JURNAL MODUL 8
RUNTIME CONFIGURATION DAN INTERNATIONALIZATION

Nama :

Alya Rabani (2311104076)

Dosen :

YUDHA ISLAMI SULISTYA

PROGRAM STUDI S1 REKAYASA PERANGKAT LUNAK
TELKOM UNIVERSITY PURWOKERTO
2025

1. IMPLEMENTASI RUNTIME CONFIGURATION
   Ini merupakan file JSON dari bank_transfer_config


```json
{
  "lang": "CONFIG1",
  "transfer": {
    "threshold": "CONFIG2",
    "low_fee": "CONFIG3",
    "high_fee": "CONFIG4"
  },
  "methods": CONFIG5,
  "confirmation": {
  "en": "CONFIG6",
  "id": "CONFIG7"
  }
}
```

Program ini menggunakan class BankTransferConfig yang secara otomatis membaca file konfigurasi JSON saat aplikasi dijalankan. Jika file tersebut tidak ditemukan, maka class akan membuat file baru berisi konfigurasi default. Dalam alur utama program, langkah pertama adalah menampilkan pesan berdasarkan bahasa yang ditentukan dalam konfigurasi (lang). Selanjutnya, program akan menghitung biaya transfer berdasarkan jumlah nominal yang dimasukkan dan ambang batas (threshold) yang telah ditentukan, lalu menampilkan total biaya yang harus dibayar. Setelah itu, pengguna diminta untuk memilih metode transfer dari daftar yang tersedia dan memberikan konfirmasi transaksi. Terakhir, program akan memvalidasi apakah input konfirmasi sesuai dengan nilai yang telah ditentukan dalam konfigurasi; jika sesuai, transaksi dianggap berhasil, jika tidak, maka transaksi dibatalkan.

Code program:

```csharp
using System;
using System.Collections.Generic;
using System.IO;
using System.Text.Json;

public class BankTransferConfig
{
    public string Lang { get; set; } = "en";
    public Transfer Transfer { get; set; } = new Transfer();
    public List<string> Methods { get; set; } = new List<string> { "RTO (real-time)",
"SKN", "RTGS", "BI FAST" };
    public Confirmation Confirmation { get; set; } = new Confirmation();

    private const string FilePath = "bank_transfer_config.json";

    public BankTransferConfig()
    {
        if (File.Exists(FilePath))
        {
            string json = File.ReadAllText(FilePath);
            var config = JsonSerializer.Deserialize<BankTransferConfig>(json);
            if (config != null)
            {
                Lang = config.Lang;
                Transfer = config.Transfer;
                Methods = config.Methods;
                Confirmation = config.Confirmation;
            }
        }
        else
        {
            // save default config if file not found
            SaveDefaultConfig();
        }
    }

    private void SaveDefaultConfig()
    {
        var options = new JsonSerializerOptions { WriteIndented = true };
        string json = JsonSerializer.Serialize(this, options);
        File.WriteAllText(FilePath, json);
    }
}
```

```csharp
public class Transfer
{
    public int Threshold { get; set; } = 25000000;
    public int Low_Fee { get; set; } = 6500;
    public int High_Fee { get; set; } = 15000;
}

public class Confirmation
{
    public string En { get; set; } = "yes";
    public string Id { get; set; } = "ya";
}


class Program
{
    static void Main()
    {
        BankTransferConfig config = new BankTransferConfig();

        string lang = config.Lang;
        string confirmationWord = (lang == "en") ? config.Confirmation.En :
config.Confirmation.Id;

        // Step 1: input jumlah transfer
        Console.WriteLine(lang == "en"
            ? "Please insert the amount of money to transfer:"
            : "Masukkan jumlah uang yang akan di-transfer:");
        int amount = int.Parse(Console.ReadLine());

        // Step 2: hitung fee dan total
        int fee = amount <= config.Transfer.Threshold ? config.Transfer.Low_Fee :
config.Transfer.High_Fee;
        int total = amount + fee;

        Console.WriteLine(lang == "en"
            ? $"Transfer fee = {fee}\nTotal amount = {total}"
            : $"Biaya transfer = {fee}\nTotal biaya = {total}");

        // Step 3: tampilkan metode transfer
        Console.WriteLine(lang == "en" ? "Select transfer method:" : "Pilih metode
transfer:");
        for (int i = 0; i < config.Methods.Count; i++)
        {
            Console.WriteLine($"{i + 1}. {config.Methods[i]}");
        }
```
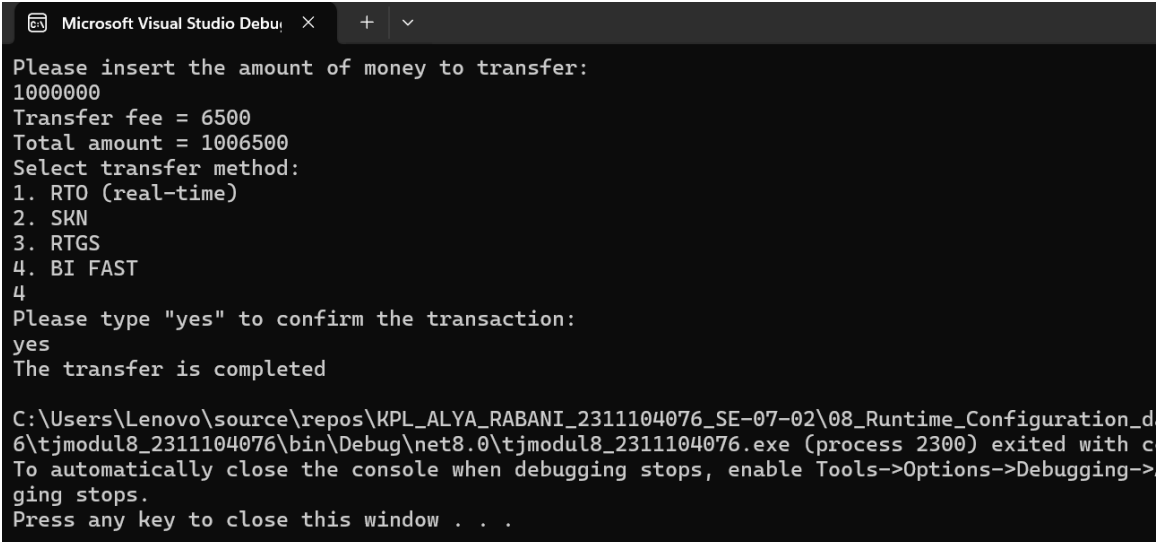
```
        Console.ReadLine(); // input pilihan metode (nggak digunakan lebih lanjut)

        // Step 4: konfirmasi
        Console.WriteLine(lang == "en"
            ? $"Please type \"{confirmationWord}\" to confirm the transaction:"
            : $"Ketik \"{confirmationWord}\" untuk mengkonfirmasi transaksi:");

        string userInput = Console.ReadLine();
        if (userInput.ToLower() == confirmationWord.ToLower())
        {
            Console.WriteLine(lang == "en"
                ? "The transfer is completed"
                : "Proses transfer berhasil");
        }
        else
        {
            Console.WriteLine(lang == "en"
                ? "Transfer is cancelled"
                : "Transfer dibatalkan");
        }
    }
}
}
```

Output yang dihasilkan:

2. Melakukan add, commit, dan push di github