

LAPORAN PRAKTKUM

GRAPHICAL USER INTERFACE

Dibuat untuk memenuhi salah satu tugas mata kuliah Pemrograman Berbasis
Objek yang diampu oleh Bapak Zulkifli Arsyad



Disusun Oleh:

NIM : 241511094
NAMA : ZAHRA ALDILA
KELAS : 2C
PROGRAM STUDI : D3 – TEKNIK INFORMATIKA
JURUSAN : TEKNIK KOMPUTER DAN INFORMATIKA

Program Studi D-3 Teknik Informatika
Jurusan Teknik Komputer dan Informatika
Politeknik Negeri Bandung
2025

DAFTAR ISI

DAFTAR ISI	i
DAFTAR GAMBAR	ii
A. Pengertian	1
1. Graphical User Interface	1
2. Java Swing	1
B. Pengerjaan.....	2
1. Deskripsi Program	2
2. Tampilan UI	2
3. Source Code	3
C. Lesson Learned	9

DAFTAR GAMBAR

Gambar 1. Tampilan UI	2
-----------------------------	---

A. Pengertian

1. Graphical User Interface

Graphical User Interface (GUI) adalah antarmuka pengguna yang memungkinkan interaksi antara manusia dan komputer melalui elemen-elemen grafis seperti tombol, jendela, ikon, menu, dan teks. Dengan GUI, pengguna tidak perlu mengetikkan perintah secara manual di command line, tetapi cukup menggunakan mouse dan keyboard untuk mengoperasikan aplikasi.

GUI berfungsi untuk mempermudah penggunaan program sehingga lebih intuitif dan efisien. Dalam konteks pemrograman, GUI dibangun menggunakan komponen visual (seperti tombol, label, dan kotak teks) yang dapat diatur dan diatur perilakunya melalui kode program.

Ciri-ciri utama GUI antara lain:

- a. Menggunakan elemen visual seperti window, menu, icon, dan button.
- b. Memungkinkan interaksi secara langsung dengan elemen tersebut (event-driven).
- c. Memberikan tampilan yang interaktif dan mudah digunakan.

2. Java Swing

Java Swing adalah salah satu library GUI di bahasa pemrograman Java yang disediakan dalam paket javax.swing. Swing digunakan untuk membuat antarmuka grafis berbasis desktop (desktop application) yang berjalan di berbagai sistem operasi karena sifat Java yang platform independent.

Swing merupakan pengembangan dari Abstract Window Toolkit (AWT) dengan tampilan dan fitur yang lebih lengkap. Komponen-komponen dalam Swing seperti JFrame, JPanel, JButton, JLabel, JTable, dan JTextField dapat digunakan untuk membangun jendela aplikasi yang interaktif.

Keunggulan Java Swing:

- a. Memiliki banyak komponen siap pakai untuk membangun antarmuka.
- b. Dapat dikustomisasi (ubah warna, ukuran, font, dan tata letak).
- c. Mendukung berbagai layout manager seperti BorderLayout, FlowLayout, dan GridLayout.

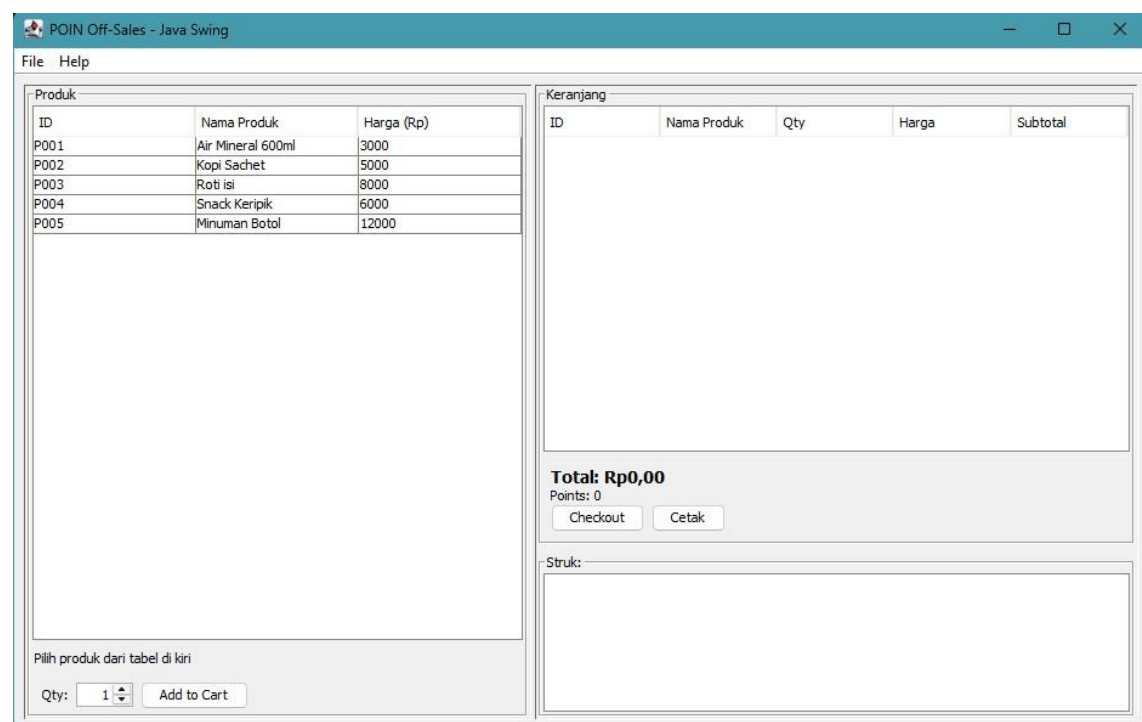
B. Pengerjaan

1. Deskripsi Program

Program ini merupakan tampilan awal aplikasi Point of Sales (POS) yang dibuat menggunakan Java Swing. Antarmuka dirancang menggunakan komponen seperti JFrame, JPanel, JTable, JButton, JLabel, dan JTextArea untuk menampilkan daftar produk, keranjang belanja, serta area struk. Layout disusun dengan kombinasi BorderLayout dan FlowLayout agar tampilan rapi dan menyerupai sistem kasir pada umumnya.

Saat ini, program hanya berfokus pada pembuatan antarmuka grafis (GUI) tanpa logika perhitungan atau transaksi. Tujuannya adalah untuk melatih pemahaman mahasiswa dalam menempatkan dan mengatur komponen GUI di dalam jendela aplikasi Java sebelum ditambahkan fungsi interaktif seperti “Add to Cart”, “Checkout”, dan “Cetak”.

2. Tampilan UI



Gambar 1. Tampilan UI

3. Source Code

```
//POSFrame.java

import javax.swing.*;
import javax.swing.table.DefaultTableModel;
import java.awt.*;

public class POSFrame extends JFrame {

    // daftar produk
    private JTable tblProduk;
    private JSpinner spQty;
    private JButton btnAddToCart;

    // keranjang & ringkasan
    private JTable tblKeranjang;
    private JLabel lblTotal;
    private JLabel lblPoints;
    private JButton btnCheckout;
    private JButton btnCetak;
    private JTextArea txtStruk;

    public POSFrame() {
        super("POIN Off-Sales - Java Swing");
        setDefaultCloseOperation(EXIT_ON_CLOSE);
        setMinimumSize(new Dimension(950, 600));
        setLocationRelativeTo(null);

        setJMenuBar(createMenuBar());
        setContentPane(buildContent());
    }

    private JMenuBar createMenuBar() {
```

```

JMenuBar bar = new JMenuBar();
JMenu mFile = new JMenu("File");
JMenu mHelp = new JMenu("Help");
bar.add(mFile);
bar.add(mHelp);
return bar;
}

private JPanel buildContent() {
    JPanel root = new JPanel(new BorderLayout(8, 8));
    root.setBorder(BorderFactory.createEmptyBorder(8, 8, 8, 8));

    JSplitPane split = new JSplitPane(JSplitPane.HORIZONTAL_SPLIT,
        buildProdukPanel(), buildRightPanel());
    split.setResizeWeight(0.4);
    root.add(split, BorderLayout.CENTER);

    return root;
}

private JPanel buildProdukPanel() {
    JPanel panel = new JPanel(new BorderLayout(6, 6));
    panel.setBorder(BorderFactory.createTitledBorder("Produk"));

    // Tabel produk
    String[] cols = {"ID", "Nama Produk", "Harga (Rp)"};
    DefaultTableModel modelProduk = new DefaultTableModel(cols, 0) {
        @Override public boolean isCellEditable(int r, int c) { return false; }
    };
    // Data dummy
    modelProduk.addRow(new Object[]{"P001", "Air Mineral 600ml",
3000});
    modelProduk.addRow(new Object[]{"P002", "Kopi Sachet", 5000});
}

```

```

        modelProduk.addRow(new Object[]{"P003", "Roti isi", 8000});
        modelProduk.addRow(new Object[]{"P004", "Snack Keripik", 6000});
        modelProduk.addRow(new Object[]{"P005", "Minuman Botol",
12000});

        tblProduk = new JTable(modelProduk);

        tblProduk.setSelectionMode(ListSelectionModel.SINGLE_SELECTION);
        tblProduk.setFillsViewportHeight(true);
        panel.add(new JScrollPane(tblProduk), BorderLayout.CENTER);

        // pilih qty + tombol Add to Cart
        JPanel bottom = new JPanel(new BorderLayout(6, 6));
        bottom.setBorder(BorderFactory.createEmptyBorder(2, 2, 2, 2));

        JLabel lblInstruksi = new JLabel("Pilih produk dari tabel di kiri");
        bottom.add(lblInstruksi, BorderLayout.NORTH);

        JPanel action = new JPanel(new FlowLayout(FlowLayout.LEFT, 6, 0));
        action.add(new JLabel("Qty:"));
        spQty = new JSpinner(new SpinnerNumberModel(1, 1, 999, 1));
        ((JSpinner.DefaultEditor)
spQty.getEditor()).getTextField().setColumns(3);
        action.add(spQty);

        btnAddToCart = new JButton("Add to Cart");
        action.add(btnAddToCart);
        bottom.add(action, BorderLayout.SOUTH);

        panel.add(bottom, BorderLayout.SOUTH);
        return panel;
    }

```



```

private JPanel buildRightPanel() {
    JPanel panel = new JPanel(new BorderLayout(6, 6));

    // Keranjang + Ringkasan + Struk
    panel.add(buildKeranjangPanel(), BorderLayout.CENTER);
    panel.add(buildStrukPanel(), BorderLayout.SOUTH);

    return panel;
}

private JComponent buildKeranjangPanel() {
    JPanel wrapper = new JPanel(new BorderLayout(6, 6));
    wrapper.setBorder(BorderFactory.createTitledBorder("Keranjang"));

    // Tabel keranjang
    String[] cols = {"ID", "Nama Produk", "Qty", "Harga", "Subtotal"};
    DefaultTableModel modelCart = new DefaultTableModel(cols, 0) {
        @Override public boolean isCellEditable(int r, int c) { return false; }
        @Override public Class<?> getColumnClass(int c) {
            return switch (c) {
                case 2, 3, 4 -> Integer.class;
                default -> String.class;
            };
        }
    };

    tblKeranjang = new JTable(modelCart);
    tblKeranjang.setFillsViewportHeight(true);
    wrapper.add(new JScrollPane(tblKeranjang), BorderLayout.CENTER);

    // total + tombol
    wrapper.add(buildSummaryPanel(), BorderLayout.SOUTH);

    return wrapper;
}

```

```

    }

    private JPanel buildSummaryPanel() {
        JPanel p = new JPanel();
        p.setLayout(new BoxLayout(p, BoxLayout.Y_AXIS));
        p.setBorder(BorderFactory.createEmptyBorder(6, 0, 6, 0));

        lblTotal = new JLabel("Total: Rp0,00");
        lblTotal.setFont(lblTotal.getFont().deriveFont(Font.BOLD, 14f));
        lblPoints = new JLabel("Points: 0");

        JPanel line1 = new JPanel(new FlowLayout(FlowLayout.LEFT, 6, 0));
        line1.add(lblTotal);
        JPanel line2 = new JPanel(new FlowLayout(FlowLayout.LEFT, 6, 0));
        line2.add(lblPoints);

        JPanel buttons = new JPanel(new FlowLayout(FlowLayout.LEFT, 6,
0));
        btnCheckout = new JButton("Checkout");
        btnCetak = new JButton("Cetak");
        buttons.add(btnCheckout);
        buttons.add(btnCetak);

        p.add(line1);
        p.add(line2);
        p.add(buttons);
        return p;
    }

    private JPanel buildStrukPanel() {
        JPanel p = new JPanel(new BorderLayout(6, 6));
        p.setBorder(BorderFactory.createTitledBorder("Struk:"));
        txtStruk = new JTextArea(6, 30);
    }

```

```

        txtStruk.setEditable(false);
        txtStruk.setLineWrap(true);
        txtStruk.setWrapStyleWord(true);
        p.add(new JScrollPane(txtStruk), BorderLayout.CENTER);
        return p;
    }

    // ===== MAIN =====

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

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

```

C. Lesson Learned

Melalui praktikum ini, saya belajar bagaimana membangun antarmuka grafis (Graphical User Interface) menggunakan Java Swing. Saya belajar fungsi dari berbagai komponen seperti JFrame sebagai jendela utama, JPanel sebagai wadah tata letak, serta komponen interaktif seperti JButton, JLabel, JTable, dan JTextArea yang digunakan untuk menampilkan data serta menerima input pengguna.

Selain itu, saya juga mempelajari cara menggunakan layout manager seperti BorderLayout dan FlowLayout untuk mengatur posisi dan ukuran komponen agar tampilan terlihat rapi serta mudah dipahami pengguna. Pembuatan tampilan UI ini memberikan gambaran nyata tentang pentingnya desain antarmuka sebelum menambahkan logika program atau fungsionalitas lainnya.

Dari praktikum ini, saya menyadari bahwa pembuatan GUI tidak hanya berfokus pada kode, tetapi juga pada kemudahan penggunaan (usability) dan keteraturan desain agar aplikasi yang dibuat dapat digunakan dengan nyaman dan efisien. Pengalaman ini menjadi dasar penting untuk pengembangan aplikasi Java yang lebih kompleks pada tahap berikutnya.