

Praktikum Pemrograman II



DISUSUN OLEH :

Hikmal Ryvaldi Pelu

223040023

TEKNIK INFORMATIKA

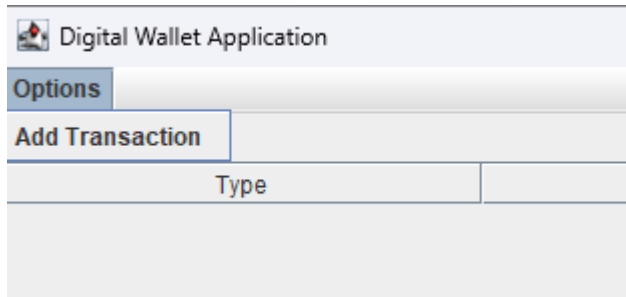
UNIVERSITAS

PASUNDAN BANDUNG

2024

1. Membuat sebuah antarmuka pengguna utama untuk navigasi ke antarmuka pengguna lain menggunakan JMenuBar dan JMenuBar

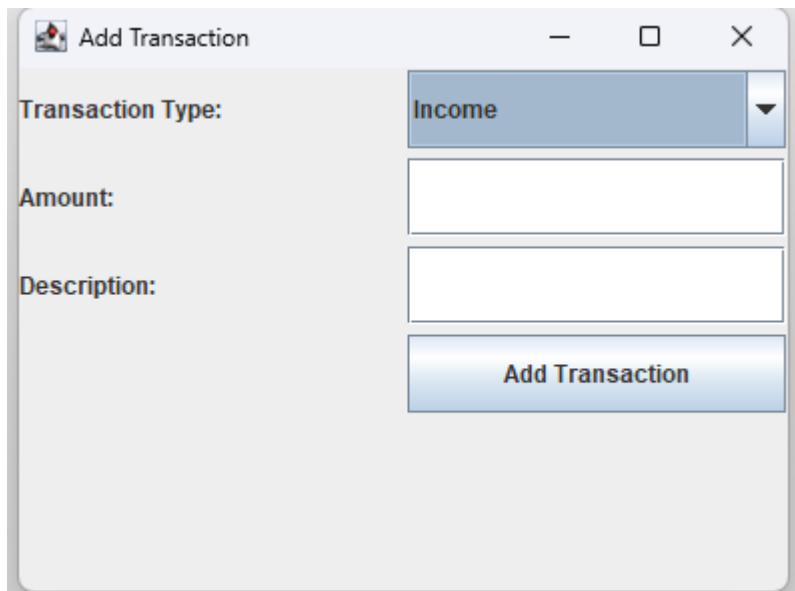
HASIL :



Pada kode ini, antarmuka utama aplikasi menggunakan JMenuBar dan JMenu di bagian atas aplikasi. JMenu diberi nama "Options" dan memiliki satu item menu bernama "Add Transaction", yang berfungsi untuk membuka form transaksi.

2. Membuat 1-3 form input yang menggunakan komponen JTextField, JTextArea, JRadioButton, JCheckBox, JComboBox, dan JList (bonus jika menggunakan JSlider dan JSpinner juga)

HASIL :



Pada kode ini, form transaksi yang terbuka saat memilih "Add Transaction" dari menu menggunakan beberapa komponen GUI sesuai tugas Anda:

- JComboBox untuk memilih jenis transaksi (Income atau Expense).
- JTextField untuk memasukkan jumlah (amountField) dan deskripsi (descriptionField).
- JButton untuk menambahkan transaksi (addButton).

3. Setiap form input dilengkapi dengan JTable untuk menampung inputan

HASIL :

Digital Wallet Application			
Options			
Balance: \$-80000.0			
Type	Amount	Description	Balance After Transaction
Income	\$100000.0	Pemasukan Gaji	\$100000.0
Expense	\$180000.0	Biaya Listrik dan Gass	\$-80000.0

Pada antarmuka utama, tabel JTable digunakan untuk menampilkan setiap transaksi yang ditambahkan. Setiap kali sebuah transaksi ditambahkan, tabel akan diperbarui dengan informasi baru.

Code :

```
import javax.swing.*;
import javax.swing.table.DefaultTableModel;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;

public class DigitalWalletApp extends JFrame {
    private JTable transactionTable;
    private DefaultTableModel tableModel;
    private JLabel balanceLabel;
    private double balance = 0.0;

    public DigitalWalletApp() {
        // Setup frame
        setTitle("Digital Wallet Application");
        setSize(700, 500);
        setDefaultCloseOperation(EXIT_ON_CLOSE);
        setLocationRelativeTo(null);

        // Setup menu bar
        JMenuBar menuBar = new JMenuBar();
        JMenu menu = new JMenu("Options");
        JMenuItem addTransactionMenuItem = new JMenuItem("Add Transaction");
        addTransactionMenuItem.addActionListener(e -> openTransactionForm());
        menu.add(addTransactionMenuItem);
        menuBar.add(menu);
        setJMenuBar(menuBar);

        // Balance label
        balanceLabel = new JLabel("Balance: $0.0", JLabel.CENTER);
        balanceLabel.setFont(new Font("Arial", Font.BOLD, 20));
        add(balanceLabel, BorderLayout.NORTH);

        // Initialize transaction table
        tableModel = new DefaultTableModel();
        tableModel.addColumn("Type");
        tableModel.addColumn("Amount");
        tableModel.addColumn("Description");
        tableModel.addColumn("Balance After Transaction");

        transactionTable = new JTable(tableModel);
        add(new JScrollPane(transactionTable), BorderLayout.CENTER);
    }

    private void openTransactionForm() {
        JFrame transactionFrame = new JFrame("Add Transaction");
        transactionFrame.setSize(400, 300);
        transactionFrame.setLayout(new GridLayout(6, 2, 5, 5));

        // Form components
        JLabel typeLabel = new JLabel("Transaction Type:");
        JComboBox<String> typeComboBox = new JComboBox<>(new String[]{"Income", "Expense"});

        JLabel amountLabel = new JLabel("Amount:");
        JTextField amountField = new JTextField();

        JLabel descriptionLabel = new JLabel("Description:");
        JTextField descriptionField = new JTextField();

        JButton addButton = new JButton("Add Transaction");
        addButton.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent e) {
                String type = (String) typeComboBox.getSelectedItem();
                double amount = Double.parseDouble(amountField.getText());
                String description = descriptionField.getText();

                // Update balance
                if ("Income".equals(type)) {
                    balance += amount;
                } else {
                    balance -= amount;
                }

                // Update table and balance label
                tableModel.addRow(new Object[]{type, "$" + amount, description, "$" +
                    balance});
                balanceLabel.setText("Balance: $" + balance);

                // Close transaction form
                transactionFrame.dispose();
            }
        });

        // Add components to the form frame
        transactionFrame.add(typeLabel);
        transactionFrame.add(typeComboBox);
        transactionFrame.add(amountLabel);
        transactionFrame.add(amountField);
        transactionFrame.add(descriptionLabel);
        transactionFrame.add(descriptionField);
        transactionFrame.add(new JLabel()); // Spacer
        transactionFrame.add(addButton);

        // Show transaction form
        transactionFrame.setVisible(true);
    }

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