**Laporan Program Manajemen Keuangan Pribadi**

**Deskripsi**

Program ini adalah aplikasi desktop berbasis Java Swing untuk membantu pengguna mencatat dan mengelola pemasukan serta pengeluaran. Aplikasi menyimpan data secara lokal dan menampilkan laporan bulanan dalam bentuk teks.

**Fitur Utama**

* **Input transaksi:**  
  Pengguna dapat menambahkan transaksi berupa keterangan, jumlah uang, dan jenis transaksi (Pemasukan atau Pengeluaran).
* **Tabel transaksi:**  
  Semua transaksi ditampilkan dalam tabel yang bisa langsung terlihat pengguna.
* **Penyimpanan lokal:**  
  Data disimpan di file data\_keuangan.dat secara otomatis menggunakan serialisasi Java.
* **Grafik bulanan (teks):**  
  Laporan keuangan ditampilkan dalam bentuk teks bulanan menggunakan JOptionPane.
* **File Batch & Shell:**  
  Dapat dijalankan lewat file runManajemenKeuangan.bat (Windows) atau runManajemenKeuangan.sh(Linux/Mac).

**Struktur Kode dan Penjelasan Fungsi**

**Kelas ManajemenKeuanganPribadi**

Merupakan kelas utama GUI aplikasi berbasis JFrame.

**Atribut:**

private JTable table;

private DefaultTableModel tableModel;

private JTextField tfKeterangan, tfJumlah;

private JComboBox<String> cbTipe;

private List<Transaksi> transaksiList = new ArrayList<>();

private final String FILE\_NAME = "data\_keuangan.dat";

* **table** dan **tableModel**: Menyimpan dan menampilkan data transaksi dalam tabel.
* **tfKeterangan**, **tfJumlah**: Input dari pengguna.
* **cbTipe**: Pilihan tipe transaksi.
* **transaksiList**: Menyimpan semua data transaksi dalam sesi program.
* **FILE\_NAME**: Nama file penyimpanan data transaksi.

**Konstruktor ManajemenKeuanganPribadi()**

Menyiapkan seluruh tampilan dan komponen:

* Menampilkan form input
* Menyiapkan tabel transaksi
* Memuat data dari file lokal
* Menambahkan tombol untuk melihat grafik bulanan

**Fungsi: tambahTransaksi()**

Menambahkan transaksi baru ke daftar dan tabel.

private void tambahTransaksi() {

...

}

Langkah-langkah:

1. Mengambil input dari form.
2. Melakukan validasi angka pada input jumlah.
3. Membuat objek Transaksi dan menambahkannya ke daftar.
4. Memperbarui tampilan tabel.
5. Menyimpan data ke file (saveData()).

**Fungsi: tampilkanGrafik()**

Menampilkan laporan pemasukan dan pengeluaran bulanan dalam bentuk teks.

private void tampilkanGrafik() {

...

}

Langkah-langkah:

1. Mengelompokkan total pemasukan dan pengeluaran berdasarkan bulan (1-12).
2. Menampilkan hasil dalam JTextArea di dalam dialog JOptionPane.

**Fungsi: saveData()**

Menyimpan semua transaksi ke file lokal menggunakan serialisasi.

private void saveData() {

...

}

**Fungsi: loadData()**

Memuat data transaksi dari file data\_keuangan.dat.

private void loadData() {

...

}

Jika file ditemukan, membaca isi dan mengisi transaksiList.

**Fungsi: main()**

Menjalankan program dan menampilkan GUI.

public static void main(String[] args) {

SwingUtilities.invokeLater(() -> new ManajemenKeuanganPribadi().setVisible(true));

}

**Kelas Transaksi**

Objek model data transaksi yang disimpan.

class Transaksi implements Serializable {

LocalDate tanggal;

String keterangan;

double jumlah;

String tipe;

...

}

* Serializable agar bisa disimpan ke file.
* Menyimpan: tanggal transaksi, keterangan, jumlah uang, dan tipe transaksi.

**Cara Build dan Jalankan**

**1. Compile Manual**

Terminal :

javac \*.java

**2. Buat JAR dengan Manifest**

Buat file manifest.txt:

Isi Teks Denagn :

Main-Class: ManajemenKeuanganPribadi

Buat JAR:

Terminal :

jar cfm ManajemenKeuanganPribadi.jar manifest.txt \*.class

**3. Jalankan Program**

Terminal :

java -jar ManajemenKeuanganPribadi.jar

**Script untuk Menjalankan Aplikasi**

**Windows: runManajemenKeuangan.bat**

@echo off

java -jar ManajemenKeuanganPribadi.jar

pause

**Linux/Mac: runManajemenKeuangan.sh**

#!/bin/bash

java -jar ManajemenKeuanganPribadi.jar

read -p "Tekan Enter untuk keluar..."