

LAPORAN AKHIR PRAKTIKUM

Mata Praktikum : Rekayasa Perangkat Lunak 2
Kelas : 4IA24
Praktikum ke- : 2
Tanggal : 21 Oktober 2024
Materi : OOP
NPM : 50421856
Nama : Muhamad Alfin Surya Pratama
Ketua Asisten :
Nama Asisten :
Paraf Asisten :
Jumlah Lembar :



LABORATORIUM TEKNIK INFORMATIKA

UNIVERSITAS GUNADARMA

2024

LISTING

```
import java.util.*;
import java.time.LocalDateTime;
import java.time.Duration;

class Kendaraan {
    String platNomor;
    String jenis;
    double tarifPerHari;
    boolean tersedia;

    public Kendaraan(String platNomor, String jenis, double tarifPerHari) {
        this.platNomor = platNomor;
        this.jenis = jenis;
        this.tarifPerHari = tarifPerHari;
        this.tersedia = true;
    }
}

class Penyewaan {
    Kendaraan kendaraan;
    String namaPenyewa;
    LocalDateTime waktuMulai;

    public Penyewaan(Kendaraan kendaraan, String namaPenyewa) {
        this.kendaraan = kendaraan;
        this.namaPenyewa = namaPenyewa;
        this.waktuMulai = LocalDateTime.now();
    }
}

public class SistemRentalKendaraan {
    private static List<Kendaraan> daftarKendaraan = new ArrayList<>();
    private static Map<String, Penyewaan> penyewaanAktif = new HashMap<>();
    private static Scanner scanner = new Scanner(System.in);

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

        while (true) {
            System.out.println("\n=== Sistem Rental Kendaraan ===");
            System.out.println("1. Sewa Kendaraan");
            System.out.println("2. Kembalikan Kendaraan");
            System.out.println("3. Lihat Kendaraan Tersedia");
            System.out.println("4. Keluar");
            System.out.print("Pilih menu (1-4): ");

            int pilihan = scanner.nextInt();
            scanner.nextLine(); // Consume newline

            switch (pilihan) {
                case 1:
                    sewaKendaraan();
                    break;
                case 2:
                    kembalikanKendaraan();
                    break;
                case 3:
                    lihatKendaraanTersedia();
                    break;
                case 4:
                    System.out.println("Terima kasih telah menggunakan Sistem Rental Kendaraan.");
                    System.exit(0);
                default:
                    System.out.println("Pilihan tidak valid.");
            }
        }
    }

    private static void inisialisasiKendaraan() {
        daftarKendaraan.add(new Kendaraan("B 1234 CD", "Mobil", 300000));
        daftarKendaraan.add(new Kendaraan("B 5678 EF", "Motor", 100000));
    }
}
```

```

        daftarKendaraan.add(new Kendaraan("B 9012 GH", "Mobil", 350000));
    }

    private static void sewaKendaraan() {
        System.out.println("Kendaraan tersedia:");
        for (Kendaraan k : daftarKendaraan) {
            if (k.tersedia) {
                System.out.println(k.platNomor + " - " + k.jenis + " (Rp " + k.tarifPerHari + "/hari)");
            }
        }

        System.out.print("Masukkan plat nomor kendaraan yang ingin disewa: ");
        String platNomor = scanner.nextLine();

        Kendaraan kendaraanPilihan = daftarKendaraan.stream()
            .filter(k -> k.platNomor.equals(platNomor) && k.tersedia)
            .findFirst()
            .orElse(null);

        if (kendaraanPilihan != null) {
            System.out.print("Masukkan nama penyewa: ");
            String namaPenyewa = scanner.nextLine();

            kendaraanPilihan.tersedia = false;
            Penyewaan penyewaan = new Penyewaan(kendaraanPilihan, namaPenyewa);
            penyewaanAktif.put(platNomor, penyewaan);

            System.out.println("Kendaraan berhasil disewa.");
        } else {
            System.out.println("Kendaraan tidak tersedia atau plat nomor tidak valid.");
        }
    }

    private static void kembalikanKendaraan() {
        System.out.print("Masukkan plat nomor kendaraan yang dikembalikan: ");
        String platNomor = scanner.nextLine();

        Penyewaan penyewaan = penyewaanAktif.remove(platNomor);
        if (penyewaan != null) {
            penyewaan.kendaraan.tersedia = true;

            long durasiSewa = Duration.between(penyewaan.waktuMulai, LocalDateTime.now()).toHours();
            double biaya = Math.ceil(durasiSewa / 24.0) * penyewaan.kendaraan.tarifPerHari;

            System.out.println("Biaya sewa: Rp " + biaya);
            System.out.println("Kendaraan berhasil dikembalikan.");
        } else {
            System.out.println("Kendaraan dengan plat nomor tersebut tidak sedang disewa.");
        }
    }

    private static void lihatKendaraanTersedia() {
        System.out.println("Kendaraan tersedia:");
        boolean adaKendaraanTersedia = false;
        for (Kendaraan k : daftarKendaraan) {
            if (k.tersedia) {
                System.out.println(k.platNomor + " - " + k.jenis + " (Rp " + k.tarifPerHari + "/hari)");
                adaKendaraanTersedia = true;
            }
        }
        if (!adaKendaraanTersedia) {
            System.out.println("Tidak ada kendaraan yang tersedia saat ini.");
        }
    }
}

```

LOGIKA

```
class Kendaraan {
```

```
    String platNomor;
```

```
    String jenis;
```

```
    double tarifPerHari;
```

```
    boolean tersedia;
```

```
}
```

- Kelas ini menyimpan informasi setiap kendaraan
- tersedia digunakan untuk menandai apakah kendaraan bisa disewa (true) atau sedang disewa (false)

```
class Penyewaan {
```

```
    Kendaraan kendaraan;
```

```
    String namaPenyewa;
```

```
    LocalDateTime waktuMulai;
```

```
}
```

- Kelas ini menyimpan data transaksi penyewaan
- Mencatat waktu mulai sewa untuk perhitungan biaya

```
private static List<Kendaraan> daftarKendaraan = new ArrayList<>();
```

```
private static Map<String, Penyewaan> penyewaanAktif = new HashMap<>();
```

- daftarKendaraan: Menyimpan semua kendaraan yang dimiliki rental
- penyewaanAktif: Menyimpan data penyewaan yang sedang berlangsung, dengan platNomor sebagai key

```
private static void inisialisasiKendaraan() {
```

```
    daftarKendaraan.add(new Kendaraan("B 1234 CD", "Mobil", 300000));
```

```
    daftarKendaraan.add(new Kendaraan("B 5678 EF", "Motor", 100000));
```

```
    daftarKendaraan.add(new Kendaraan("B 9012 GH", "Mobil", 350000));
```

```
}
```

- Menambahkan data awal kendaraan saat program dimulai
- Ada 2 mobil dan 1 motor dengan tarif berbeda

```

private static void sewaKendaraan() {
    // Menampilkan kendaraan yang tersedia
    for (Kendaraan k : daftarKendaraan) {
        if (k.tersedia) {
            System.out.println(k.platNomor + " - " + k.jenis + " (Rp " + k.tarifPerHari + "/hari)");
        }
    }

    // Mencari kendaraan berdasarkan plat nomor
    Kendaraan kendaraanPilihan = daftarKendaraan.stream()
        .filter(k -> k.platNomor.equals(platNomor) && k.tersedia)
        .findFirst()
        .orElse(null);

    if (kendaraanPilihan != null) {
        // Proses penyewaan
        kendaraanPilihan.tersedia = false; // Tandai sudah tidak tersedia
        Penyewaan penyewaan = new Penyewaan(kendaraanPilihan, namaPenyewa);
        penyewaanAktif.put(platNomor, penyewaan); // Simpan ke daftar aktif
    }
}

private static void kembalikanKendaraan() {
    // Ambil dan hapus data penyewaan dari daftar aktif
    Penyewaan penyewaan = penyewaanAktif.remove(platNomor);

    if (penyewaan != null) {
        // Hitung durasi dalam jam
        long durasiSewa = Duration.between(penyewaan.waktuMulai,
            LocalDateTime.now()).toHours();

        // Hitung biaya: dibulatkan ke atas per hari
        double biaya = Math.ceil(durasiSewa / 24.0) * penyewaan.kendaraan.tarifPerHari;

        // Tandai kendaraan tersedia kembali
    }
}

```

```

        penyewaan.kendaraan.tersedia = true;
    }
}

```

- Menggunakan Duration.between() untuk menghitung selisih waktu dalam jam
- Math.ceil(durasiSewa / 24.0) untuk membulatkan ke atas jumlah hari
- Contoh:

Jika disewa 25 jam = 2 hari (dibulatkan ke atas)

Jika tarif 300.000/hari = $2 \times 300.000 = 600.000$

```

private static void lihatKendaraanTersedia() {
    boolean adaKendaraanTersedia = false;
    for (Kendaraan k : daftarKendaraan) {
        if (k.tersedia) {
            // Tampilkan kendaraan yang tersedia
            adaKendaraanTersedia = true;
        }
    }
    // Jika tidak ada yang tersedia, tampilkan pesan
    if (!adaKendaraanTersedia) {
        System.out.println("Tidak ada kendaraan yang tersedia saat ini.");
    }
}

```

- Memeriksa ketersediaan kendaraan sebelum penyewaan
- Memeriksa keberadaan penyewaan aktif saat pengembalian
- Menggunakan stream().filter() untuk mencari kendaraan yang sesuai

OUTPUT

```
=== Sistem Rental Kendaraan ===
1. Sewa Kendaraan
2. Kembalikan Kendaraan
3. Lihat Kendaraan Tersedia
4. Keluar
Pilih menu (1-4): 3

Kendaraan tersedia:
B 5678 EF - Motor (Rp 100000.0/hari)
B 9012 GH - Mobil (Rp 350000.0/hari)

=== Sistem Rental Kendaraan ===
1. Sewa Kendaraan
2. Kembalikan Kendaraan
3. Lihat Kendaraan Tersedia
4. Keluar
Pilih menu (1-4): 2

Masukkan plat nomor kendaraan yang dikembalikan: B 1234 CD
Biaya sewa: Rp 300000.0
Kendaraan berhasil dikembalikan.
```