l : daftarLayanan) {
            if (l.getIdLayanan() == id) {
                return l;
            }
        }
        return null;
    }
}

// Kelas Kendaraan
class Kendaraan {
    private String namaKendaraan;
    private Mekanik mekanik;
    private Layanan layanan;
```



TUGAS PRAKTIKUM

```
    public Kendaraan(String namaKendaraan, Mekanik mekanik,
Layanan layanan) {
        this.namaKendaraan = namaKendaraan;
        this.mekanik = mekanik;
        this.layanan = layanan;
    }

    public String getNamaKendaraan() {
        return namaKendaraan;
    }

    public Mekanik getMekanik() {
        return mekanik;
    }

    public Layanan getLayanan() {
        return layanan;
    }
}

// Kelas Mekanik
class Mekanik {
    private static int counterMekanik = 1;
    private int idMekanik;
    private String namaMekanik;

    public Mekanik(String namaMekanik) {
        this.idMekanik = counterMekanik++;
        this.namaMekanik = namaMekanik;
    }

    public int getIdMekanik() {
        return idMekanik;
    }

    public String getNamaMekanik() {
        return namaMekanik;
    }
}

// Kelas Layanan
class Layanan {
    private static int counterLayanan = 1;
    private int idLayanan;
    private String namaLayanan;

    public Layanan(String namaLayanan) {
        this.idLayanan = counterLayanan++;
        this.namaLayanan = namaLayanan;
    }

    public int getIdLayanan() {
        return idLayanan;
    }
}
```




TUGAS PRAKTIKUM

```
public String getNamaLayanan() {  
    return namaLayanan;  
}  
}
```

Source Code

Penjelasan

Tulis Penjelasan disini ...

Output



TUGAS PRAKTIKUM

```
=== Sistem Informasi Bengkel ===
```

1. Kelola Kendaraan
2. Kelola Mekanik
3. Kelola Layanan
4. Keluar

```
Pilih menu: 1
```

```
|
```

```
=== Kelola Kendaraan ===
```

1. Tambah Kendaraan
2. Tampilkan Kendaraan
3. Hapus Kendaraan
4. Kembali

```
Pilih menu: 1
```

```
Masukkan nama kendaraan: Mio
```

```
Pilih mekanik:
```

1. Adam
2. Risky

```
Masukkan ID mekanik: 1
```

```
Pilih layanan:
```

1. Servis Body
2. Ganti Velg

```
Masukkan ID layanan: 1
```

```
Kendaraan berhasil ditambahkan.
```

```
=== Kelola Kendaraan ===
```

1. Tambah Kendaraan
2. Tampilkan Kendaraan
3. Hapus Kendaraan
4. Kembali

```
Pilih menu: 2
```

```
Pilih menu: 2
```

```
=== Daftar Kendaraan ===
```

```
1. Mio - Mekanik: Adam - Layanan: Servis Body
```



TUGAS PRAKTIKUM



TUGAS PRAKTIKUM

Soal Praktikum

Buatlah program untuk manajemen sistem sewa kendaraan, yang terdiri dari kendaraan listrik dan kendaraan berbahan bakar bensin. Berikut fitur yang harus dimiliki aplikasi:

- Kelas abstrak Kendaraan yang memiliki metode abstrak `hitungHargaSewa()`
- Dua kelas turunan yaitu
 - a. KendaraanListrik: Menghitung harga sewa berdasarkan durasi sewa dan tingkat konsumsi daya.
 - b. KendaraanBensin: Menghitung harga sewa berdasarkan durasi sewa dan jumlah bahan bakar yang digunakan.
- Interface Diskon untuk menghitung diskon khusus untuk

KendaraanListrik. [Wajib]

Jawaban

Ketik jawaban disini ...

Source Code

```
package Soal3;

import java.util.ArrayList;
import java.util.Scanner;

abstract class Kendaraan {
    private String nama;
    private String nomorPlat;

    public Kendaraan(String nama, String nomorPlat) {
        this.nama = nama;
        this.nomorPlat = nomorPlat;
    }

    public String getNama() {
        return nama;
    }
}
```



TUGAS PRAKTIKUM

```
}

public String getNomorPlat() {
    return nomorPlat;
}

public abstract double hitungHargaSewa(int durasi);

@Override
public String toString() {
    return "Nama: " + nama + ", Plat: " + nomorPlat;
}
}

interface Diskon {
    double hitungDiskon(double hargaSewa);
}

class KendaraanListrik extends Kendaraan implements Diskon {
    private double konsumsiDaya; // dalam kWh
    private static final double HARGA_PER_KWH = 2000;

    public KendaraanListrik(String nama, String nomorPlat,
double konsumsiDaya) {
        super(nama, nomorPlat);
        this.konsumsiDaya = konsumsiDaya;
    }

    @Override
    public double hitungHargaSewa(int durasi) {
        double harga = durasi * konsumsiDaya * HARGA_PER_KWH;
        return harga - hitungDiskon(harga);
    }

    @Override
    public double hitungDiskon(double hargaSewa) {
        return hargaSewa * 0.1;
    }
}

class KendaraanBensin extends Kendaraan {
    private double konsumsiBahanBakar;
    private static final double HARGA_PER_LITER = 10000;

    public KendaraanBensin(String nama, String nomorPlat,
double konsumsiBahanBakar) {
        super(nama, nomorPlat);
        this.konsumsiBahanBakar = konsumsiBahanBakar;
    }

    @Override
    public double hitungHargaSewa(int durasi) {
        return durasi * konsumsiBahanBakar * HARGA_PER_LITER;
    }
}
```



TUGAS PRAKTIKUM

```
}

public class main {
    private static ArrayList<Kendaraan> daftarKendaraan = new
    ArrayList<>();

    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        daftarKendaraan.add(new KendaraanListrik("Tesla Model
S", "B 1234 TES", 15));
        daftarKendaraan.add(new KendaraanBensin("Toyota
Avanza", "D 5678 TOY", 8));

        while (true) {
            tampilkanMenu();
            int pilihan = scanner.nextInt();
            scanner.nextLine();

            switch (pilihan) {
                case 1 -> tambahKendaraan(scanner);
                case 2 -> tampilkanDaftarKendaraan();
                case 3 -> hitungHargaSewa(scanner);
                case 4 -> {
                    System.out.println("Keluar dari sistem.
Terima kasih!");
                    scanner.close();
                    return;
                }
                default -> System.out.println("Pilihan tidak
valid!");
            }
        }
    }

    private static void tampilkanMenu() {
        System.out.println("\n=== Sistem Sewa Kendaraan ===");
        System.out.println("1. Tambah Kendaraan");
        System.out.println("2. Tampilkan Daftar Kendaraan");
        System.out.println("3. Hitung Harga Sewa");
        System.out.println("4. Keluar");
        System.out.print("Pilih menu: ");
    }

    private static void tambahKendaraan(Scanner scanner) {
        System.out.println("\n=== Tambah Kendaraan ===");
        System.out.println("1. Kendaraan Listrik");
        System.out.println("2. Kendaraan Bensin");
        System.out.print("Pilih jenis kendaraan: ");
        int jenis = scanner.nextInt();
        scanner.nextLine();

        System.out.print("Masukkan nama kendaraan: ");
        String nama = scanner.nextLine();
        System.out.print("Masukkan nomor plat: ");
    }
}
```



TUGAS PRAKTIKUM

```
String nomorPlat = scanner.nextLine();

if (jenis == 1) {
    System.out.print("Masukkan konsumsi daya (kWh): ");
    double konsumsiDaya = scanner.nextDouble();
    daftarKendaraan.add(new KendaraanListrik(nama, nomorPlat, konsumsiDaya));
} else if (jenis == 2) {
    System.out.print("Masukkan konsumsi bahan bakar (liter): ");
    double konsumsiBahanBakar = scanner.nextDouble();
    daftarKendaraan.add(new KendaraanBensin(nama, nomorPlat, konsumsiBahanBakar));
} else {
    System.out.println("Jenis kendaraan tidak valid!");
}
System.out.println("Kendaraan berhasil ditambahkan.");
}

private static void tampilkanDaftarKendaraan() {
    if (daftarKendaraan.isEmpty()) {
        System.out.println("Tidak ada kendaraan yang tersedia.");
    } else {
        System.out.println("\n=== Daftar Kendaraan ===");
        for (int i = 0; i < daftarKendaraan.size(); i++) {
            System.out.printf("%d. %s\n", i + 1, daftarKendaraan.get(i));
        }
    }
}

private static void hitungHargaSewa(Scanner scanner) {
    tampilkanDaftarKendaraan();
    if (daftarKendaraan.isEmpty()) return;

    System.out.print("Pilih kendaraan berdasarkan nomor: ");
    int nomor = scanner.nextInt();
    if (nomor < 1 || nomor > daftarKendaraan.size()) {
        System.out.println("Nomor kendaraan tidak valid!");
        return;
    }

    Kendaraan kendaraan = daftarKendaraan.get(nomor - 1);
    System.out.print("Masukkan durasi sewa (jam): ");
    int durasi = scanner.nextInt();

    double hargaSewa = kendaraan.hitungHargaSewa(durasi);
    System.out.printf("Harga sewa untuk %s (%s) adalah Rp %.2f\n", kendaraan.getNama(), kendaraan.getNomorPlat(),
```



TUGAS PRAKTIKUM

```
hargaSewa);  
    }  
}
```

Penjelasan

Tulis Penjelasan disini ...

Output

```
=== Sistem Sewa Kendaraan ===  
1. Tambah Kendaraan  
2. Tampilkan Daftar Kendaraan  
3. Hitung Harga Sewa  
4. Keluar  
Pilih menu: 1  
  
=== Tambah Kendaraan ===  
1. Kendaraan Listrik  
2. Kendaraan Bensin  
Pilih jenis kendaraan: 1  
Masukkan nama kendaraan: tesla  
Masukkan nomor plat: 1 024 B  
Masukkan konsumsi daya (kWh): 10  
Kendaraan berhasil ditambahkan.  
  
=== Daftar Kendaraan ===  
1. Nama: Tesla Model S, Plat: B 1234 TES  
2. Nama: Toyota Avanza, Plat: D 5678 TOY  
3. Nama: tesla, Plat: 1 024 B  
  
=== Daftar Kendaraan ===  
1. Nama: Tesla Model S, Plat: B 1234 TES  
2. Nama: Toyota Avanza, Plat: D 5678 TOY  
3. Nama: tesla, Plat: 1 024 B  
Pilih kendaraan berdasarkan nomor:
```




TUGAS PRAKTIKUM



TUGAS PRAKTIKUM

Soal Praktikum

Kembangkan Soal 2 sehingga data bengkel yang dikelola dapat disimpan ke dalam file teks.

Peetunjuk:

- Simpan data Kendaraan, Mekanik, dan Layanan ke dalam file dataBengkel.txt saat program berakhir.
- Saat program dimulai, baca data dari file tersebut jika file sudah ada.
- Gunakan exception handling untuk menangani kasus seperti file tidak

ditemukan atau format data yang salah. [Wajib]

Jawaban

Ketik jawaban disini ...

Source Code

```
package Soal4;

import java.io.*;
import java.util.ArrayList;
import java.util.Scanner;

// Kelas Kendaraan
class Kendaraan {
    private String platNomor;
    private String pemilik;
    private Mekanik mekanik;
    private Layanan layanan;

    public Kendaraan(String platNomor, String pemilik, Mekanik mekanik, Layanan layanan) {
        this.platNomor = platNomor;
        this.pemilik = pemilik;
        this.mekanik = mekanik;
        this.layanan = layanan;
    }

    public String getPlatNomor() {
        return platNomor;
    }
}
```



TUGAS PRAKTIKUM

```
public String getPemilik() {
    return pemilik;
}

public Mekanik getMekanik() {
    return mekanik;
}

public Layanan getLayanan() {
    return layanan;
}

@Override
public String toString() {
    return platNomor + "," + pemilik + "," +
mekanik.getNama() + "," + layanan.getNamaLayanan();
}

public static Kendaraan fromString(String data,
ArrayList<Mekanik> mekanikList, ArrayList<Layanan>
layananList) {
    String[] parts = data.split(",");
    if (parts.length != 4) throw new
IllegalArgumentException("Format data Kendaraan salah.");
    Mekanik mekanik = mekanikList.stream()
        .filter(m -> m.getNama().equals(parts[2]))
        .findFirst()
        .orElseThrow(() -> new
IllegalArgumentException("Mekanik tidak ditemukan: " +
parts[2]));
    Layanan layanan = layananList.stream()
        .filter(l ->
l.getNamaLayanan().equals(parts[3]))
        .findFirst()
        .orElseThrow(() -> new
IllegalArgumentException("Layanan tidak ditemukan: " +
parts[3]));
    return new Kendaraan(parts[0], parts[1], mekanik,
layanan);
}

// Kelas Mekanik
class Mekanik {
    private String nama;
    private String spesialisasi;

    public Mekanik(String nama, String spesialisasi) {
        this.nama = nama;
        this.spesialisasi = spesialisasi;
    }

    public String getNama() {
        return nama;
    }
}
```



TUGAS PRAKTIKUM

```
public String getSpesialisasi() {
    return spesialisasi;
}

@Override
public String toString() {
    return nama + "," + spesialisasi;
}

public static Mekanik fromString(String data) {
    String[] parts = data.split(",");
    if (parts.length != 2) throw new
IllegalArgumentException("Format data Mekanik salah.");
    return new Mekanik(parts[0], parts[1]);
}
}

// Kelas Layanan
class Layanan {
    private String namaLayanan;
    private double harga;

    public Layanan(String namaLayanan, double harga) {
        this.namaLayanan = namaLayanan;
        this.harga = harga;
    }

    public String getNamaLayanan() {
        return namaLayanan;
    }

    public double getHarga() {
        return harga;
    }

    @Override
    public String toString() {
        return namaLayanan + "," + harga;
    }

    public static Layanan fromString(String data) {
        String[] parts = data.split(",");
        if (parts.length != 2) throw new
IllegalArgumentException("Format data Layanan salah.");
        return new Layanan(parts[0],
Double.parseDouble(parts[1]));
    }
}

// Kelas Bengkel
public class main {
    private static final String FILE_NAME = "dataBengkel.txt";
    private static ArrayList<Kendaraan> daftarKendaraan = new
ArrayList<>();
}
```



TUGAS PRAKTIKUM

```
private static ArrayList<Mekanik> daftarMekanik = new
ArrayList<>();
private static ArrayList<Layanan> daftarLayanan = new
ArrayList<>();

public static void main(String[] args) {
    bacaDataDariFile();

    Scanner scanner = new Scanner(System.in);
    while (true) {
        tampilkanMenu();
        int pilihan = scanner.nextInt();
        scanner.nextLine();

        switch (pilihan) {
            case 1 -> tambahKendaraan(scanner);
            case 2 -> tambahMekanik(scanner);
            case 3 -> tambahLayanan(scanner);
            case 4 -> tampilkanData();
            case 5 -> {
                simpanDataKeFile();
                System.out.println("Data disimpan. Program
selesai.");
                scanner.close();
                return;
            }
            default -> System.out.println("Pilihan tidak
valid!");
        }
    }

    private static void tampilkanMenu() {
        System.out.println("\n=== Manajemen Bengkel ===");
        System.out.println("1. Tambah Kendaraan");
        System.out.println("2. Tambah Mekanik");
        System.out.println("3. Tambah Layanan");
        System.out.println("4. Tampilkan Semua Data");
        System.out.println("5. Keluar");
        System.out.print("Pilih menu: ");
    }

    private static void tambahKendaraan(Scanner scanner) {
        System.out.println("\n=== Tambah Kendaraan ===");
        System.out.print("Plat Nomor: ");
        String platNomor = scanner.nextLine();
        System.out.print("Nama Pemilik: ");
        String pemilik = scanner.nextLine();
        System.out.println("Pilih Mekanik:");
        for (int i = 0; i < daftarMekanik.size(); i++) {
            System.out.println((i + 1) + ". " +
daftarMekanik.get(i).getNama());
        }
        int mekanikIndex = scanner.nextInt() - 1;
        scanner.nextLine();
    }
}
```



TUGAS PRAKTIKUM

```
System.out.println("Pilih Layanan:");
for (int i = 0; i < daftarLayanan.size(); i++) {
    System.out.println((i + 1) + ". " +
daftarLayanan.get(i).getNamaLayanan());
}
int layananIndex = scanner.nextInt() - 1;
scanner.nextLine();

daftarKendaraan.add(new Kendaraan(platNomor, pemilik,
daftarMekanik.get(mekanikIndex),
daftarLayanan.get(layananIndex));
System.out.println("Kendaraan berhasil ditambahkan.");
}

private static void tambahMekanik(Scanner scanner) {
    System.out.println("\n=== Tambah Mekanik ===");
    System.out.print("Nama Mekanik: ");
    String nama = scanner.nextLine();
    System.out.print("Spesialisasi Mekanik: ");
    String spesialisasi = scanner.nextLine();
    daftarMekanik.add(new Mekanik(nama, spesialisasi));
    System.out.println("Mekanik berhasil ditambahkan.");
}

private static void tambahLayanan(Scanner scanner) {
    System.out.println("\n=== Tambah Layanan ===");
    System.out.print("Nama Layanan: ");
    String namaLayanan = scanner.nextLine();
    System.out.print("Harga Layanan: ");
    double harga = scanner.nextDouble();
    daftarLayanan.add(new Layanan(namaLayanan, harga));
    System.out.println("Layanan berhasil ditambahkan.");
}

private static void tampilkanData() {
    System.out.println("\n=== Data Kendaraan ===");
    if (daftarKendaraan.isEmpty())
System.out.println("Tidak ada data kendaraan.");
    else daftarKendaraan.forEach(System.out::println);

    System.out.println("\n=== Data Mekanik ===");
    if (daftarMekanik.isEmpty()) System.out.println("Tidak
ada data mekanik.");
    else daftarMekanik.forEach(System.out::println);

    System.out.println("\n=== Data Layanan ===");
    if (daftarLayanan.isEmpty()) System.out.println("Tidak
ada data layanan.");
    else daftarLayanan.forEach(System.out::println);
}

private static void simpanDataKeFile() {
    try (PrintWriter writer = new PrintWriter(new
FileWriter(FILE_NAME))) {
```



TUGAS PRAKTIKUM

```
        writer.println("Kendaraan:");
        for (Kendaraan kendaraan : daftarKendaraan)
writer.println(kendaraan);

        writer.println("Mekanik:");
        for (Mekanik mekanik : daftarMekanik)
writer.println(mekanik);

        writer.println("Layanan:");
        for (Layanan layanan : daftarLayanan)
writer.println(layanan);
    } catch (IOException e) {
        System.out.println("Gagal menyimpan data: " +
e.getMessage());
    }
}

private static void bacaDataDariFile() {
    File file = new File(FILE_NAME);
    if (!file.exists()) return;

    try (BufferedReader reader = new BufferedReader(new
FileReader(file))) {
        String line;
        String section = "";

        while ((line = reader.readLine()) != null) {
            if (line.equals("Kendaraan:")) section =
"Kendaraan";
            else if (line.equals("Mekanik:")) section =
"Mekanik";
            else if (line.equals("Layanan:")) section =
"Layanan";
            else if (!line.isBlank()) {
                switch (section) {
                    case "Kendaraan" ->
daftarKendaraan.add(Kendaraan.fromString(line, daftarMekanik,
daftarLayanan));
                    case "Mekanik" ->
daftarMekanik.add(Mekanik.fromString(line));
                    case "Layanan" ->
daftarLayanan.add(Layanan.fromString(line));
                }
            }
        }
    } catch (IOException | IllegalArgumentException e) {
        System.out.println("Gagal membaca data: " +
e.getMessage());
    }
}
}
```

Tulis kode program dikotak ini...



TUGAS PRAKTIKUM

1 kotak dan 1 Penjelasan untuk 1 Class

Penjelasan

Tulis Penjelasan disini ...

Output

Masukan screenshot output disini



TUGAS PRAKTIKUM

Soal Praktikum

Ketik soal disini ...

Jawaban

Ketik jawaban disini ...

Source Code

Tulis kode program dikotak ini...

1 kotak dan 1 Penjelasan untuk 1 Class

Penjelasan

Tulis Penjelasan disini ...

Output

Masukan screenshot output disini



TUGAS PRAKTIKUM

Soal Praktikum

Ketik soal disini ...

Jawaban

Ketik jawaban disini ...

Source Code

Tulis kode program dikotak ini...

1 kotak dan 1 Penjelasan untuk 1 Class

Penjelasan

Tulis Penjelasan disini ...

Output

Masukan screenshot output disini