

Laporan Praktikum 10 - Event-Driven Programming (Event Handling)

Nama : Cania Nabilatul Adawah

NIM : 2403102

Kelas : D3TI2C

Mata Kuliah : Pemrograman Berorientasi Objek

1. Latihan event handling Counter App

a. Kode Program

```
import javax.swing.JButton;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JPanel;
import javax.swing.JScrollPane;
import javax.swing.SwingUtilities;
import javax.swing.UIManager;

import java.awt.Dimension;
import java.awt.Font;

import com.formdev.flatlaf.themes.FlatMacLightLaf;

import net.miginfocom.swing.MigLayout;

public class LatihanCounterApp {

    private static int nilaiCounter = 0;

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

        SwingUtilities.invokeLater(() -> {
```

```

JFrame frame = new JFrame("Counter App");
frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
frame.setPreferredSize(new Dimension(400, 300));
frame.setMinimumSize(new Dimension(200, 150));

// Layout dengan MigLayout
JPanel panelCounter = new JPanel(new MigLayout(
    "fill, wrap 1",
    "[grow, center]",
    "[grow] [grow]"
));

JLabel labelCounter = new JLabel(String.valueOf(nilaiCounter));
labelCounter.setFont(new Font("Arial", Font.BOLD, 50));

// POSISI bottom pada row 1
panelCounter.add(labelCounter, "bottom");

JButton buttonAdd = new JButton("+");
buttonAdd.setFont(new Font("Arial", Font.BOLD, 25));

// POSISI top pada row 2
panelCounter.add(buttonAdd, "top");

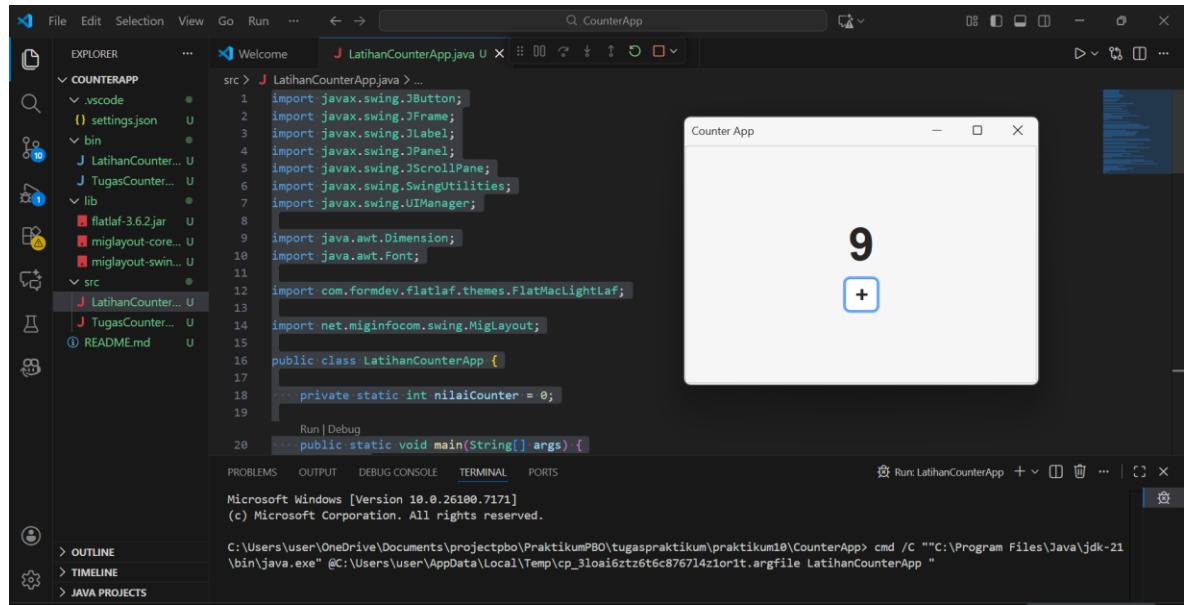
buttonAdd.addActionListener(e -> {
    nilaiCounter++;
    labelCounter.setText(String.valueOf(nilaiCounter));
});

JScrollPane scrollPane = new JScrollPane(panelCounter);
scrollPane.getVerticalScrollBar().setUnitIncrement(10);
scrollPane.getVerticalScrollBar()
    .putClientProperty("JScrollBar.fastWheelScrolling", true);

frame.add(scrollPane);
frame.pack();
frame.setLocationRelativeTo(null);
frame.setVisible(true);
});
}
}

```

- b. Hasil Running



2. Tugas event handling Counter App

a. Kode Program

```
import javax.swing.JButton;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JPanel;
import javax.swing.JScrollPane;
import javax.swing.SwingUtilities;
import javax.swing.UIManager;

import java.awt.CardLayout;
import java.awt.Dimension;
import java.awt.Font;

import com.formdev.flatlaf.themes.FlatMacLightLaf;
import net.miginfocom.swing.MigLayout;

public class TugasCounterApp {

    // variabel static untuk menyimpan nilai counter yang akan ditampilkan
    // di layar
    private static int nilaiCounter = 0;

    public static void main(String[] args) {
        try {
            // Mengganti tampilan default Swing dengan tema FlatLaf
            (FlatMacLightLaf)
```

```

        // Agar tampilan tombol, frame, dsb jadi lebih modern.
        UIManager.setLookAndFeel(new FlatMacLightLaf());
    } catch (Exception e) {
        // Jika gagal set look and feel, cetak error-nya di konsol, tapi
        program tetap dijalankan.
        e.printStackTrace();
    }

    // SwingUtilities.invokeLater() memastikan kode GUI dijalankan di
    Event Dispatch Thread (EDT)
    // Ini adalah best practice ketika bekerja dengan komponen Swing.
    SwingUtilities.invokeLater(() -> {

        // Membuat window utama aplikasi dengan judul "Counter App"
        JFrame frame = new JFrame("Counter App");
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        // ketika di-close, aplikasi berhenti
        frame.setPreferredSize(new Dimension(430, 300));      // ukuran
        awal jendela ketika pertama dibuka
        frame.setMinimumSize(new Dimension(250, 150));        // ukuran minimum ketika jendela di-resize

        // Panel "cards" akan berfungsi sebagai container utama yang
        memakai CardLayout
        // CardLayout memungkinkan kita berganti "halaman"
        (panelCounter <-> panelSummary)
        JPanel cards = new JPanel(new CardLayout());

        //

=====

====

        // PANEL 1 - HALAMAN COUNTER
        //

=====

====

        // panelCounter adalah panel untuk halaman utama yang
        menampilkan angka dan tombol - / +
        // MigLayout("fill, wrap 1", "[grow, center]", "[grow] [grow]
        [grow]")
        // - "fill" : panel diisi penuh area yang tersedia
        // - "wrap 1" : setiap baris hanya satu kolom (satu komponen per
        baris)
        // - "[grow, center]" : kolom tunggal dengan lebar fleksibel (grow)
        dan align center

```

```

    // - "[grow] [grow] [grow]" : 3 baris, masing-masing mempunyai
    tinggi fleksibel
    JPanel panelCounter = new JPanel(new MigLayout(
        "fill, wrap 1",
        "[grow, center]",
        "[grow] [grow] [grow]"
    ));

    // =====
    // Baris 1: Label angka
    // =====

    // labelCounter menampilkan nilai counter (angka di tengah layar)
    JLabel labelCounter = new JLabel(String.valueOf(nilaiCounter));
    labelCounter.setFont(new Font("Arial", Font.BOLD, 50)); // ukuran font besar agar jelas terlihat

    // "bottom" → komponen ini akan ditempatkan di bagian bawah
    // baris pertama.
    // Dengan baris pertama bertipe [grow], efeknya: angka berada
    // sedikit di bawah tengah area baris atas.
    panelCounter.add(labelCounter, "bottom");

    // =====
    // Baris 2: Tombol - dan +
    // =====

    // panelButtons adalah panel kecil untuk menampung dua tombol
    // (- dan +)
    // "insets 0" → tanpa padding dalam
    // "[]15[]" → dua kolom, di antara keduanya ada jarak horizontal
    // 15 pixel
    JPanel panelButtons = new JPanel(new MigLayout("insets 0",
        "[]15[]"));

    // Tombol untuk mengurangi nilai counter
    JButton btnMin = new JButton("-");
    // Tombol untuk menambah nilai counter
    JButton btnPlus = new JButton("+");

    // Set font kedua tombol agar tampilannya konsisten
    btnMin.setFont(new Font("Arial", Font.BOLD, 25));
    btnPlus.setFont(new Font("Arial", Font.BOLD, 25));

    // Tambahkan tombol - dan + ke panelButtons secara berurutan
    panelButtons.add(btnMin);

```

```

panelButtons.add(btnPlus);

        // Tambahkan panelButtons ke baris kedua panelCounter.
        // "top" → panelButtons akan ditempatkan di bagian atas baris
        // kedua.
        panelCounter.add(panelButtons, "top");

        // =====
        // Baris 3: Tombol >
        // =====

        // Tombol untuk berpindah ke halaman berikutnya (summary)
        JButton btnNext = new JButton(">");
        btnNext.setFont(new Font("Arial", Font.BOLD, 22)); // sedikit
        lebih kecil dari angka
        btnNext.setPreferredSize(new Dimension(40, 40)); // ukuran
        kotak tombol

        // Tambahkan btnNext ke panelCounter dengan constraint pos
        // "pos 1.0al 1.0al" artinya:
        // - 1.0al pada sumbu X → posisi di sisi paling kanan (alignment
        1.0 = kanan)
        // - 1.0al pada sumbu Y → posisi di sisi paling bawah (alignment
        1.0 = bawah)
        // "gapright 15" → beri jarak 15 pixel dari sisi kanan panel
        // "gapbottom 12" → beri jarak 12 pixel dari sisi bawah panel
        // Hasilnya: tombol > berada di pojok kanan bawah, tapi tidak
        menempel langsung ke border.
        panelCounter.add(btnNext, "pos 1.0al 1.0al, gapright 15,
        gapbottom 12");

        // Bungkus panelCounter di dalam JScrollPane agar sesuai dengan
        contoh tugas
        // dan memiliki border tipis di dalam frame.
        JScrollPane scroll1 = new JScrollPane(panelCounter);
        // setBorder(EtchedBorder) membuat border tipis abu-abu seperti
        panel contoh

scroll1.setBorder(javax.swing.BorderFactory.createEtchedBorder());

        //
=====

=====
// PANEL 2 - HALAMAN SUMMARY

```

```

// =====
==

    // panelSummary adalah panel halaman kedua yang menampilkan
    // teks keterangan + nilai terakhir
    JPanel panelSummary = new JPanel(new MigLayout(
        "fill, wrap 1",
        "[grow, center]",
        "[grow] [grow] [grow]"
    ));

    // ===== Baris 1: Teks + Angka (dalam satu blok) =====

    // summaryBlock adalah panel kecil untuk mengelompokkan
    // label teks dan angka
    // "wrap 1" → satu kolom (teks di atas, angka di bawah)
    // "insets 0" → tanpa padding
    // "[center]" → isi rata tengah
    // "[]5[]" → dua baris, jarak vertikal 5 pixel di antara mereka
    JPanel summaryBlock = new JPanel(new MigLayout("wrap 1",
    insets 0, "[center]", "[]5[]"));
    summaryBlock.setOpaque(false); // transparent, supaya warna
    latar belakang mengikuti panelSummary

    // Label teks yang menjelaskan arti angka
    JLabel lblInfo = new JLabel("Nilai terakhir Anda adalah");
    lblInfo.setFont(new Font("Arial", Font.PLAIN, 15));
    summaryBlock.add(lblInfo); // baris pertama di summaryBlock

    // Label angka yang menampilkan nilaiCounter ketika pindah ke
    // halaman summary
    JLabel lblResult = new JLabel(String.valueOf(nilaiCounter));
    lblResult.setFont(new Font("Arial", Font.BOLD, 50));
    summaryBlock.add(lblResult); // baris kedua di summaryBlock

    // Tambahkan summaryBlock ke panelSummary di baris pertama.
    // "bottom" → summaryBlock diletakkan di bagian bawah baris
    // pertama (yang bertipe [grow]),
    // sehingga posisinya sedikit turun dari tengah, seperti contoh
    // gambar.
    panelSummary.add(summaryBlock, "bottom");

    // ===== Baris 3: Tombol Kembali (<) =====

    // Tombol untuk kembali ke halaman utama (counter)

```

```

JButton btnBack = new JButton("<");
btnBack.setFont(new Font("Arial", Font.BOLD, 22));
btnBack.setPreferredSize(new Dimension(40, 40));

// Tambahkan btnBack ke panelSummary.
// "pos 0.0al 1.0al" → pojok kiri bawah (X=0.0 = kiri, Y=1.0 =
bawah)
// "gapleft 15" → jarak 15 pixel dari sisi kiri
// "gapbottom 12" → jarak 12 pixel dari sisi bawah
// Hasilnya: tombol < berada di pojok kiri bawah panel summary
sesuai contoh.
panelSummary.add(btnBack, "pos 0.0al 1.0al, gapleft 15,
gapbottom 12");

// Bungkus panelSummary ke dalam JScrollPane juga, supaya
konsisten dengan halaman pertama
JScrollPane scroll2 = new JScrollPane(panelSummary);

scroll2.setBorder(javax.swing.BorderFactory.createEtchedBorder());

// =====
=====  

=====  

// CARD SWITCHING (LOGIKA GANTI HALAMAN)
// =====
=====  

=====  

// Tambahkan kedua scrollPane ke panel "cards" dengan nama
kartu "main" dan "summary"
cards.add(scroll1, "main"); // halaman utama (counter)
cards.add(scroll2, "summary"); // halaman ringkas (summary)

// Ambil referensi CardLayout dari panel cards
CardLayout cl = (CardLayout) cards.getLayout();

// ===== EVENT HANDLER TOMBOL + =====
btnPlus.addActionListener(e -> {
    // Tambahkan nilai counter
    nilaiCounter++;
    // Update teks labelCounter di halaman utama
    labelCounter.setText(String.valueOf(nilaiCounter));
});

// ===== EVENT HANDLER TOMBOL - =====
btnMin.addActionListener(e -> {

```

```

        // Kurangi nilai counter
        nilaiCounter--;
        // Update teks labelCounter di halaman utama
        labelCounter.setText(String.valueOf(nilaiCounter));
    });

    // ===== EVENT HANDLER TOMBOL > (NEXT) =====
    btnNext.addActionListener(e -> {
        // Set label di halaman summary agar menampilkan nilai
        counter yang terbaru
        lblResult.setText(String.valueOf(nilaiCounter));
        // Perintahkan CardLayout untuk menampilkan kartu dengan
        nama "summary"
        cl.show(cards, "summary");
    });

    // ===== EVENT HANDLER TOMBOL < (BACK) =====
    btnBack.addActionListener(e -> {
        // Kembali ke halaman utama (counter)
        cl.show(cards, "main");
    });

    //

=====
===
    // SETUP FRAME UTAMA
    //
=====

    // Tambahkan panel cards (yang berisi kedua halaman) ke frame
    utama
    frame.add(cards);

    // pack() akan menyesuaikan ukuran frame berdasarkan
    preferredSize komponen-komponen di dalamnya
    frame.pack();

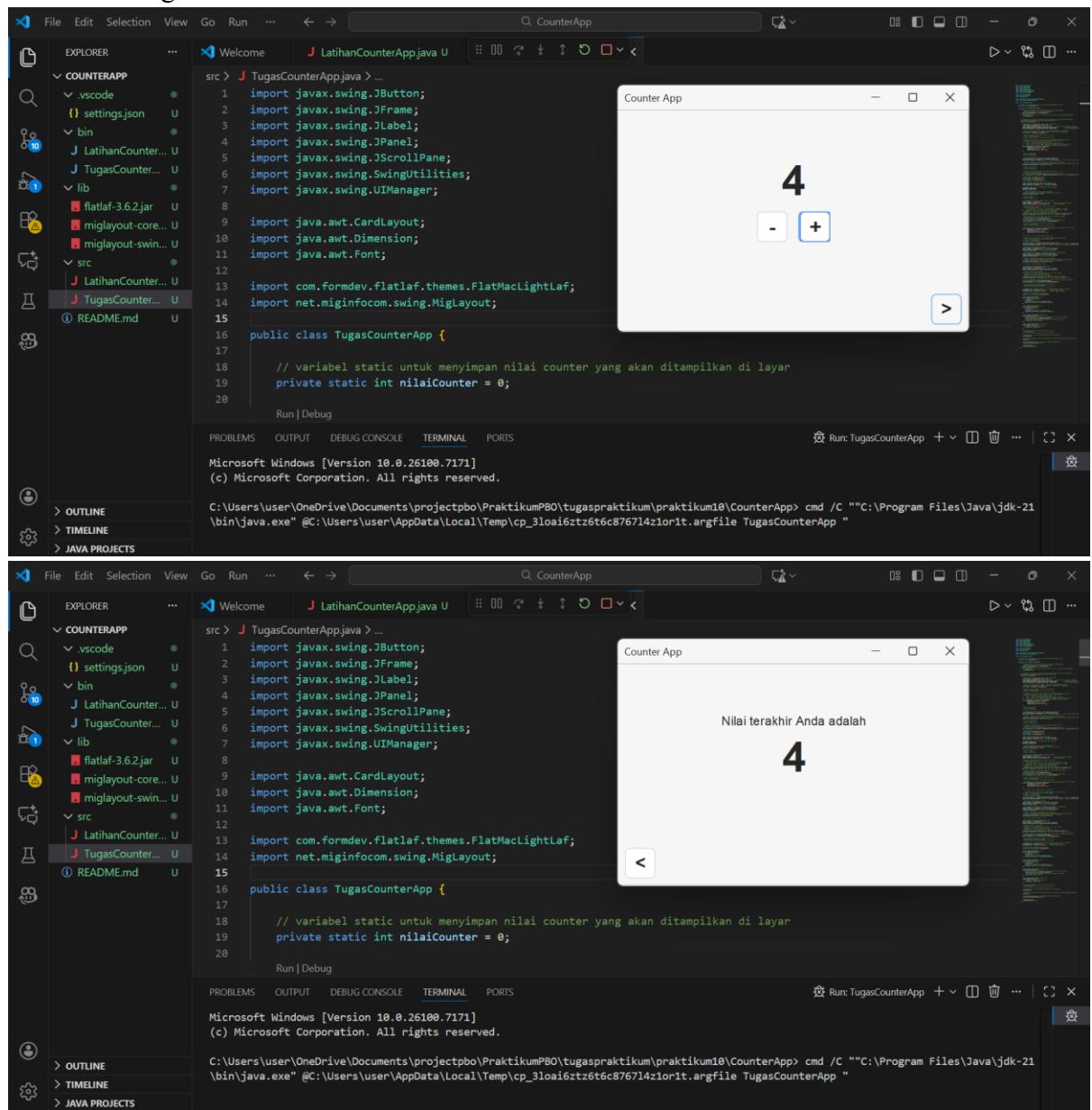
    // setLocationRelativeTo(null) → posisikan jendela di tengah
    layar
    frame.setLocationRelativeTo(null);

    // Tampilkan window
    frame.setVisible(true);
});
}

```



b. Hasil Running



3. Terdapat foto (scan) gambar dari media kertas beserta deskripsi penjelasannya mengenai rancangan tampilan UI menu utama aplikasi yang telah disepakati bersama oleh tim (mengacu ke tim proyek) dengan ketentuan:

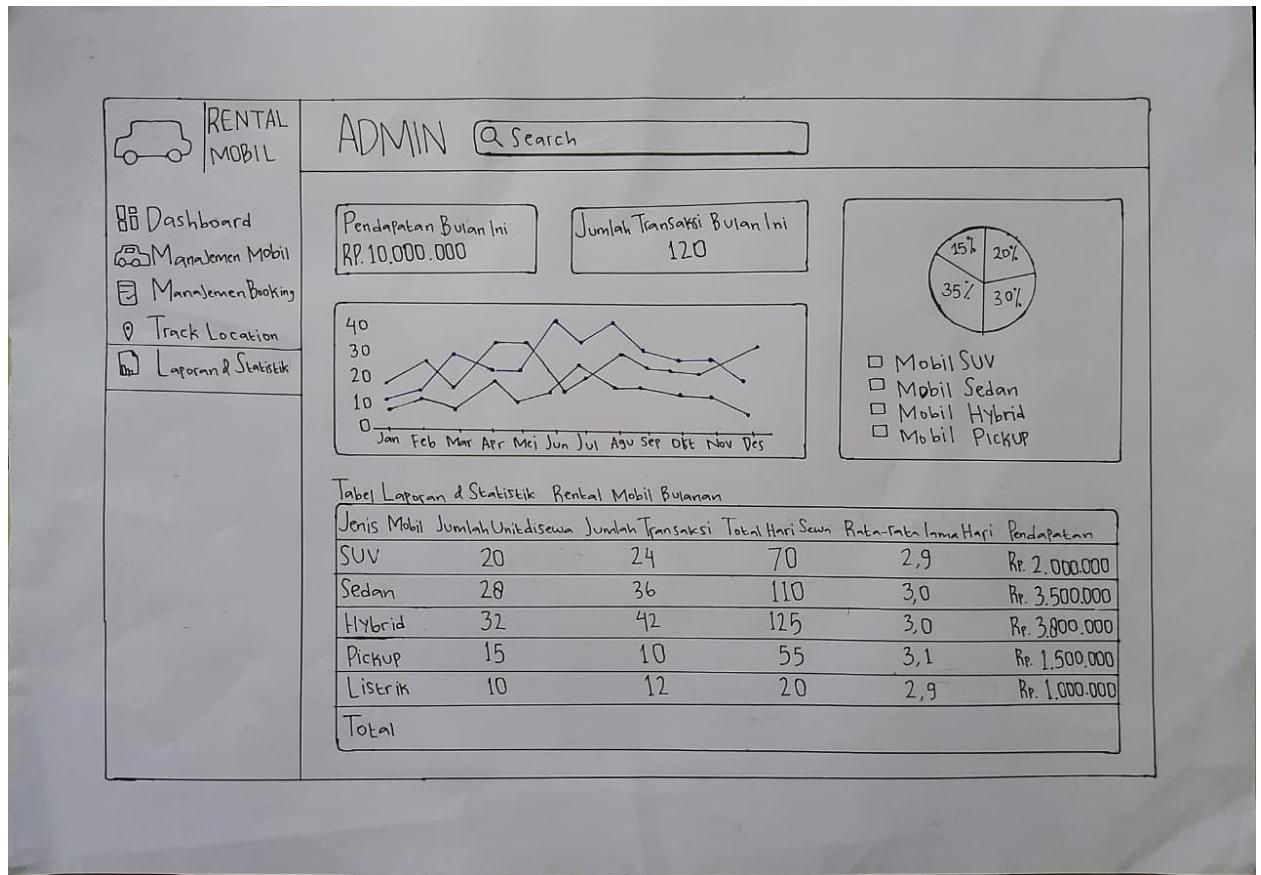
Kelompok 2

1. Affan Rifa'i (2403034)

2. Audy Zahra Aditya Putri (2403023)

3. Cania Nabilatul Adawah (2403102)

(a) tampilan UI menu utama aplikasi pada ukuran ideal,



Penjelasan:

1. Proporsi Sidebar vs Konten Seimbang
 - Sidebar tidak terlalu lebar atau terlalu sempit.
 - Masih mudah dibaca dan tidak mendominasi layar.
2. Header Lebar Secukupnya
 - Tidak melebar berlebihan.
 - Elemen di header (logo, judul, search bar) tersusun simetris.
3. Grid Layout Stabil & Rapi
 - Card pendapatan dan transaksi sejajar dalam 1 baris.
 - Grafik dan pie chart berada di grid yang seimbang: kiri–kanan.
 - Tabel berada di bawah, lebar cukup proporsional.

4. Ruang Kosong (White Space) Pas

- Tidak terlalu sempit, tidak terlalu luas.
- Membuat tampilan tidak sesak.

5. Tidak Terlihat Elemen Over-stretch

Berbeda dengan tampilan tidak ideal (layar terlalu lebar) yang membuat:

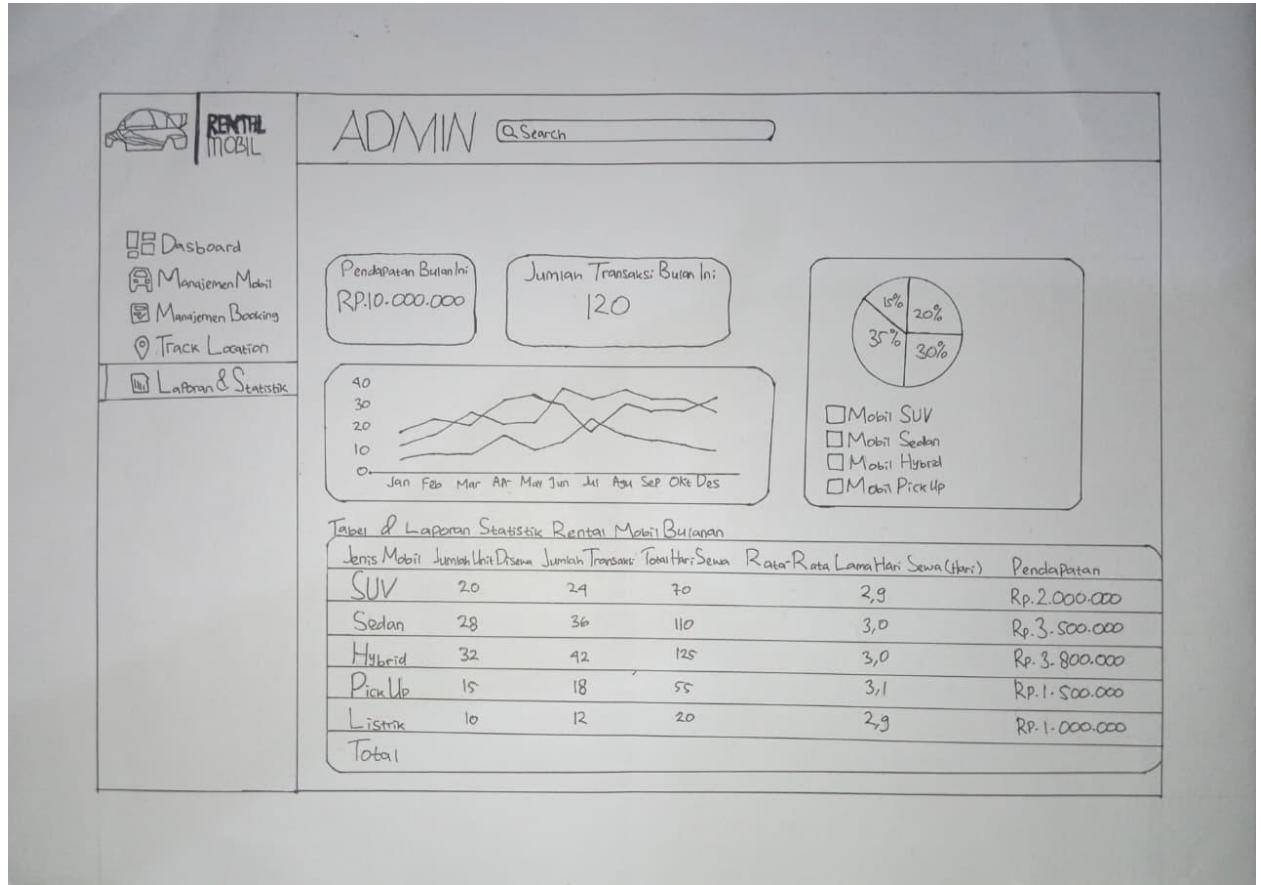
- tabel melebar terlalu besar,
- grafik tampak tidak seimbang,
- jarak antar block terlalu jauh.

Pada gambar ini hal itu *tidak terjadi*.

Kesimpulan:

UI ini proporsional.

(b) tampilan UI menu utama aplikasi pada ukuran tidak ideal (bisa jadi ukuran resolusi layar terlalu luas, atau lebih sempit dari ukuran ideal)



Penjelasan:

1. Area Konten Terlalu Kecil (Compressed Layout)

Ruang di kanan tampak terlalu luas dibandingkan area konten utama. Grafik, card, pie chart, dan tabel tampak “menyempit” ke tengah.

Dampak:

- Informasi terlihat padat pada area tengah.
- Banyak ruang kosong tidak terpakai di sisi kanan dan bawah.

2. Komponen Tidak Berada Dalam Grid Proporsional

Pada tampilan ideal, layout biasanya mengikuti struktur grid 12 kolom. Pada gambar ini:

- Sidebar tampak sangat lebar.
- Konten tampak terlalu sempit.

Dampak: keseimbangan visual hilang.

3. Card Informasi (Pendapatan & Transaksi) Tidak Simetris

- Card kanan lebih lebar dari card kiri.
- Jarak antar card tidak seimbang.

Ini membuat tampilan tidak rapi secara visual.

4. Pie Chart Terlalu Besar Dibanding Grafik Garis

Komponen visual yang lebih kecil (data pie) malah diberikan area lebih besar dari grafik utama (line chart).

Secara hirarki informasi, pie chart seharusnya kecil, grafik garis lebih menonjol.

5. Tabel Terlihat Terlalu “Padat” dan Tidak Memenuhi Lebar Grid

Tabel mulainya terlalu ke kiri dan lebarnya tidak proporsional.

Tabel seharusnya melebar sejajar dengan grafik, bukan lebih terlalu kekanan.

6. Terlalu Banyak Ruang Kosong Sebelah Kanan Grafik

Area di kanan grafik sangat luas tetapi tidak digunakan dengan efektif.

Kesimpulan Final

Tampilan ini Tidak Ideal untuk dashboard karena:

- proporsi layout tidak seimbang
- sidebar terlalu besar
- konten terlalu sempit
- grid layout tidak digunakan dengan benar
- komponen tidak simetris
- penggunaan ruang kosong tidak optimal

4. Pengalaman pembelajaran yang didapat (secara pribadi)

Saya mendapatkan pemahaman baru tentang bagaimana sebuah program dapat merespons tindakan pengguna melalui event, listener, dan event handler saat saya mempelajari topik pemrograman berbasis event. Pada awalnya, ide ini tampak berbeda dari pemrograman biasa yang bersifat linear. Namun, setelah mempelajari teori dan contoh yang diberikan, saya mulai memahami alurnya.

Saya belajar menggunakan Java Swing selama mengerjakan praktikum, seperti menambahkan ActionListener ke tombol dan mengatur perilaku komponen saat terjadi event. Proses ini membantu saya memahami bagaimana interaksi kecil, seperti klik tombol, dapat secara langsung mengubah tampilan aplikasi.

Link github : <https://github.com/caniaye/PraktikumPBO>