

## Laporan Praktikum 9 - Analisis Layout & UI Swing

Nama : Cania Nabilatul Adawah

NIM : 2403102

Mata Kuliah : Pemrograman Berbasis Objek

Sajikanlah Laporan Praktikum 9 - Analisis Layout & UI Swing dengan format (\*.pdf) dan ketentuan sebagai berikut:

1. Terdapat analisis layout & komponen penyusun UI yang disertai dengan kode program dan hasil running (compile & build) berdasarkan studi kasus yang disajikan pada slide
  2. Terdapat penyajian redaksi pengalaman pembelajaran yang didapat (secara pribadi) selama melakukan Analisis Layout & UI Swing
- 
1. Analisis Layout & Komponen Penyusun UI
    - a. Pemilihan Layout Manager (MigLayout) MigLayout dipilih karena fleksibilitas tinggi, mendukung pola grid responsif, dan sintaks yang ringkas sehingga memudahkan pengaturan panel Client, Information, dan Additional Info. Berbeda dengan FlowLayout yang terlalu sederhana atau GridBagLayout yang rumit, MigLayout memberikan keseimbangan antara kontrol dan kemudahan penulisan kode.
    - b. Komponen UI yang Digunakan Aplikasi memanfaatkan JLabel, JTextField non-editable, JTable, JScrollPane dengan scrollbar modern, JButton dengan custom rendering, serta Checkbox khusus dengan PurpleCheckBoxIcon. Semua komponen diberikan styling konsisten agar tampilan lebih profesional.
    - c. Desain Tiga Panel Utama Ketiga panel utama ditempatkan dalam tiga kolom menggunakan “wrap 3, fillx”, menghasilkan layout bersih dan tidak bertumpuk. Gap dan padding diatur secara presisi untuk memastikan tampilan nyaman dilihat pada resolusi besar seperti 1920×1080.
    - d. Tabel (Product, Tasks, Comments) Product List menggunakan tema ungu dengan highlight pada baris terpilih, sedangkan Tasks dan Comments memakai tema putih

- minimalis. Pengaturan alignment kolom, tinggi baris, dan header styling membuat data mudah dibaca.
- e. Scrollbar Modern Scrollbar dibuat kecil, rounded, dan berwarna abu agar tidak mengganggu tampilan. BasicScrollBarUI dioVERRIDE untuk menghasilkan efek modern minimalis.
  - f. Checkbox Kustom Checkbox digambar manual menggunakan Graphics2D agar warnanya senada dengan tema ungu. Centang putih tebal memberikan kesan modern dan jelas saat aktif.
  - g. Struktur Layout Bertingkat UI dibagi menjadi tiga tingkat: panel informasi utama, panel data (Product, Tasks, Comments), dan action bar paling bawah. Struktur ini membuat pengguna dapat memahami alur data dari atas ke bawah tanpa kebingungan.
  - h. Prinsip Desain Visual Tema warna ungu digunakan untuk elemen penting, panel memakai border radius 8, tombol radius 8–20, dan spacing dibuat rapi menggunakan gapx, gapy, dan insets. Hasilnya UI tidak terlihat seperti aplikasi Swing lama.
  - i. Responsivitas UI dioptimalkan untuk resolusi 1920×1080, memastikan semua panel muat tanpa scroll global. Penggunaan grow dan fillx membuat komponen menyesuaikan ruang secara otomatis.

Kesimpulan: Antarmuka AnalisisLayout berhasil memadukan kejelasan struktur, estetika modern, dan konsistensi tema. Pemilihan MigLayout, kustomisasi komponen, dan penerapan prinsip desain modern menghasilkan UI yang profesional, rapi, dan nyaman digunakan.

```
import net.miginfocom.swing.MigLayout;

import javax.swing.*;
import javax.swing.border.EmptyBorder;
import javax.swing.border.LineBorder;
import javax.swing.border.TitledBorder;
import javax.swing.table.DefaultTableCellRenderer;
import javax.swing.table.DefaultTableModel;
import javax.swing.table.TableCellRenderer;
import java.awt.*;
import java.awt.event.MouseAdapter;
import java.awt.event.MouseEvent;
import java.util.Vector;

public class DetailProyekForm extends JFrame {

    // --- Warna dan Font Konstanta ---
    private static final Color PRIMARY_PURPLE = new Color(155, 77, 255);
    private static final Color BUTTON_PURPLE = new Color(140, 90, 255);
    private static final Color BUTTON_HOVER_PURPLE = new Color(170, 100, 255);
    private static final Color BUTTON_WHITE_BG = Color.WHITE;
    private static final Color BUTTON_BORDER_COLOR = new Color(180, 180, 180);
    private static final Color BUTTON_TEXT_COLOR = Color.BLACK;
    private static final Color TEXT_FIELD_BG = Color.WHITE;
    private static final Color TEXT_FIELD_BORDER_COLOR = new Color(200, 200, 200);
    private static final Color PANEL_BG_COLOR = Color.WHITE;
    private static final Color BORDER_COLOR = new Color(229, 229, 229);
    private static final Color TITLE_COLOR = new Color(80, 80, 80);
    private static final Color TABLE_HEADER_BACKGROUND = new Color(240, 240, 240);
    private static final Color TABLE_GRID_COLOR = new Color(220, 220, 220);
    private static final Color SELECTED_ROW_PRODUCT_PURPLE = new Color(148, 87, 255);
    private static final Color SELECTED_ROW_SIMPLE_GREY = new Color(200, 200, 200);

    private static final Font BASE_FONT = new Font("Segoe UI", Font.PLAIN, 12);
    private static final Font BOLD_FONT = new Font("Segoe UI", Font.BOLD, 12);
    private static final Font BOLD_VAL_FONT = new Font("Segoe UI", Font.BOLD, 13);
```

```
    private static final Font BUTTON_FONT = new Font("Segoe UI Semibold",  
Font.PLAIN, 12);  
    private static final Font SMALL_BUTTON_FONT = new Font("Segoe UI",  
Font.PLAIN, 12);  
  
    // --- Variabel untuk Model Tabel ---  
    private DefaultTableModel modelProduct;  
    private DefaultTableModel modelTasks;  
    private DefaultTableModel modelComments;  
  
    // --- Variabel untuk Tabel ---  
    private JTable tblProduct;  
    private JTable tblTasks;  
    private JTable tblComments;  
  
    public DetailProyekForm() {  
        setTitle("Detail Proyek: RNV-JKT-AXG-001");  
        setSize(1300, 820); // agak dipendekkan sedikit  
        setLocationRelativeTo(null);  
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);  
        setLayout(new BorderLayout());  
  
        UIManager.put("Label.font", BASE_FONT);  
        UIManager.put("TextField.font", BASE_FONT);  
        UIManager.put("Table.font", BASE_FONT);  
        UIManager.put("TitledBorder.font", BASE_FONT);  
        UIManager.put("Button.font", SMALL_BUTTON_FONT);  
  
        JPanel contentPanel = new JPanel(new MigLayout(  
            "wrap 3, fillx, insets 4 8 0 8, gapy 4",  
            "[grow 30, fill][grow 40, fill][grow 30, fill]"  
        ));  
        contentPanel.setBackground(PANEL_BG_COLOR);  
  
        contentPanel.add(createClientPanel(), "grow");  
        contentPanel.add(createInfoPanel(), "grow");  
        contentPanel.add(createAdditionalInfoPanel(), "grow");  
  
        contentPanel.add(createProductListPanel(), "span 3, growx");  
        contentPanel.add(createTasksPanel(), "span 3, growx");  
        contentPanel.add(createCommentsPanel(), "span 3, growx");  
  
        add(contentPanel, BorderLayout.CENTER);  
        add(createBottomControlsPanel(), BorderLayout.SOUTH);  
    }  
}
```

```

// ====== PANEL CLIENT
=====
private JPanel createClientPanel() {
    JPanel pnlClient = new JPanel(new MigLayout(
        "wrap 2, insets 6 14 8 14, gapx 8, gapy 3",
        "[right][left]"
));
    pnlClient.setBackground(PANEL_BG_COLOR);
    pnlClient.setBorder(titledBorder("Client", BASE_FONT.getSize()));

    pnlClient.add(new JLabel("Client ID:"), "gapleft 18");
    pnlClient.add(boldVal("101"));

    pnlClient.add(new JLabel("Name:"), "gapleft 18");
    pnlClient.add(boldVal("Bapak Alex Gunawan"));

    pnlClient.add(new JLabel("Phone:"), "gapleft 18");
    pnlClient.add(boldVal("(+62) 8123456789"));

    pnlClient.add(new JLabel("Registration No:"), "gapleft 18");
    pnlClient.add(boldVal("RNV-JKT-AXG-001"));

    JButton btnDetails = purpleButton("Details");
    btnDetails.setFont(BUTTON_FONT);
    btnDetails.setPreferredSize(new Dimension(100, 28));

    pnlClient.add(btnDetails, "span 2, center, gaptop 4");

    btnDetails.addActionListener(e ->
        JOptionPane.showMessageDialog(this, "Tombol 'Details' Client ditekan.");
    );
}

return pnlClient;
}

// ====== STYLE DETAIL BUTTON
=====
private JButton purpleButton(String text) {
    JButton b = new JButton(text) {
        @Override
        protected void paintComponent(Graphics g) {
            Graphics2D g2 = (Graphics2D) g.create();
            g2.setRenderingHint(RenderingHints.KEY_ANTIALIASING,
                RenderingHints.VALUE_ANTIALIAS_ON);
        }
    };
}

```

```

        g2.setColor(getBackground());
        int arc = 20;
        g2.fillRoundRect(0, 0, getWidth(), getHeight(), arc, arc);
        g2.dispose();
        super.paintComponent(g);
    }
};

b.setForeground(Color.WHITE);
b.setBackground(BUTTON_PURPLE);
b.setOpaque(false);
b.setContentAreaFilled(false);
b.setBorder(new RoundedBorder(new Color(0, 0, 0), 1, 20));
b.setFocusPainted(false);
b.setCursor(Cursor.getPredefinedCursor(Cursor.HAND_CURSOR));
b.setFont(SMALL_BUTTON_FONT);
b.setMargin(new Insets(4, 18, 4, 18));

b.addMouseListener(new MouseAdapter() {
    @Override
    public void mouseEntered(MouseEvent e) {
        b.setBackground(BUTTON_HOVER_PURPLE);
        b.repaint();
    }

    @Override
    public void mouseExited(MouseEvent e) {
        b.setBackground(BUTTON_PURPLE);
        b.repaint();
    }
});

return b;
}

// ===== SMALL BUTTON
=====

private JButton smallBtn(String text) {
    JButton b = new JButton(text) {
        @Override
        protected void paintComponent(Graphics g) {
            Graphics2D g2 = (Graphics2D) g.create();
            g2.setRenderingHint(RenderingHints.KEY_ANTIALIASING,
                RenderingHints.VALUE_ANTIALIAS_ON);
            g2.setColor(getBackground());

```

```

        g2.fillRoundRect(0, 0, getWidth(), getHeight(), 8, 8);
        g2.dispose();
        super.paintComponent(g);
    }
};

b.setForeground(BUTTON_TEXT_COLOR);
b.setBackground(Color.WHITE);
b.setOpaque(false);
b.setContentAreaFilled(false);
b.setBorder(new RoundedBorder(BUTTON_BORDER_COLOR, 1, 8));
b.setFont(SMALL_BUTTON_FONT);
b.setCursor(Cursor.getPredefinedCursor(Cursor.HAND_CURSOR));
b.setPreferredSize(new Dimension(85, 24));
b.setMargin(new Insets(3, 10, 3, 10));

b.addMouseListener(new MouseAdapter() {
    @Override
    public void mouseEntered(MouseEvent e) {
        b.setBackground(new Color(245, 245, 245));
        b.repaint();
    }

    @Override
    public void mouseExited(MouseEvent e) {
        b.setBackground(Color.WHITE);
        b.repaint();
    }
});

return b;
}

// ====== INFO PANEL ======
private JPanel createInfoPanel() {
    JPanel pnlInfo = new JPanel(new MigLayout(
        "wrap 3, insets 6 14 8 14, gapx 5, gapy 3",
        "[right][grow][right]"
    ));
    pnlInfo.setBackground(PANEL_BG_COLOR);
    pnlInfo.setBorder(titledBorder("Information", BASE_FONT.getSize()));

    JButton btnAddDays = smallBtn("Add Days");
    pnlInfo.add(new JLabel("Reserve days:"));
    pnlInfo.add(field("0 of 30"), "growx");
}

```

```

pnlInfo.add(btnAddDays);

pnlInfo.add(new JLabel("Buyer:"));
pnlInfo.add(field("Bapak Alex Gunawan"), "span 2, growx");

pnlInfo.add(new JLabel("Seller:"));
pnlInfo.add(field("PT Bangun Jaya Abadi"), "span 2, growx");

pnlInfo.add(new JLabel("Address:"));
pnlInfo.add(field("Jl. Raya Lohbener Baru, 77BA"), "span 2, growx");

JButton btnSp = smallBtn("S&P Update");
pnlInfo.add(new JLabel("Credit rating:"));
pnlInfo.add(field("AAA"), "growx");
pnlInfo.add(btnSp);

JCheckBox chk = new JCheckBox();
chk.setSelected(true);
chk.setBackground(PANEL_BG_COLOR);
chk.setFont(BASE_FONT);
chk.setIcon(new PurpleCheckBoxIcon());
chk.setSelectedIcon(new PurpleCheckBoxIcon());

JLabel lblApproved = new JLabel("Approved:");
lblApproved.setFont(BASE_FONT);

 JPanel approvedPanel = new JPanel(new FlowLayout(FlowLayout.LEFT, 4,
0));
approvedPanel.setBackground(PANEL_BG_COLOR);
approvedPanel.setBorder(new EmptyBorder(0, 18, 0, 0));
approvedPanel.add(chk);
approvedPanel.add(lblApproved);

pnlInfo.add(approvedPanel, "gaptop 3");
pnlInfo.add(field("Proyek renovasi telah disetujui, siap dimulai"),
"span 2, growx, gaptop 3");

 return pnlInfo;
}

// ===== ADDITIONAL INFO PANEL =====

private JPanel createAdditionalInfoPanel() {
 JPanel pnl = new JPanel(new MigLayout(
 "wrap 3, insets 6 14 8 14, gapx 5, gapy 3",

```

```

        "[right][grow][right]"
));
pnl.setBackground(PANEL_BG_COLOR);
pnl.setBorder(titledBorder("Additional Information", BASE_FONT.getSize()));

pnl.add(new JLabel("Estimated close:"));
JTextField txtEstimated = field("2025-12-15");
pnl.add(txtEstimated, "w 140!");
JButton btnEditEstimated = smallBtn("Edit");
pnl.add(btnEditEstimated, "w 80!");

pnl.add(new JLabel("Creation date:"));
pnl.add(bold("2025-10-15"), "span 2, growx");
pnl.add(new JLabel("Created by:"));
pnl.add(bold("Admin"), "span 2, growx");
pnl.add(new JLabel("Last edit date:"));
pnl.add(bold("2025-11-16"), "span 2, growx");
pnl.add(new JLabel("Last edited by:"));
pnl.add(bold("Warnoto"), "span 2, growx");
pnl.add(new JLabel("Closed date:"));
pnl.add(bold("null"), "span 2, growx");
pnl.add(new JLabel("Closed by:"));
pnl.add(bold("null"), "span 2, growx");

return pnl;
}

```

//

-----

PRODUCT

PANEL

```

private JPanel createProductListPanel() {
    JPanel pnl = new JPanel(new MigLayout(
        "wrap 2, fillx, insets 6 10 6 10, gapx 8, gapy 4",
        "[grow][right]"
));
pnl.setBackground(PANEL_BG_COLOR);
pnl.setBorder(titledBorder("Product List", BASE_FONT.getSize() + 1));

String[] headProduct = {
    "Renovation", "Description", "Part #", "Quantity",
    "List Price", "Discount", "Price",
    "Wholesale Discount", "Wholesaler Price"
};

Object[][] data = {

```

```

        {"Dapur", "Keramik Dinding Putih", "KW-PT-DLX-01", 50, "150000",
        "0", "7500000", "5", "7125000"},

        {"Dapur", "Lem Keramik Instan", "LMI-GRY-STD", 5, "50000", "0",
        "250000", "0", "250000"},

        {"Dapur", "Pipa PVC 3 inch", "PVC-3IN-STD", 12, "35000", "0",
        "420000", "10", "378000"}}

    };

modelProduct = new DefaultTableModel(data, headProduct);
tblProduct = purpleTable(modelProduct);

JScrollPane sp = new JScrollPane(tblProduct);
sp.getVerticalScrollBar().setUI(new ModernScrollBarUI());
sp.getHorizontalScrollBar().setUI(new ModernScrollBarUI());
sp.getVerticalScrollBar().setPreferredSize(new Dimension(6, 0));
sp.getHorizontalScrollBar().setPreferredSize(new Dimension(0, 6));

int height = tblProduct.getRowHeight() * modelProduct.getRowCount()
        + tblProduct.getTableHeader().getPreferredSize().height + 4;
sp.setPreferredSize(new Dimension(1000, height));
sp.setBorder(new LineBorder(BORDER_COLOR));

JPanel purpleBackground = new JPanel(new BorderLayout());
purpleBackground.setBackground(new Color(243, 232, 255));
purpleBackground.setBorder(new LineBorder(new Color(155, 77, 255), 4,
true));
purpleBackground.add(sp, BorderLayout.CENTER);

pnl.add(purpleBackground, "grow, span 1 2");

JPanel btns = new JPanel(new MigLayout("wrap 1, insets 0, gapy 4"));
btns.setBackground(PANEL_BG_COLOR);

JButton add = smallBtn("Add");
JButton edit = smallBtn("Edit");
JButton del = smallBtn("Delete");

add.addActionListener(e -> addRow(modelProduct, headProduct, "Produk"));
edit.addActionListener(e -> editRow(tblProduct, modelProduct, headProduct,
"Produk"));
del.addActionListener(e -> deleteRow(tblProduct, modelProduct, "Produk"));

btns.add(add, "growx");
btns.add(edit, "growx");
btns.add(del, "growx");

```

```

        pnl.add(btns, "top");

        JPanel subtotal = new JPanel(new MigLayout("wrap 1, align right, insets 0,
gapy 1"));
        subtotal.setBackground(PANEL_BG_COLOR);
        subtotal.add(createSubtotalLabel("Subtotal list price:", "Rp 15.570.000"));
        subtotal.add(createSubtotalLabel("Total retailer price:", "Rp 15.182.500"));
        subtotal.add(createSubtotalLabel("Total wholesaler price:", "Rp 14.249.625"));

        pnl.add(subtotal, "cell 0 2, growx, align right, gapright 8, gaptop 2");

        return pnl;
    }

    private JLabel createSubtotalLabel(String label, String value) {
        JLabel l = new JLabel("<html><font color='#808080'>" + label +
        "</font> <b>" + value + "</b></html>");
        l.setFont(BASE_FONT);
        l.setHorizontalAlignment(SwingConstants.RIGHT);
        return l;
    }

// ===== TASK PANEL =====
private JPanel createTasksPanel() {
    JPanel pnl = new JPanel(new MigLayout(
        "wrap 2, fillx, insets 6 10 6 10"
    ));
    pnl.setBackground(PANEL_BG_COLOR);
    pnl.setBorder(titledBorder("Tasks", BASE_FONT.getSize() + 1));

    String[] head = {"State", "Task", "Assigner", "Executer",
        "Creation Date", "Estimated Date", "Executed Date"};

    Object[][] data = {
        {"Completed", "Pemasangan pipa PVC di area garasi", "Warnoto",
        "Goang", "2025-10-20", "2025-10-21", "2025-10-21"},

        {"Completed", "Pengecatan ulang ruang tamu", "Warnoto", "Toni", "2025-
        10-25", "2025-10-28", "2025-10-27"},

        {"Delayed", "Pemasangan keramik dinding dapur", "Warnoto", "Toni",
        "2025-11-01", "2025-11-04", ""}
    };

    modelTasks = new DefaultTableModel(data, head);
    tblTasks = simpleTable(modelTasks);
}

```

```

JScrollPane sp = new JScrollPane(tblTasks);
sp.getVerticalScrollBar().setUI(new ModernScrollBarUI());
sp.getHorizontalScrollBar().setUI(new ModernScrollBarUI());
sp.getVerticalScrollBar().setPreferredSize(new Dimension(6, 0));
sp.getHorizontalScrollBar().setPreferredSize(new Dimension(0, 6));

int height = tblTasks.getRowHeight() * modelTasks.getRowCount()
        + tblTasks.getTableHeader().getPreferredSize().height + 4;
sp.setPreferredSize(new Dimension(800, height));
sp.setBorder(new LineBorder(BORDER_COLOR));

pnl.add(sp, "grow");
if (modelTasks.getRowCount() > 0) {
    tblTasks.setRowSelectionInterval(2, 2);
}

JPanel btns = new JPanel(new MigLayout("wrap 1, insets 0, gapy 4"));
btns.setBackground(PANEL_BG_COLOR);

JButton add = smallBtn("Add");
JButton edit = smallBtn("Edit");
JButton del = smallBtn("Delete");

add.addActionListener(e -> addRow(modelTasks, head, "Tugas"));
edit.addActionListener(e -> editRow(tblTasks, modelTasks, head, "Tugas"));
del.addActionListener(e -> deleteRow(tblTasks, modelTasks, "Tugas"));

btns.add(add, "growx");
btns.add(edit, "growx");
btns.add(del, "growx");

pnl.add(btns, "top");

return pnl;
}

// ====== COMMENTS PANEL ======
=====

private JPanel createCommentsPanel() {
    JPanel pnl = new JPanel(new MigLayout(
        "wrap 2, fillx, insets 6 10 8 10"
    ));
    pnl.setBackground(PANEL_BG_COLOR);
    pnl.setBorder(titledBorder("Comments", BASE_FONT.getSize() + 1));
}

```

```

String[] head = {"Date", "User", "Comment"};
Object[][] data = {
    {"2025-10-26 10:15", "Toni", "Cat tembok sudah dilolesi lapisan pertama.  
Menunggu kering sebelum lapisan kedua."},
    {"2025-11-03 16:45", "Toni", "Cuaca hujan deras selama 2 hari, area kerja  
sedikit basah. Pemasangan ditunda besok pagi."},
    {"2025-11-15 14:30", "Goang", "Barang sudah sampai di lokasi. Mulai  
proses pembongkaran closet lama sore ini."}
};

modelComments = new DefaultTableModel(data, head);
tblComments = simpleTable(modelComments);

tblComments.getColumnModel().getColumn(0).setPreferredWidth(120);
tblComments.getColumnModel().getColumn(1).setPreferredWidth(50);
tblComments.getColumnModel().getColumn(2).setPreferredWidth(500);

JScrollPane sp = new JScrollPane(tblComments);
sp.getVerticalScrollBar().setUI(new ModernScrollBarUI());
sp.getHorizontalScrollBar().setUI(new ModernScrollBarUI());
sp.getVerticalScrollBar().setPreferredSize(new Dimension(6, 0));
sp.getHorizontalScrollBar().setPreferredSize(new Dimension(0, 6));

int height = tblComments.getRowHeight() * modelComments.getRowCount()
        + tblComments.getTableHeader().getPreferredSize().height + 4;
sp.setPreferredSize(new Dimension(800, height));
sp.setBorder(new LineBorder(BORDER_COLOR));

pnl.add(sp, "grow");
if (modelComments.getRowCount() > 0) {
    tblComments.setRowSelectionInterval(1, 1);
}

JPanel btns = new JPanel(new MigLayout("wrap 1, insets 0, gapy 4"));
btns.setBackground(PANEL_BG_COLOR);

JButton add = smallBtn("Add");
JButton edit = smallBtn("Edit");
JButton del = smallBtn("Delete");

add.addActionListener(e -> addRow(modelComments, head, "Komentar"));
edit.addActionListener(e -> editRow(tblComments, modelComments, head,
"Komentar"));

```

```

        del.addActionListener(e -> deleteRow(tblComments, modelComments,
"Komentar"));

        btns.add(add, "growx");
        btns.add(edit, "growx");
        btns.add(del, "growx");

        pnl.add(btns, "top");

        return pnl;
    }

    // ===== BOTTOM BUTTON PANEL
=====

    private JPanel createBottomControlsPanel() {
        JPanel pnlBottom = new JPanel(new MigLayout("insets 6 10 6 10, fillx, gap
0"));
        pnlBottom.setBackground(PANEL_BG_COLOR);
        pnlBottom.setBorder(BorderFactory.createMatteBorder(1, 0, 0, 0, BORDER_COLOR));

        JButton btnExport = smallBtn("Export");
        btnExport.addActionListener(e ->
            JOptionPane.showMessageDialog(this, "Aksi: Mengakses data proyek.")
        );
        pnlBottom.add(btnExport, "left, gapleft 5");

        pnlBottom.add(new JLabel(""), "growx");

        JPanel pnlRightButtons = new JPanel(new MigLayout("insets 0, gap 4"));
        pnlRightButtons.setBackground(PANEL_BG_COLOR);

        JButton btnSave = smallBtn("Save");
        JButton btnCancel = smallBtn("Cancel");

        btnSave.addActionListener(e ->
            JOptionPane.showMessageDialog(this, "Aksi: Menyimpan perubahan data
proyek.")
        );
        btnCancel.addActionListener(e -> {
            int confirm = JOptionPane.showConfirmDialog(
                this,
                "Yakin ingin membatalkan perubahan?",
                "Konfirmasi",
                JOptionPane.YES_NO_OPTION
            );
        });
    }
}

```

```

    );
    if(confirm == JOptionPane.YES_OPTION) {
        JOptionPane.showMessageDialog(this, "Aksi: Membatalkan perubahan.");
    }
});

pnlRightButtons.add(btnSave);
pnlRightButtons.add(btnCancel);

pnlBottom.add(pnlRightButtons, "right");

return pnlBottom;
}

// ====== CRUD + UTILITIES ======
=====

private void addRow(TableModel model, String[] headers, String itemType) {
    JComponent[] fields = new JComponent[headers.length];
    JPanel panel = new JPanel(new MigLayout("wrap 2, gap 10"));

    boolean isComment = itemType.equals("Komentar");

    for (int i = 0; i < headers.length; i++) {
        panel.add(new JLabel(headers[i] + ":"));

        if (isComment && headers[i].equals("Comment")) {
            JTextArea ta = new JTextArea(5, 30);
            ta.setLineWrap(true);
            ta.setWrapStyleWord(true);
            fields[i] = new JScrollPane(ta);
        } else {
            JTextField tf = new JTextField(20);
            tf.setBorder(new LineBorder(TEXT_FIELD_BORDER_COLOR));
            tf.setBackground(TEXT_FIELD_BG);
            fields[i] = tf;
        }
        panel.add(fields[i]);
    }

    if (JOptionPane.showConfirmDialog(
        this, panel, "Tambah " + itemType,
        JOptionPane.OK_CANCEL_OPTION, JOptionPane.PLAIN_MESSAGE
    ) == JOptionPane.OK_OPTION) {

```

```

Vector<String> newRow = new Vector<>();
for (JComponent f : fields) {
    if (f instanceof JTextField)
        newRow.add(((JTextField) f).getText());
    else if (f instanceof JScrollPane)
        newRow.add(((JTextArea)
            ((ScrollPane) f).getViewport().getView()).getText());
}
model.addRow(newRow);
}

private void editRow(JTable table, DefaultTableModel model, String[] headers,
String itemType) {
    int selected = table.getSelectedRow();
    if (selected == -1) {
        JOptionPane.showMessageDialog(this, "Silakan pilih baris untuk di edit.");
        return;
    }

    int modelRow = table.convertRowIndexToModel(selected);

    JComponent[] fields = new JComponent[headers.length];
    JPanel panel = new JPanel(new MigLayout("wrap 2, gap 10"));

    boolean isComment = itemType.equals("Komentar");

    for (int i = 0; i < headers.length; i++) {
        panel.add(new JLabel(headers[i] + ":"));

        String value = model.getValueAt(modelRow, i) != null
            ? model.getValueAt(modelRow, i).toString()
            : "";

        if (isComment && headers[i].equals("Comment")) {
            JTextArea ta = new JTextArea(value, 5, 30);
            ta.setLineWrap(true);
            ta.setWrapStyleWord(true);
            fields[i] = new JScrollPane(ta);
        } else {
            JTextField tf = new JTextField(value, 20);
            tf.setBorder(new LineBorder(TEXT_FIELD_BORDER_COLOR));
            tf.setBackground(TEXT_FIELD_BG);
            fields[i] = tf;
        }
    }
}

```

```
        panel.add(fields[i]);
    }

    if (JOptionPane.showConfirmDialog(
        this, panel, "Edit " + itemType,
        JOptionPane.OK_CANCEL_OPTION, JOptionPane.PLAIN_MESSAGE
    ) == JOptionPane.OK_OPTION) {

        for (int i = 0; i < headers.length; i++) {
            if (fields[i] instanceof JTextField)
                model.setValueAt(((JTextField) fields[i]).getText(), modelRow, i);
            else if (fields[i] instanceof JScrollPane)
                model.setValueAt(((JTextArea)
                    ((ScrollPane) fields[i]).getViewport().getView()).getText(),
                    modelRow, i);
        }
    }
}

private void deleteRow(JTable table, DefaultTableModel model, String itemType)
{
    int selected = table.getSelectedRow();
    if (selected == -1) {
        JOptionPane.showMessageDialog(this, "Silakan pilih baris untuk dihapus.");
        return;
    }

    if (JOptionPane.showConfirmDialog(
        this,
        "Yakin ingin menghapus " + itemType + "?",
        "Konfirmasi",
        JOptionPane.YES_NO_OPTION
    ) == JOptionPane.YES_OPTION) {

        int modelRow = table.convertRowIndexToModel(selected);
        model.removeRow(modelRow);
    }
}

private JLabel bold(String txt) {
    JLabel l = new JLabel(txt);
    l.setFont(BOLD_FONT);
    return l;
}
```

```

private JLabel boldVal(String txt) {
    JLabel l = new JLabel(txt);
    l.setFont(BOLD_VAL_FONT);
    return l;
}

private JTextField field(String txt) {
    JTextField f = new JTextField(txt);
    f.setEditable(false);
    f.setBorderStyle(new LineBorder(TEXT_FIELD_BORDER_COLOR));
    f.setBackground(TEXT_FIELD_BG);
    f.setMargin(new Insets(2, 4, 2, 4));
    f.setPreferredSize(new Dimension(95, 22));
    f.setFont(BASE_FONT);
    return f;
}

// ===== TABLES =====
private JTable purpleTable(DefaultTableModel model) {
    JTable tbl = new JTable(model) {
        @Override
        public Component prepareRenderer(TableCellRenderer r, int row, int col) {
            Component c = super.prepareRenderer(r, row, col);
            if(c instanceof JComponent) {
                if(isRowSelected(row)) {
                    c.setBackground(SELECTED_ROW_PRODUCT_PURPLE);
                    c.setForeground(Color.WHITE);
                } else {
                    c.setBackground(Color.WHITE);
                    c.setForeground(Color.BLACK);
                }
            }
            return c;
        }
    };
    DefaultTableCellRenderer right = new DefaultTableCellRenderer();
    right.setHorizontalAlignment(SwingConstants.RIGHT);
    for(int i = 4; i <= 8; i++) {
        if(i < tbl.getColumnCount()) {
            tbl.getColumnModel().getColumn(i).setCellRenderer(right);
        }
    }
}

```

```
tbl.getTableHeader().setBackground(TABLE_HEADER_BACKGROUND);
tbl.getTableHeader().setFont(BASE_FONT);
tbl.getTableHeader().setPreferredSize(new Dimension(0, 26));
tbl.setRowHeight(24);
tbl.setGridColor(TABLE_GRID_COLOR);
tbl.getTableHeader().setReorderingAllowed(false);

if (model.getRowCount() >= 3) {
    tbl.setRowSelectionInterval(0, 2);
}

return tbl;
}

private JTable simpleTable(DefaultTableModel model) {
    JTable tbl = new JTable(model);

    tbl.getTableHeader().setBackground(TABLE_HEADER_BACKGROUND);
    tbl.getTableHeader().setFont(BASE_FONT);
    tbl.getTableHeader().setPreferredSize(new Dimension(0, 26));
    tbl.setRowHeight(22);
    tbl.setGridColor(TABLE_GRID_COLOR);
    tbl.getTableHeader().setReorderingAllowed(false);

    tbl.setDefaultRenderer(Object.class, new DefaultTableCellRenderer() {
        @Override
        public Component getTableCellRendererComponent(
            JTable t, Object val, boolean sel, boolean focus, int row, int col
        ) {
            Component c = super.getTableCellRendererComponent(t, val, sel, focus,
row, col);
            if (c instanceof JComponent) {
                if (sel) {
                    c.setBackground(SELECTED_ROW_SIMPLE_GREY);
                    c.setForeground(Color.BLACK);
                } else {
                    c.setBackground(Color.WHITE);
                    c.setForeground(Color.BLACK);
                }
            }
            return c;
        }
    });
}

return tbl;
}
```

```

    }

// ===== BORDER =====
private TitledBorder titled(String text, int fontSize) {
    TitledBorder tb = new TitledBorder(
        new RoundedBorder(BORDER_COLOR, 1, 8),
        text
    );
    tb.setTitleFont(new Font("Segoe UI", Font.PLAIN, fontSize));
    tb.setTitleColor(TITLE_COLOR);
    return tb;
}

private static class RoundedBorder extends LineBorder {
    private final int radius;

    public RoundedBorder(Color color, int thickness, int radius) {
        super(color, thickness, true);
        this.radius = radius;
    }

    @Override
    public void paintBorder(Component c, Graphics g, int x, int y, int w, int h) {
        Graphics2D g2 = (Graphics2D) g.create();
        g2.setRenderingHint(RenderingHints.KEY_ANTIALIASING,
            RenderingHints.VALUE_ANTIALIAS_ON);
        g2.setColor(lineColor);
        g2.setStroke(new BasicStroke(thickness));
        int offs = thickness / 2;
        g2.drawRoundRect(x + offs, y + offs, w - thickness, h - thickness, radius,
            radius);
        g2.dispose();
    }
}

// ===== MODERN SCROLLBAR =====
class ModernScrollBarUI extends javax.swing.plaf.basic.BasicScrollBarUI {

    private final int thickness = 6;

    @Override
    protected JButton createDecreaseButton(int orientation) {
        return createZeroButton();
    }
}

```

```
@Override
protected JButton createIncreaseButton(int orientation) {
    return createZeroButton();
}

private JButton createZeroButton() {
    JButton btn = new JButton();
    btn.setPreferredSize(new Dimension(0, 0));
    btn.setMinimumSize(new Dimension(0, 0));
    btn.setMaximumSize(new Dimension(0, 0));
    return btn;
}

@Override
protected void configureScrollBarColors() {
    this.thumbColor = new Color(180, 180, 180);
    this.trackColor = new Color(245, 245, 245);
}

@Override
protected void paintThumb(Graphics g, JComponent c, Rectangle r) {
    Graphics2D g2 = (Graphics2D) g.create();
    g2.setRenderingHint(RenderingHints.KEY_ANTIALIASING,
    RenderingHints.VALUE_ANTIALIAS_ON);
    g2.setColor(thumbColor);
    g2.fillRoundRect(r.x, r.y, r.width, r.height, 10, 10);
    g2.dispose();
}

@Override
protected void paintTrack(Graphics g, JComponent c, Rectangle r) {
    Graphics2D g2 = (Graphics2D) g.create();
    g2.setColor(trackColor);
    g2.fillRect(r.x, r.y, r.width, r.height);
    g2.dispose();
}

@Override
protected Dimension getMinimumThumbSize() {
    return new Dimension(thickness, 30);
}

@Override
protected Dimension getMaximumThumbSize() {
```

```
        return new Dimension(thickness, 100);
    }

}

// ===== PURPLE CHECKBOX ICON =====

class PurpleCheckBoxIcon implements Icon {
    private final int size = 16;
    private final Color borderColor = new Color(155, 77, 255);
    private final Color fillColor = new Color(155, 77, 255);

    @Override
    public int getIconWidth() { return size; }

    @Override
    public int getIconHeight() { return size; }

    @Override
    public void paintIcon(Component c, Graphics g, int x, int y) {
        JCheckBox cb = (JCheckBox) c;
        Graphics2D g2 = (Graphics2D) g.create();
        g2.setRenderingHint(RenderingHints.KEY_ANTIALIASING,
RenderingHints.VALUE_ANTIALIAS_ON);

        g2.setColor(borderColor);
        g2.drawRoundRect(x, y, size - 1, size - 1, 4, 4);

        if (cb.isSelected()) {
            g2.setColor(fillColor);
            g2.fillRoundRect(x + 2, y + 2, size - 4, size - 4, 4, 4);

            g2.setColor(Color.WHITE);
            g2.setStroke(new BasicStroke(2f));
            g2.drawLine(x + 4, y + 8, x + 7, y + 12);
            g2.drawLine(x + 7, y + 12, x + 12, y + 4);
        }

        g2.dispose();
    }
}

public static void main(String[] args) {
    SwingUtilities.invokeLater(() -> new DetailProyekForm().setVisible(true));
}
}
```

## Hasil

## Running

The screenshot displays a Java Swing application window titled "Detail Proyek: RNV-JKT-AXG-001". The interface is divided into several sections:

- Client:** Shows Client ID: 101, Name: Bapak Alex Gunawan, Phone: (+62) 8123456789, Registration No: RNV-JKT-AXG-001. A "Details" button is present.
- Information:** Reserve days: 0 of 30, Add Days button. Buyer: Bapak Alex Gunawan, Seller: PT Bangun Jaya Abadi, Address: JL. Raya Lohbener Baru, 77BA. Credit rating: AAA, S&P Update button. A checked checkbox says Approved: Proyek renovasi telah disetujui, siap dimulai.
- Additional Information:** Estimated close: 2025-12-15, Creation date: 2025-10-15, Created by: Admin, Last edit date: 2025-11-16, Last edited by: Warnoto, Closed date: null, Closed by: null.
- Product List:** A table showing items: Dapur (Keramik Dinding Putih, KW-PT-DLX-01, 50 units, 150000 price), Dapur (Lem Keramik Instan, LMI-GRY-STD, 5 units, 50000 price). Subtotal list price: Rp 15.570.000, Total retailer price: Rp 15.182.500, Total wholesaler price: Rp 14.249.625.
- Tasks:** A table with columns State, Task, Assigner, Executer, Creation Date, Estimated Date, Executed Date. Rows include Completed task (Pengecatan ulang ruang... assigned to Warnoto, executed by Toni on 2025-10-25, estimated 2025-10-28, executed 2025-10-27) and Delayed task (Pemasangan keramik di... assigned to Warnoto, executed by Toni on 2025-11-01, estimated 2025-11-04).
- Comments:** A table with columns Date, User, Comment. Rows include a comment from Toni on 2025-11-03 at 16:45 about rainy weather, and a comment from Goang on 2025-11-15 at 14:30 about items arriving at the location.
- Buttons:** Add, Edit, Delete, Export, Save, Cancel.

## 2. Pengalaman pembelajaran yang didapat

Selama proses analisis dan pengembangan UI menggunakan Java Swing, saya mendapatkan beberapa pengalaman penting. Pertama, saya belajar bahwa penataan komponen dalam Swing bukan hanya sekadar meletakkan elemen, tetapi memastikan semuanya konsisten, mudah dibaca, dan nyaman digunakan. MigLayout sangat membantu karena lebih fleksibel dibandingkan layout dasar seperti BorderLayout atau GridBagLayout. Saya juga memahami bahwa detail kecil seperti jarak antar komponen, warna, garis tepi, dan respons tombol sangat berpengaruh pada profesionalitas tampilan. Membuat custom button dan scrollbar memberikan insight bahwa Swing bisa dibuat modern meskipun merupakan teknologi lama. Selain itu, proses menyusun UI ini mengajarkan pentingnya iterasi — melakukan revisi ukuran panel, merapikan alignment, dan memastikan tabel tidak melebar berlebihan. Saya juga belajar mengatur keseimbangan antara estetika dan fungsionalitas, sehingga UI tidak hanya bagus dilihat tetapi juga mudah digunakan. Secara keseluruhan, pengalaman ini memperdalam pemahaman saya terkait desain antarmuka berbasis desktop, terlebih bagaimana mengoptimalkan elemen visual di Java Swing agar terasa modern dan nyaman digunakan.

Link github :

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

