

PROYEK AKHIR SEMESTER GASAL



NAMA : VINA SUKASIH
KELAS : X PPLG 1
NIS : 258741

JUDUL PROYEK : APLIKASI SMARTFINANCE TRACKER
(Aplikasi Manajemen Keuangan Sederhana)

PENGEMBANGAN PERANGKAT LUNAK DAN GIM
SMK NEGERI 1 KANDEMAN
TAHUN PELAJARAN 2025/2026

DESKRIPSI PROYEK

SmartFinance Tracker adalah sebuah aplikasi console berbasis C# yang berfungsi sebagai alat bantu pencatatan keuangan sederhana. Melalui aplikasi ini, pengguna dapat mencatat pemasukan serta pengeluaran, melihat daftar transaksi yang telah dimasukkan, dan menampilkan ringkasan saldo akhir secara otomatis. Semua transaksi disimpan secara lokal selama program berjalan tanpa memerlukan database atau koneksi internet.

Melalui aplikasi ini, pengguna dapat:

1. Menambahkan transaksi pemasukan dan pengeluaran.
2. Melihat riwayat transaksi yang telah dicatat.
3. Melihat laporan keuangan berupa total pemasukan, total pengeluaran, dan saldo akhir.
4. Menggunakan navigasi menu interaktif berbasis teks.

Proyek ini mengimplementasikan **konsep dasar pemrograman** seperti:

1. **Variabel dan Tipe Data** Digunakan untuk menyimpan data transaksi seperti jenis, nominal, dan keterangan. Contoh: int, double, string.
2. **Operator** Penggunaan operator membantu proses perhitungan dan logika secara efisien.
3. **Perulangan (looping)** Program menggunakan for untuk menampilkan dan menghitung transaksi, serta do-while untuk menjalankan menu secara berulang hingga pengguna memilih keluar.
4. **Percabangan** Penggunaan percabangan memungkinkan program merespon input yang berbeda secara dinamis.
5. **Array atau List** Digunakan sebagai struktur data, data transaksi disimpan dalam array seperti jenisTransaksi[], nominalTransaksi[], dan keterangan[].
6. **Fungsi/Method** Program menggunakan pemecahan fungsi seperti: Loading() untuk animasi loading, Tampilkan Transaksi() untuk menampilkan daftar transaksi, TampilkanRingkasan() untuk menghitung dan menampilkan total keuangan.
7. **Input dan Output Console** Input diterima menggunakan Console.ReadLine() dan output ditampilkan dengan Console.WriteLine() untuk interaksi pengguna.

RINGKASAN TEORI C# YANG DIGUNAKAN

1. Variabel dan Tipe Data

Digunakan untuk menyimpan data transaksi seperti jenis, nominal, dan keterangan. Contoh: int, double, string.

```
string[] jenisTransaksi = new string[100];  
double[] nominalTransaksi = new double[100];  
string[] keterangan = new string[100];  
int jumlahTransaksi = 0;  
int menu;
```

- string → menyimpan teks (jenis transaksi & keterangan).
- int → menyimpan bilangan bulat (jumlah transaksi, pilihan menu).
- double → menyimpan angka desimal (nominal uang).

2. Operator

Penggunaan operator membantu proses perhitungan dan logika secara efisien.

Contoh:

```
totalPemasukan += transaksi[i];  
jumlahTransaksi++;
```

Jenis operator yang digunakan:

- **Aritmatika:** +, -
- **Perbandingan:** ==
- **Penugasan:** =
- **Logika:** !
- **Increment:** ++

3. Percabangan (Kondisi)

Penggunaan percabangan memungkinkan program merespon input yang berbeda secara dinamis.

Contoh:

```
if (jenis[i].ToLower() == "pemasukan")  
    totalPemasukan += transaksi[i];
```

```
switch(menu)  
{  
    case 1:  
        // Tambah transaksi  
        break;  
}
```

4. Perulangan (Looping)

Program menggunakan for untuk menampilkan dan menghitung transaksi, serta do-while untuk menjalankan menu secara berulang hingga pengguna memilih keluar.

Contoh for:

```
for (int i = 0; i < jumlah; i++)  
{  
    Console.WriteLine($"{i + 1}. [{jenis[i]}] Rp {nominal[i]} - {keterangan[i]}");  
}
```

Contoh do-while:

```
do  
{  
    // menu program  
} while (menu != 4);
```

5. Array atau List

Data transaksi disimpan dalam array seperti jenisTransaksi[], nominalTransaksi[], dan keterangan[].

Contoh:

```
string[] jenisTransaksi = new string[100];  
double[] nominalTransaksi = new double[100];  
string[] keterangan = new string[100];
```

6. Fungsi atau Method

Program menggunakan pemecahan fungsi seperti: Loading() untuk animasi loading, TampilkanRingkasan() untuk menghitung dan menampilkan total keuangan. Contoh:

```
static void Loading(string teks)
{
    Console.Write(teks);
    for (int i = 0; i < 3; i++)
    {
        Console.Write(".");
        Thread.Sleep(300);
    }
}

static void TampilkanRingkasan(double[] transaksi, string[]
jenis, int jumlah)
```

7. Input dan Output

Input dan Output Console input diterima menggunakan Console.ReadLine() dan output ditampilkan dengan Console.WriteLine() untuk interaksi pengguna. Contoh Input:

```
Console.Write("Nominal (Rp): ");
nominalTransaksi[jumlahTransaksi]
double.Parse(Console.ReadLine());
```

Contoh Output:

```
Console.WriteLine("=== RINGKASAN KEUANGAN ===");
```

PERANCANGAN PROGRAM

Rancangan Tampilan Aplikasi SmartFinance Tracker

=====

WELCOME TO SMARTFINANCE TRACKER

=====

=== MENU UTAMA ===

1. Tambah Transaksi
2. Lihat Riwayat Transaksi
3. Lihat Ringkasan Saldo
4. Keluar

Pilih menu (1-4): 1

=== TAMBAH TRANSAKSI ===

Jenis (Pemasukan/Pengeluaran): Pemasukan

Nominal (Rp): 1.500.000

Keterangan: gaji bulan november

Menyimpan data...

Transaksi berhasil disimpan!

=== MENU UTAMA ===

1. Tambah Transaksi
2. Lihat Riwayat Transaksi
3. Lihat Ringkasan Saldo
4. Keluar

Pilih menu (1-4): 1

=== TAMBAH TRANSAKSI ===

Jenis (Pemasukan/Pengeluaran): Pengeluaran

Nominal (Rp): 245.000

Keterangan: untuk membeli beras dan buah

Menyimpan data...

Transaksi berhasil disimpan!

=== MENU UTAMA ===

1. Tambah Transaksi
2. Lihat Riwayat Transaksi
3. Lihat Ringkasan Saldo
4. Keluar

Pilih menu (1-4): 3

=== RINGKASAN KEUANGAN ===

Total Pemasukan : Rp 1500000

Total Pengeluaran : Rp 245000

Saldo Akhir : Rp 1255000

=== MENU UTAMA ===

1. Tambah Transaksi
2. Lihat Riwayat Transaksi
3. Lihat Ringkasan Saldo
4. Keluar

Pilih menu (1-4): 4

Menutup aplikasi...

Terima kasih telah menggunakan SmartFinance!

IMPLEMENTASI PROGRAM

Kode program utama

```
using System;
```

```
namespace Projek_Akhir_SmartFinance_vinasukasih_xpplgl
```

```
{
```

```
    internal class Program
```

```
    {
```

```
        // Fungsi untuk menampilkan teks dengan efek loading
        static void Loading(string teks)
```

```
        {
```

```
            Console.Write(teks);
```

```
            for (int i = 0; i < 3; i++)
```

```
            {
```

```
                Console.Write(".");
```

```
                Thread.Sleep(300);
```

```
            }
```

```
            Console.WriteLine();
```

```
        }
```

```
        // Fungsi untuk menampilkan ringkasan pemasukan, pengeluaran, dan saldo
        static void TampilkanRingkasan(double[] transaksi, string[] jenis, int jumlah)
```

```
        {
```

```
            double totalPemasukan = 0, totalPengeluaran = 0;
```

```
            // Menghitung total pemasukan & total total pengeluaran
```

```
            for (int i = 0; i < jumlah; i++)
```

```
            {
```

```
                if (jenis[i].ToLower() == "pemasukan")
```

```
                    totalPemasukan += transaksi[i];
```

```
                else if (jenis[i].ToLower() == "pengeluaran")
```

```
                    totalPengeluaran += transaksi[i];
```

```
            }
```

```
            double saldoAkhir = totalPemasukan - totalPengeluaran;
```

```
            // Menampilkan hasil ringkasan
```

```
            Console.ForegroundColor = ConsoleColor.Yellow;
```

```
            Console.WriteLine("\n=== RINGKASAN KEUANGAN ===");
```

```
            Console.ResetColor();
```

```
            Console.WriteLine($"Total Pemasukan : Rp {totalPemasukan}");
```

```
            Console.WriteLine($"Total Pengeluaran : Rp {totalPengeluaran}");
```

```
            Console.WriteLine($"Saldo Akhir : Rp {saldoAkhir}");
```

```
        }
```

```

// Fungsi untuk menampilkan semua transaksi
static void TampilkanTransaksi(string[] jenis, double[] nominal, string[]
keterangan, int jumlah)
{
    Console.ForegroundColor = ConsoleColor.Cyan;
    Console.WriteLine("\n=== RIWAYAT TRANSAKSI ===");
    Console.ResetColor();

    if (jumlah == 0) // Jika belum ada transaksi
    {
        Console.WriteLine("Belum ada transaksi yang dicatat.");
        return;
    }

    // Loop untuk menampilkan semua transaksi
    for (int i = 0; i < jumlah; i++)
    {
        Console.WriteLine($"{i + 1}. [{jenis[i]}] Rp {nominal[i]} -
{keterangan[i]}");
    }
}

static void Main(string[] args)
{
    // Array sebagai penyimpanan data transaksi
    string[] jenisTransaksi = new string[100];
    double[] nominalTransaksi = new double[100];
    string[] keterangan = new string[100];
    int jumlahTransaksi = 0;

    int menu;

    // Tampilan awal aplikasi
    Console.ForegroundColor = ConsoleColor.Blue;

    Console.WriteLine("=====");
    Console.WriteLine(" WELCOME TO SMARTFINANCE TRACKER
");

    Console.WriteLine("=====");
    Console.ResetColor();

    // Perulangan utama menu program
    do
    {
        // Menampilkan menu
        Console.WriteLine("\n=== MENU UTAMA ===");
        Console.WriteLine("1. Tambah Transaksi");
        Console.WriteLine("2. Lihat Riwayat Transaksi");
        Console.WriteLine("3. Lihat Ringkasan Saldo");
    }
}

```

```

        Console.WriteLine("4. Keluar");
        Console.Write("Pilih menu (1-4): ");

        // Validasi input menu
        if (!int.TryParse(Console.ReadLine(), out menu))
        {
            Console.WriteLine("Input tidak valid!");
            continue;
        }

        switch (menu)
        {
            case 1:
                // Input data transaksi
                Console.ForegroundColor =
ConsoleColor.Green;
                Console.WriteLine("\n==== TAMBAH
TRANSAKSI ====");
                Console.ResetColor();

                Console.Write("Jenis (Pemasukan/Pengeluaran):
");
                jenisTransaksi[jumlahTransaksi] =
Console.ReadLine();

                Console.Write("Nominal (Rp): ");
                nominalTransaksi[jumlahTransaksi] =
double.Parse(Console.ReadLine());

                Console.Write("Keterangan: ");
                keterangan[jumlahTransaksi] =
Console.ReadLine();

                jumlahTransaksi++; // Menambah jumlah data
yang tersimpan

                Loading("Menyimpan data");
                Console.ForegroundColor =
ConsoleColor.Green;
                Console.WriteLine("Transaksi berhasil
disimpan!");
                Console.ResetColor();
                break;

            case 2:
                // Menampilkan riwayat transaksi
                TampilkanTransaksi(jenisTransaksi,
nominalTransaksi, keterangan, jumlahTransaksi);
                break;

```



```

                                case 3:
                                    // Menampilkan ringkasan saldo
                                    TampilkanRingkasan(nominalTransaksi,
jenisTransaksi, jumlahTransaksi);
                                    break;

                                case 4:
                                    // Keluar dari program
                                    Console.ForegroundColor =
ConsoleColor.Yellow;
                                    Loading("\nMenutup aplikasi");
                                    Console.WriteLine("Terima kasih telah
menggunakan SmartFinance!");
                                    Console.ResetColor();
                                    break;

                                default:
                                    Console.WriteLine("Pilihan tidak valid!");
                                    break;
                            }
                        } while (menu != 4); // Program berjalan terus sampai memilih keluar
                    }
                }
            }

```

Penjelasan Kode Program

1. Deklarasi Namespace dan Class using System;

```

namespace Projek_Akhir_SmartFinance_vinasukasih_xpplgl
{
    internal class Program
    }

```

- using System; → digunakan agar kita bisa memakai perintah dasar C# seperti Console.WriteLine() dan Console.ReadLine().
- namespace Projek_Akhir_SmartFinance → ruang lingkup (wadah) program agar tidak bentrok dengan program lain.
- class Program → kelas utama tempat kode program dijalankan.

2. Method Loading()

```

static void Loading(string teks)

```

fungsi ini menampilkan animasi loading dengan menambahkan titik tiga secara bertahap.

Fungsinya:

- Agar tampilan lebih menarik dan tidak terlalu kaku.
- Memberikan efek seolah-olah sedang diproses.

3. Method TampilkanRingkasan()

```
static void TampilkanRingkasan(double[] transaksi, string[]  
jenis, int jumlah)
```

- transaksi[] → Array yang menyimpan nominal uang.
- jenis[] → Array menyimpan jenis transaksi (pemasukan/pengeluaran).
- jumlah → jumlah data transaksi yang tersimpan.

Fungsi:

- Menghitung total pemasukan
- Menghitung total pengeluaran
- Menampilkan saldo akhir

4. Method TampilkanTransaksi()

```
static void TampilkanRingkasan(double[] transaksi, string[]  
jenis, int jumlah)
```

Fungsinya: Menampilkan daftar transaksi yang tersimpan, dan Menampilkan nomor urut, jenis, nominal, dan keterangan.

Parameter:

- jenis[] → menyimpan tulisan “Pemasukan” / “Pengeluaran”
- nominal[] → menyimpan angka uang
- keterangan[] → catatan transaksi
- jumlah → berapa data yang sudah dimasukan

5. Method Main() – Inti Program

5.1 Deklarasi Variabel dan Array

```
string[] jenisTransaksi = new string[100];  
double[] nominalTransaksi = new double[100];  
string[] keterangan = new string[100];  
int jumlahTransaksi = 0;  
int menu;
```

Penjelasan:

- jenisTransaksi[] menggunakan tipe string – Untuk menyimpan jenis transaksi (“pemasukan” / “pengeluaran”).
- nominalTransaksi[] menggunakan tipe double – Untuk menyimpan nilai uang.
- keterangan[] menggunakan string – Untuk menyimpan deskripsi transaksi.
- jumlahTransaksi[] menggunakan tipe int – Untuk menghitung data, menentukan index array
- menu menggunakan tipe int – untuk menyimpan pilihan menu dari user.

5.2 Tampilan Awal

```
Console.WriteLine("WELCOME TO SMARTFINANCE TRACKER");
```

- Muncul hanya 1x di awal program

- Ditampilkan dalam warna untuk memperndah tampilan

5.3 Perulangan Menu Utama

```
do
{
...
} while (menu != 4);
```

- ❖ Untuk menjaga agar program terus berjalan sampai user memilih keluar

5.4 Validasi Input Menu

```
if (!int.TryParse(Console.ReadLine(), out menu))
{
    Console.WriteLine("Input tidak valid!");
    continue;
}
```

- ❖ Untuk mencegah error jika input bukan angka.
- ❖ Untuk membuat program lebih aman.

5.5 Switch Menu

switch(menu)

Penjelasan:

Case Fungsi

1. Tambah transaksi
 2. Lihat riwayat transaksi
 3. Tampilkan ringkasan saldo
 4. Keluar program
- default Menangani input salah

5.6 Penjelasan Tambahan Transaksi

```
Console.Write("Jenis:");
jenisTransaksi[jumlahTransaksi] = Console.ReadLine();
```

- ❖ Proses:
 - Data disimpan ke array sesuai urutan
 - Index ditentukan oleh jumlahTransaksi
 - Setelah input selesai → index ditambah

Tampilan Output Hasil Program

=====

WELCOME TO SMARTFINANCE TRACKER

=====

=== MENU UTAMA ===

1. Tambah Transaksi
2. Lihat Riwayat Transaksi
3. Lihat Ringkasan Saldo
4. Keluar

Pilih menu (1-4): 1

=== TAMBAH TRANSAKSI ===

Jenis (Pemasukan/Pengeluaran): Pemasukan

Nominal (Rp): 1.500.000

Keterangan: gaji bulan november

Menyimpan data...

Transaksi berhasil disimpan!

=== MENU UTAMA ===

1. Tambah Transaksi
2. Lihat Riwayat Transaksi
3. Lihat Ringkasan Saldo
4. Keluar

Pilih menu (1-4): 1

=== TAMBAH TRANSAKSI ===

Jenis (Pemasukan/Pengeluaran): Pengeluaran

Nominal (Rp): 245.000

Keterangan: untuk membeli beras dan buah

Menyimpan data...

Transaksi berhasil disimpan!

=== MENU UTAMA ===

1. Tambah Transaksi
2. Lihat Riwayat Transaksi
3. Lihat Ringkasan Saldo
4. Keluar

Pilih menu (1-4): 3

=== RINGKASAN KEUANGAN ===

Total Pemasukan : Rp 1500000

Total Pengeluaran : Rp 245000

Saldo Akhir : Rp 1255000

=== MENU UTAMA ===

1. Tambah Transaksi
2. Lihat Riwayat Transaksi
3. Lihat Ringkasan Saldo
4. Keluar

Pilih menu (1-4): 4

Menutup aplikasi...

Terima kasih telah menggunakan SmartFinance!

LAMPIRAN

Kode Program Lengkap

```
8 namespace Projek_Akhir_SmartFinance_vinasukasih_xpplg1
9 {
10     // 0 references
11     internal class Program
12     {
13         // Fungsi untuk menampilkan teks dengan efek loading
14         // 2 references
15         static void Loading(string teks)
16         {
17             Console.Write(teks);
18             for (int i = 0; i < 3; i++)
19             {
20                 Console.Write(".");
21                 Thread.Sleep(300);
22             }
23             Console.WriteLine();
24         }
25
26         // Fungsi untuk menampilkan ringkasan pemasukan, pengeluaran, dan saldo
27         // 1 reference
28         static void TampilkanRingkasan(double[] transaksi, string[] jenis, int jumlah)
29         {
30             double totalPemasukan = 0, totalPengeluaran = 0;
31
32             // Menghitung total pemasukan & total total pengeluaran
33             for (int i = 0; i < jumlah; i++)
34             {
35                 if (jenis[i].ToLower() == "pemasukan")
36                     totalPemasukan += transaksi[i];
37
38                 else if (jenis[i].ToLower() == "pengeluaran")
39                     totalPengeluaran += transaksi[i];
40             }
41
42             double saldoAkhir = totalPemasukan - totalPengeluaran;
43
44             // Menampilkan hasil ringkasan
45             Console.ForegroundColor = ConsoleColor.Yellow;
46             Console.WriteLine("\n=== RINGKASAN KEUANGAN ===");
47             Console.ResetColor();
48             Console.WriteLine($"Total Pemasukan : Rp {totalPemasukan}");
49             Console.WriteLine($"Total Pengeluaran : Rp {totalPengeluaran}");
50             Console.WriteLine($"Saldo Akhir : Rp {saldoAkhir}");
51         }
52
53         // Fungsi untuk menampilkan semua transaksi
54         // 1 reference
55         static void TampilkanTransaksi(string[] jenis, double[] nominal, string[] keterangan, int jumlah)
56         {
57             Console.ForegroundColor = ConsoleColor.Cyan;
58             Console.WriteLine("\n=== RIWAYAT TRANSAKSI ===");
59             Console.ResetColor();
60
61             if (jumlah == 0) // Jika belum ada transaksi
62             {
63                 Console.WriteLine("Belum ada transaksi yang dicatat.");
64                 return;
65             }
66
67             // Loop untuk menampilkan semua transaksi
68         }
69     }
70 }
```

```

64     for (int i = 0; i < jumlah; i++)
65     {
66         Console.WriteLine($"{i + 1}. [{jenis[i]}] Rp {nominal[i]} - {keterangan[i]}");
67     }
68 }
69
70 // references
71 static void Main(string[] args)
72 {
73     // Array sebagai penyimpanan data transaksi
74     string[] jenisTransaksi = new string[100];
75     double[] nominalTransaksi = new double[100];
76     string[] keterangan = new string[100];
77     int jumlahTransaksi = 0;
78
79     int menu;
80
81     // Tampilan awal aplikasi
82     Console.ForegroundColor = ConsoleColor.Blue;
83     Console.WriteLine("=====");
84     Console.WriteLine("  WELCOME TO SMARTFINANCE TRACKER ");
85     Console.WriteLine("=====");
86     Console.ResetColor();
87
88     // Perulangan utama menu program
89     do
90     {
91         // Menampilkan menu
92         Console.WriteLine("\n=== MENU UTAMA ===");
93         Console.WriteLine("1. Tambah Transaksi");
94
95         Console.WriteLine("2. Lihat Riwayat Transaksi");
96         Console.WriteLine("3. Lihat Ringkasan Saldo");
97         Console.WriteLine("4. Keluar");
98         Console.Write("Pilih menu (1-4): ");
99
100        // Validasi input menu
101        if (!int.TryParse(Console.ReadLine(), out menu))
102        {
103            Console.WriteLine("Input tidak valid!");
104            continue;
105        }
106
107        switch (menu)
108        {
109            case 1:
110                // Input data transaksi
111                Console.ForegroundColor = ConsoleColor.Green;
112                Console.WriteLine("\n=== TAMBAH TRANSAKSI ===");
113                Console.ResetColor();
114
115                Console.Write("Jenis (Pemasukan/Pengeluaran): ");
116                jenisTransaksi[jumlahTransaksi] = Console.ReadLine();
117
118                Console.Write("Nominal (Rp): ");
119                nominalTransaksi[jumlahTransaksi] = double.Parse(Console.ReadLine());
120
121                Console.Write("Keterangan: ");
122                keterangan[jumlahTransaksi] = Console.ReadLine();
123
124                jumlahTransaksi++; // Menambah jumlah data yang tersimpan

```

```

123 Loading("Menyimpan data");
124 Console.ForegroundColor = ConsoleColor.Green;
125 Console.WriteLine("Transaksi berhasil disimpan!");
126 Console.ResetColor();
127 break;
128
129
130 case 2:
131     // Menampilkan riwayat transaksi
132     TampilkanTransaksi(jenisTransaksi, nominalTransaksi, keterangan, jumlahTransaksi);
133     break;
134
135 case 3:
136     // Menampilkan ringkasan saldo
137     TampilkanRingkasan(nominalTransaksi, jenisTransaksi, jumlahTransaksi);
138     break;
139
140 case 4:
141     // Keluar dari program
142     Console.ForegroundColor = ConsoleColor.Yellow;
143     Loading("\nMenutup aplikasi");
144     Console.WriteLine("Terima kasih telah menggunakan SmartFinance!");
145     Console.ResetColor();
146     break;
147
148 default:
149     Console.WriteLine("Pilihan tidak valid!");
150     break;
151 }
152
153 } while (menu != 4); // Program berjalan terus sampai memilih keluar
154 }
155

```

ScreenShot Tampilan Program

=====

WELCOME TO SMARTFINANCE TRACKER

=====

=== MENU UTAMA ===

1. Tambah Transaksi
2. Lihat Riwayat Transaksi
3. Lihat Ringkasan Saldo
4. Keluar

Pilih menu (1-4): 1

=== TAMBAH TRANSAKSI ===

Jenis (Pemasukan/Pengeluaran): Pemasukan

Nominal (Rp): 1.500.000

Keterangan: gaji bulan november

Menyimpan data...

Transaksi berhasil disimpan!

=== MENU UTAMA ===

1. Tambah Transaksi
2. Lihat Riwayat Transaksi
3. Lihat Ringkasan Saldo
4. Keluar

Pilih menu (1-4): 1

=== TAMBAH TRANSAKSI ===

Jenis (Pemasukan/Pengeluaran): Pengeluaran

Nominal (Rp): 245.000

Keterangan: untuk membeli beras dan buah

Menyimpan data...

Transaksi berhasil disimpan!

=== MENU UTAMA ===

1. Tambah Transaksi
2. Lihat Riwayat Transaksi
3. Lihat Ringkasan Saldo
4. Keluar

Pilih menu (1-4): 3

=== RINGKASAN KEUANGAN ===

Total Pemasukan : Rp 1500000

Total Pengeluaran : Rp 245000

Saldo Akhir : Rp 1255000

=== MENU UTAMA ===

1. Tambah Transaksi
2. Lihat Riwayat Transaksi
3. Lihat Ringkasan Saldo
4. Keluar

Pilih menu (1-4): 4

Menutup aplikasi...

Terima kasih telah menggunakan SmartFinance!