

Nama : Dewa Pargusha

NIM : 221011400469

Kelas : 06TPLE006

Berikut Kodenya :

```
import javax.swing.*;

import javax.swing.table.DefaultTableModel;

import javax.swing.table.TableCellRenderer;

import java.awt.*;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import java.util.LinkedList;

import java.util.Queue;

import java.util.Collections;

import java.util.Comparator;

import java.util.List;

import java.util.ArrayList;


public class AplikasiManajemenAntrianPasien extends JFrame {


    private AntrianPasien antrianPasien = new AntrianPasien();

    private DefaultTableModel tableModel;


    // Komponen UI

    private JTextField namaField;
```

```
private JTextField usiaField;

private JComboBox<String> kondisiComboBox;

private JTable tabelPasien;

private JLabel infoLabel;

private JLabel jumlahPasienLabel;


public AplikasiManajemenAntrianPasien() {

    // Konfigurasi frame utama

    setTitle("Manajemen Antrian Pasien - Rumah Sakit Sejahtera");

    setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

    setSize(1000, 650);

    setLocationRelativeTo(null);


    // Layout utama

    JPanel mainPanel = new JPanel(new BorderLayout(10, 10));

    mainPanel.setBorder(BorderFactory.createEmptyBorder(15, 15, 15, 15));

    mainPanel.setBackground(new Color(240, 240, 240));


    // Header

    JPanel headerPanel = new JPanel();

    headerPanel.setLayout(new BoxLayout(headerPanel, BoxLayout.Y_AXIS));

    headerPanel.setBackground(new Color(240, 240, 240));


    JLabel titleLabel = new JLabel("MANAJEMEN ANTRIAN PASIEN");

    titleLabel.setFont(new Font("Arial", Font.BOLD, 24));
```

```
titleLabel.setForeground(new Color(0, 0, 139)); // Dark blue
titleLabel.setAlignmentX(Component.CENTER_ALIGNMENT);

JLabel subtitleLabel = new JLabel("Rumah Sakit Sejahtera");
subtitleLabel.setFont(new Font("Arial", Font.PLAIN, 14));
subtitleLabel.setForeground(Color.GRAY);
subtitleLabel.setAlignmentX(Component.CENTER_ALIGNMENT);

headerPanel.add(titleLabel);
headerPanel.add(Box.createRigidArea(new Dimension(0, 5)));
headerPanel.add(subtitleLabel);

mainPanel.add(headerPanel, BorderLayout.NORTH);

// Content - Form dan Tabel
JPanel contentPanel = new JPanel(new GridLayout(1, 2, 20, 0));
contentPanel.setBackground(new Color(240, 240, 240));

// Panel Form
JPanel formPanel = new JPanel();
formPanel.setLayout(new BoxLayout(formPanel, BoxLayout.Y_AXIS));
formPanel.setBorder(BorderFactory.createCompoundBorder(
    BorderFactory.createTitledBorder("TAMBAH PASIEN BARU"),
    BorderFactory.createEmptyBorder(10, 10, 10, 10)
));
```

```
formPanel.setBackground(Color.WHITE);

formPanel.setPreferredSize(new Dimension(300, 400));


// Komponen form

JPanel formGrid = new JPanel(new GridLayout(3, 2, 10, 15));

formGrid.setBackground(Color.WHITE);


JLabel namaLabel = new JLabel("Nama Pasien:");

namaLabel.setFont(new Font("Arial", Font.BOLD, 12));

namaField = new JTextField();

namaField.setBorder(BorderFactory.createCompoundBorder(

    BorderFactory.createLineBorder(new Color(200, 200, 200)),

    BorderFactory.createEmptyBorder(5, 5, 5, 5)

));


JLabel usiaLabel = new JLabel("Usia:");

usiaLabel.setFont(new Font("Arial", Font.BOLD, 12));

usiaField = new JTextField();

usiaField.setBorder(BorderFactory.createCompoundBorder(

    BorderFactory.createLineBorder(new Color(200, 200, 200)),

    BorderFactory.createEmptyBorder(5, 5, 5, 5)

));


JLabel kondisiLabel = new JLabel("Kondisi:");

kondisiLabel.setFont(new Font("Arial", Font.BOLD, 12));
```

```
kondisiComboBox = new JComboBox<>(new String[]{"Normal", "Darurat", "Kritis"});
```

```
formGrid.add(namaLabel);
```

```
formGrid.add(namaField);
```

```
formGrid.add(usiaLabel);
```

```
formGrid.add(usiaField);
```

```
formGrid.add(kondisiLabel);
```

```
formGrid.add(kondisiComboBox);
```

```
// Tombol Tambah
```

```
JButton tambahButton = new JButton("TAMBAH PASIEN");
```

```
tambahButton.setFont(new Font("Arial", Font.BOLD, 12));
```

```
tambahButton.setBackground(new Color(0, 0, 255)); // Green
```

```
tambahButton.setForeground(Color.BLACK);
```

```
tambahButton.setBorder(BorderFactory.createEmptyBorder(10, 0, 10, 0));
```

```
tambahButton.setAlignmentX(Component.CENTER_ALIGNMENT);
```

```
tambahButton.addActionListener(e -> handleTambahPasien());
```

```
formPanel.add(formGrid);
```

```
formPanel.add(Box.createRigidArea(new Dimension(0, 20)));
```

```
formPanel.add(tambahButton);
```

```
// Panel Tabel
```

```
JPanel tablePanel = new JPanel(new BorderLayout());
```

```
tablePanel.setBorder(BorderFactory.createCompoundBorder(
```

```

        BorderFactory.createTitledBorder("DAFTAR ANTRIAN PASIEN"),

        BorderFactory.createEmptyBorder(10, 10, 10, 10)

    ));

tablePanel.setBackground(Color.WHITE);

// Model tabel

tableModel = new DefaultTableModel() {

    @Override

    public boolean isCellEditable(int row, int column) {

        return false; // Membuat tabel tidak bisa di-edit

    }

};

tableModel.addColumn("NAMA PASIEN");

tableModel.addColumn("USIA");

tableModel.addColumn("KONDISI");

tableModel.addColumn("PRIORITAS");

tabelPasien = new JTable(tableModel) {

    // Warna baris berdasarkan prioritas

    @Override

    public Component prepareRenderer(TableCellRenderer renderer, int row, int column) {

        Component c = super.prepareRenderer(renderer, row, column);

        String prioritas = (String) getValueAt(row, 3);

        if (prioritas.equals("Tinggi")) {

```

```

        c.setBackground(new Color(255, 205, 210)); // Light red
    } else if (prioritas.equals("Sedang")) {
        c.setBackground(new Color(255, 249, 196)); // Light yellow
    } else {
        c.setBackground(new Color(200, 230, 201)); // Light green
    }

    if (column == 3) { // Kolom prioritas
        c.setFont(c.getFont().deriveFont(Font.BOLD));
    }

    return c;
}

};

tabelPasien.setFont(new Font("Arial", Font.PLAIN, 12));
tabelPasien.setRowHeight(25);
tabelPasien.getTableHeader().setFont(new Font("Arial", Font.BOLD, 12));

JScrollPane scrollPane = new JScrollPane(tabelPasien);
tablePanel.add(scrollPane, BorderLayout.CENTER);

// Panel Tombol Aksi

JPanel actionPanel = new JPanel(new FlowLayout(FlowLayout.CENTER, 10, 10));
actionPanel.setBackground(Color.WHITE);

```

```
JButton panggilButton = new JButton("PANGGIL PASIEN BERIKUTNYA");
```

```
styleButton(panggilButton, new Color(255, 255, 250)); // Black
```

```
JButton urutUsiaButton = new JButton("URUTKAN BERDASARKAN USIA");
```

```
styleButton(urutUsiaButton, new Color(0, 0, 255)); // Black
```

```
JButton urutPrioritasButton = new JButton("URUTKAN BERDASARKAN PRIORITAS");
```

```
styleButton(urutPrioritasButton, new Color(0, 0, 255)); // Black
```

```
panggilButton.addActionListener(e -> handlePanggilPasien());
```

```
urutUsiaButton.addActionListener(e -> handleUrutkanUsia());
```

```
urutPrioritasButton.addActionListener(e -> handleUrutkanPrioritas());
```

```
actionPanel.add(panggilButton);
```

```
actionPanel.add(urutUsiaButton);
```

```
actionPanel.add(urutPrioritasButton);
```

```
// Panel Info
```

```
JPanel infoPanel = new JPanel(new FlowLayout(FlowLayout.LEFT, 10, 5));
```

```
infoPanel.setBackground(Color.white);
```

```
infoLabel = new JLabel("");
```

```
infoLabel.setFont(new Font("Arial", Font.BOLD, 12));
```

```
infoLabel.setForeground(new Color(233, 30, 99)); // Pink
```



```
jumlahPasienLabel = new JLabel("Jumlah Pasien: 0");

jumlahPasienLabel.setFont(new Font("Arial", Font.BOLD, 12));


infoPanel.add(infoLabel);

infoPanel.add(jumlahPasienLabel);


// Gabungkan komponen tabel

JPanel tableContainer = new JPanel(new BorderLayout());

tableContainer.add(tablePanel, BorderLayout.CENTER);

tableContainer.add(actionPanel, BorderLayout.SOUTH);

tableContainer.add(infoPanel, BorderLayout.NORTH);


contentPanel.add(formPanel);

contentPanel.add(tableContainer);


mainPanel.add(contentPanel, BorderLayout.CENTER);


// Footer

JLabel footerLabel = new JLabel("© 2023 Rumah Sakit Sejahtera - Sistem Manajemen Antrian
Pasien");

footerLabel.setFont(new Font("Arial", Font.PLAIN, 11));

footerLabel.setForeground(Color.GRAY);

footerLabel.setHorizontalAlignment(SwingConstants.CENTER);


mainPanel.add(footerLabel, BorderLayout.SOUTH);
```

```
add(mainPanel);  
}
```

```
private void styleButton(JButton button, Color color) {  
    button.setFont(new Font("Arial", Font.BOLD, 12));  
    button.setBackground(color);  
    button.setForeground(Color.BLACK);  
    button.setFocusPainted(false);  
    button.setBorder(BorderFactory.createEmptyBorder(8, 15, 8, 15));  
}
```

```
private void handleTambahPasien() {  
    try {  
        String nama = namaField.getText().trim();  
        String usiaText = usiaField.getText().trim();  
  
        if (nama.isEmpty()) {  
            infoLabel.setText("Nama tidak boleh kosong!");  
            return;  
        }  
  
        if (usiaText.isEmpty()) {  
            infoLabel.setText("Usia tidak boleh kosong!");  
            return;  
        }  
    }  
}
```

```
}
```

```
int usia = Integer.parseInt(usiaText);
```

```
String kondisi = (String) kondisiComboBox.getSelectedItem();
```

```
if (usia <= 0 || usia > 120) {
```

```
    infoLabel.setText("Usia harus antara 1-120 tahun!");
```

```
    return;
```

```
}
```

```
Pasien pasienBaru = new Pasien(nama, usia, kondisi);
```

```
antrianPasien.tambahPasien(pasienBaru);
```

```
refreshTabel();
```

```
namaField.setText("");
```

```
usiaField.setText("");
```

```
infoLabel.setText("Pasien " + nama + " telah ditambahkan ke antrian.");
```

```
jumlahPasienLabel.setText("Jumlah Pasien: " + antrianPasien.getJumlahPasien());
```

```
} catch (NumberFormatException e) {
```

```
    infoLabel.setText("Usia harus berupa angka!");
```

```
}
```

```
}
```

```
private void handlePanggilPasien() {
```

```
    Pasien pasien = antrianPasien.panggilPasienBerikutnya();
```

```
if (pasien != null) {  
    infoLabel.setText("Memanggil pasien: " + pasien.getNama());  
    refreshTabel();  
    jumlahPasienLabel.setText("Jumlah Pasien: " + antrianPasien.getJumlahPasien());  
} else {  
    infoLabel.setText("Tidak ada pasien dalam antrian.");  
}  
}
```

```
private void handleUrutkanUsia() {  
    antrianPasien.urutkanBerdasarkanUsia();  
    refreshTabel();  
    infoLabel.setText("Antrian telah diurutkan berdasarkan usia (tertua ke termuda).");  
}
```

```
private void handleUrutkanPrioritas() {  
    antrianPasien.urutkanBerdasarkanPrioritas();  
    refreshTabel();  
    infoLabel.setText("Antrian telah diurutkan berdasarkan prioritas.");  
}
```

```
private void refreshTabel() {  
    tableModel.setRowCount(0);  
  
    for (Pasien pasien : antrianPasien.getDaftarPasien()) {
```

```

        tableModel.addRow(new Object[]{
            pasien.getNama(),
            pasien.getUsia(),
            pasien.getKondisi(),
            pasien.getPrioritas()
        });
    }
}

```

```

public static void main(String[] args) {
    SwingUtilities.invokeLater(() -> {
        try {
            // Set look and feel untuk tampilan lebih modern
            UIManager.setLookAndFeel(UIManager.getSystemLookAndFeelClassName());
        } catch (Exception e) {
            e.printStackTrace();
        }

        AplikasiManajemenAntrianPasien app = new AplikasiManajemenAntrianPasien();
        app.setVisible(true);
    });
}
}

```

```

class Pasien {

```

```
private String nama;
```

```
private int usia;
```

```
private String kondisi;
```

```
private String prioritas;
```

```
public Pasien(String nama, int usia, String kondisi) {
```

```
    this.nama = nama;
```

```
    this.usia = usia;
```

```
    this.kondisi = kondisi;
```

```
    this.prioritas = hitungPrioritas(usia, kondisi);
```

```
}
```

```
private String hitungPrioritas(int usia, String kondisi) {
```

```
    if (kondisi.equalsIgnoreCase("kritis")) {
```

```
        return "Tinggi";
```

```
    } else if (usia > 60 || kondisi.equalsIgnoreCase("darurat")) {
```

```
        return "Sedang";
```

```
    } else {
```

```
        return "Rendah";
```

```
    }
```

```
}
```

```
public String getNama() {
```

```
    return nama;
```

```
}
```

```
public int getUsia() {  
    return usia;  
}  
  
public String getKondisi() {  
    return kondisi;  
}  
  
public String getPrioritas() {  
    return prioritas;  
}  
}  
  
class AntrianPasien {  
    private Queue<Pasien> antrian;  
    private Queue<Pasien> antrianPrioritas;  
  
    public AntrianPasien() {  
        antrian = new LinkedList<>();  
        antrianPrioritas = new LinkedList<>();  
    }  
  
    public void tambahPasien(Pasien pasien) {  
        if (pasien.getPrioritas().equals("Tinggi") || pasien.getPrioritas().equals("Sedang")) {
```

```
        antrianPrioritas.add(pasien);  
    } else {  
        antrian.add(pasien);  
    }  
}
```

```
public Pasien panggilPasienBerikutnya() {  
    if (!antrianPrioritas.isEmpty()) {  
        return antrianPrioritas.poll();  
    } else if (!antrian.isEmpty()) {  
        return antrian.poll();  
    }  
    return null;  
}
```

```
public List<Pasien> getDaftarPasien() {  
    List<Pasien> semuaPasien = new ArrayList<>();  
    semuaPasien.addAll(antrianPrioritas);  
    semuaPasien.addAll(antrian);  
    return semuaPasien;  
}
```

```
public void urutkanBerdasarkanUsia() {  
    List<Pasien> semuaPasien = getDaftarPasien();  
    Collections.sort(semuaPasien, Comparator.comparingInt(Pasien::getUsia).reversed());  
}
```



```
antrianPrioritas.clear();

antrian.clear();


for (Pasien pasien : semuaPasien) {

    tambahPasien(pasien);

}

}


public void urutkanBerdasarkanPrioritas() {

    List<Pasien> semuaPasien = getDaftarPasien();

    Collections.sort(semuaPasien, (p1, p2) -> {

        int prioritas1 = getNilaiPrioritas(p1.getPrioritas());

        int prioritas2 = getNilaiPrioritas(p2.getPrioritas());

        return Integer.compare(prioritas2, prioritas1);

    });

    antrianPrioritas.clear();

    antrian.clear();


    for (Pasien pasien : semuaPasien) {

        tambahPasien(pasien);

    }

}
```

```
private int getNilaiPrioritas(String prioritas) {  
    switch (prioritas) {  
        case "Tinggi": return 3;  
        case "Sedang": return 2;  
        case "Rendah": return 1;  
        default: return 0;  
    }  
}  
  
public int getJumlahPasien() {  
    return antrian.size() + antrianPrioritas.size();  
}  
}
```

Hasil Dari kodenya :

