

LAPORAN AKHIR PRAKTIKUM

Mata Praktikum : Rekayasa Perangkat Lunak 2
Kelas : 4IA24
Praktikum ke- : 3
Tanggal : 28 Oktober 2024
Materi :
NPM : 50421856
Nama : Muhamad Alfin Surya Pratama
Ketua Asisten :
Nama Asisten : Robby Nugraha
Paraf Asisten :
Jumlah Lembar : 9 Lembar



LABORATORIUM TEKNIK INFORMATIKA

UNIVERSITAS GUNADARMA

2024

LISTING

```
import javax.swing.*;
import javax.swing.table.*;
import java.awt.*;
import java.awt.event.*;
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 SistemRentalKendaraanGUI extends JFrame {
    private List<Kendaraan> daftarKendaraan = new ArrayList<>();
    private Map<String, Penyewaan> penyewaanAktif = new HashMap<>();

    private JTable tabelKendaraan;
    private DefaultTableModel modelTabel;
    private JButton btnSewa, btnKembali, btnRefresh;
    private JPanel mainPanel;

    public SistemRentalKendaraanGUI() {
        // Setup frame
        super("Sistem Rental Kendaraan");
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setSize(800, 500);
        setLocationRelativeTo(null);

        // Inisialisasi data
        inisialisasiKendaraan();
    }
}
```

```

    // Setup komponen
    setupKomponen();

    // Refresh tabel
    refreshTabel();
}

private 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 void setupKomponen() {
    // Setup main panel
    mainPanel = new JPanel(new BorderLayout(10, 10));
    mainPanel.setBorder(BorderFactory.createEmptyBorder(10, 10, 10, 10));

    // Setup tabel
    String[] kolom = {"Plat Nomor", "Jenis", "Tarif per Hari", "Status"};
    modelTabel = new DefaultTableModel(kolom, 0) {
        @Override
        public boolean isCellEditable(int row, int column) {
            return false;
        }
    };

    tabelKendaraan = new JTable(modelTabel);

    JScrollPane scrollPane = new JScrollPane(tabelKendaraan);
    mainPanel.add(scrollPane, BorderLayout.CENTER);

    // Setup panel tombol
    JPanel buttonPanel = new JPanel(new FlowLayout(FlowLayout.CENTER, 10, 10));

    btnSewa = new JButton("Sewa Kendaraan");
    btnKembali = new JButton("Kembalikan Kendaraan");
    btnRefresh = new JButton("Refresh");

    buttonPanel.add(btnSewa);
    buttonPanel.add(btnKembali);
    buttonPanel.add(btnRefresh);

    mainPanel.add(buttonPanel, BorderLayout.SOUTH);

    // Tambah action listeners
    btnSewa.addActionListener(e -> sewaKendaraan());
    btnKembali.addActionListener(e -> kembalikanKendaraan());
    btnRefresh.addActionListener(e -> refreshTabel());

    // Tambah panel ke frame
    add(mainPanel);
}

private void refreshTabel() {
    modelTabel.setRowCount(0);
}

```

```

        for (Kendaraan k : daftarKendaraan) {
            modelTabel.addRow(new Object[]{
                k.platNomor,
                k.jenis,
                String.format("Rp %.2f", k.tarifPerHari),
                k.tersedia ? "Tersedia" : "Disewa"
            });
        }
    }

    private void sewaKendaraan() {
        int selectedRow = tabelKendaraan.getSelectedRow();
        if (selectedRow == -1) {
            JOptionPane.showMessageDialog(this,
                "Pilih kendaraan yang ingin disewa terlebih dahulu",
                "Peringatan",
                JOptionPane.WARNING_MESSAGE);
            return;
        }

        String platNomor = (String) tabelKendaraan.getValueAt(selectedRow, 0);
        Kendaraan kendaraan = daftarKendaraan.stream()
            .filter(k -> k.platNomor.equals(platNomor) && k.tersedia)
            .findFirst()
            .orElse(null);

        if (kendaraan == null || !kendaraan.tersedia) {
            JOptionPane.showMessageDialog(this,
                "Kendaraan tidak tersedia untuk disewa",
                "Error",
                JOptionPane.ERROR_MESSAGE);
            return;
        }

        String namaPenyewa = JOptionPane.showInputDialog(this,
            "Masukkan nama penyewa:",
            "Sewa Kendaraan",
            JOptionPane.QUESTION_MESSAGE);

        if (namaPenyewa != null && !namaPenyewa.trim().isEmpty()) {
            kendaraan.tersedia = false;
            Penyewaan penyewaan = new Penyewaan(kendaraan, namaPenyewa);
            penyewaanAktif.put(platNomor, penyewaan);

            JOptionPane.showMessageDialog(this,
                "Kendaraan berhasil disewa oleh " + namaPenyewa,
                "Sukses",
                JOptionPane.INFORMATION_MESSAGE);

            refreshTabel();
        }
    }

    private void kembalikanKendaraan() {

```

```

int selectedRow = tabelKendaraan.getSelectedRow();
if (selectedRow == -1) {
    JOptionPane.showMessageDialog(this,
        "Pilih kendaraan yang ingin dikembalikan terlebih dahulu",
        "Peringatan",
        JOptionPane.WARNING_MESSAGE);
    return;
}

String platNomor = (String) tabelKendaraan.getValueAt(selectedRow, 0);
Penyewaan penyewaan = penyewaanAktif.remove(platNomor);

if (penyewaan == null) {
    JOptionPane.showMessageDialog(this,
        "Kendaraan ini tidak sedang disewa",
        "Error",
        JOptionPane.ERROR_MESSAGE);
    return;
}

penyewaan.kendaraan.tersedia = true;

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

JOptionPane.showMessageDialog(this,
    String.format("Biaya sewa: Rp %.2f\nKendaraan berhasil dikembalikan.", biaya),

```

```

        "Informasi Pengembalian",
        JOptionPane.INFORMATION_MESSAGE);

```

```

    refreshTabel();
}

```

```

public static void main(String[] args) {
    try {
        UIManager.setLookAndFeel(UIManager.getSystemLookAndFeelClassName());
    } catch (Exception e) {
        e.printStackTrace();
    }

    SwingUtilities.invokeLater(() -> {
        new SistemRentalKendaraanGUI().setVisible(true);
    });
}
}

```

LOGIKA

Struktur Data

// Class untuk menyimpan data kendaraan

```
class Kendaraan {  
    String platNomor    // Identifikasi unik kendaraan  
    String jenis        // Jenis kendaraan (Mobil/Motor)  
    double tarifPerHari // Biaya sewa per hari  
    boolean tersedia    // Status ketersediaan  
}
```

// Class untuk mencatat penyewaan

```
class Penyewaan {  
    Kendaraan kendaraan    // Kendaraan yang disewa  
    String namaPenyewa     // Nama penyewa  
    LocalDateTime waktuMulai // Waktu mulai sewa  
}
```

// Penyimpanan data di class utama

```
List<Kendaraan> daftarKendaraan // Menyimpan semua kendaraan  
Map<String, Penyewaan> penyewaanAktif // Menyimpan penyewaan yang sedang berlangsung
```

Inisialisasi

- Program dimulai (main)
- Set tampilan native OS
- Buat window utama
- Inisialisasi 3 kendaraan default
- Setup komponen GUI
- Tampilkan tabel kendaraan

Proses Penyewaan

```
sewaKendaraan() {
```

1. Cek apakah ada baris terpilih di tabel

- Jika tidak → Tampilkan peringatan

2. Ambil plat nomor dari baris terpilih

3. Cari kendaraan di daftarKendaraan

- Filter berdasarkan plat nomor dan status tersedia
- Jika tidak ditemukan → Tampilkan error

4. Minta input nama penyewa

- Jika dibatalkan/kosong → Batal sewa

5. Jika nama valid:

- Set status kendaraan = tidak tersedia
- Buat objek Penyewaan baru
- Simpan ke penyewaanAktif
- Tampilkan pesan sukses
- Refresh tampilan tabel

}

Proses Pengembalian

kembalikanKendaraan() {

1. Cek apakah ada baris terpilih di tabel

- Jika tidak → Tampilkan peringatan

2. Ambil plat nomor dari baris terpilih

3. Cari & hapus data penyewaan dari penyewaanAktif

- Jika tidak ditemukan → Tampilkan error

4. Jika ditemukan:

- Set status kendaraan = tersedia
- Hitung durasi sewa (waktuSekarang - waktuMulai)
- Hitung biaya = ceil(durasiJam/24) * tarifPerHari

- Tampilkan info biaya
- Refresh tampilan tabel

}

Perhitungan Biaya

- Durasi dihitung dalam jam
- Dibulatkan ke atas ke hari penuh
- $\text{Biaya} = \text{jumlahHari} \times \text{tarifPerHari}$

Contoh:

- Sewa 25 jam dengan tarif 100rb/hari
- $\text{ceil}(25/24) = 2 \text{ hari}$
- $\text{Biaya} = 2 \times 100\text{rb} = 200\text{rb}$

Fitur Keamanan

- Tabel tidak bisa diedit langsung
- Validasi input pada setiap operasi
- Pengecekan status sebelum operasi
- Penanganan kasus-kasus error

Manajemen Data

- Data kendaraan disimpan di ArrayList
- Penyewaan aktif disimpan di HashMap dengan key = platNomor
- Status tersedia/tidak langsung diupdate di objek Kendaraan
- Setiap perubahan data diikuti refresh tampilan

OUTPUT

Plat Nomor	Jenis	Tarif per Hari	Status
B 1234 CD	Mobil	Rp 300.000	Tersedia
B 5678 EF	Motor	Rp 100.000	Tersedia
B 9012 GH	Mobil	Rp 350.000	Tersedia