

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);
    });
}
}

```

Output

Sistem Rental Kendaraan			
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