

**LAPORAN JURNAL
MODUL 8**



Disusun Oleh :

Zaenarif Putra 'Ainurdin – 2311104049

Kelas :

SE-07-02

Dosen :

Yudha Islami Sulistya

**PROGRAM STUDI SOFTWARE ENGINEERING
DIREKTORAT KAMPUS PURWOKERTO
TELKOM UNIVERSITY
PURWOKERTO
2025**

I. Link Github

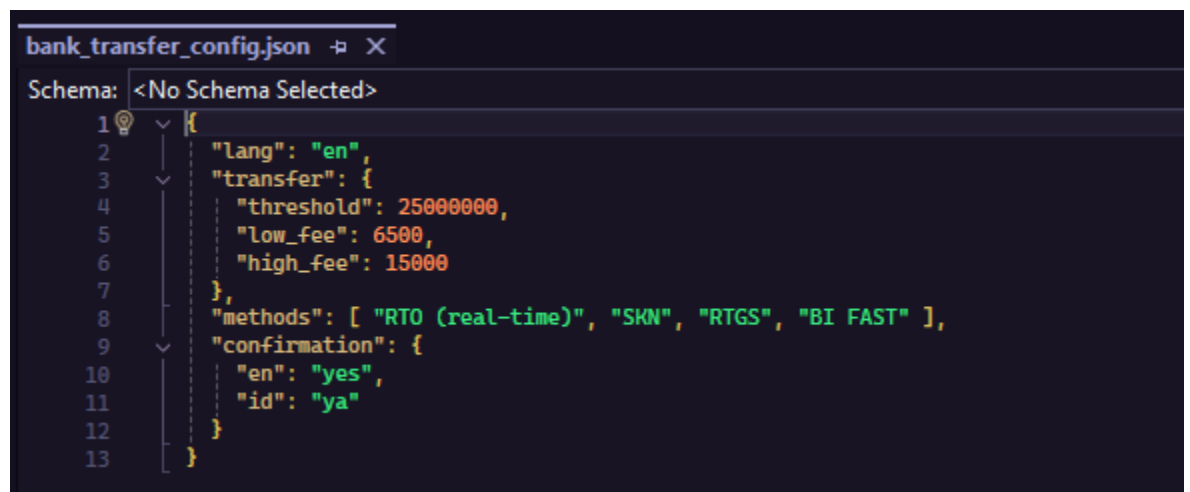
https://github.com/zaenarifputra/KPL_Zaenarif-Putra-Ainurdin_2311104049_S1-SE-07-02/tree/1a07bb8542100d8c1e27aa6ea200414e56281c04/07_Grammar-Based_Input_Processing_Parsing/TJ

II. Alur Pengerjaan

1. Membuat Project Awalan Dengan ConsoleApp, dan diberi nama *TjModul8_2311104049*



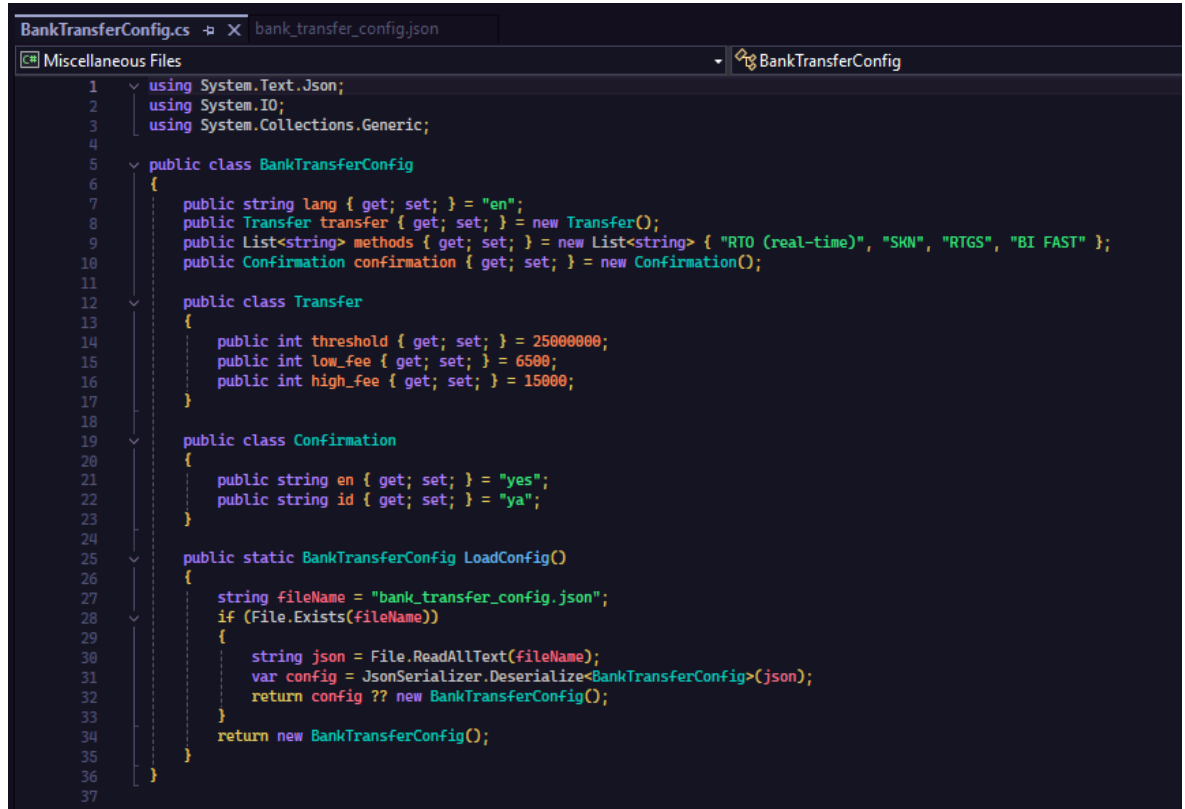
2. Setelah Selesai membuat project selanjutnya membuat file json yang dimana untuk konfigurasi pada projectnya file json diberi nama *bank_transfer_config*, Berikut sedikit penjelasan dan kegunaan pada file *bank_transfer_config.json* :



pada class *bank_transfer_config.json* yang dimana merupakan file konfigurasi yang digunakan untuk program yang menyimpan pengaturan sistem transfer bank agar mudah diubah tanpa perlu mengedit kode langsung. Di dalamnya terdapat pengaturan bahasa (*lang*) yang menentukan apakah tampilan program akan berbahasa Inggris ("en") atau Indonesia ("id"), lalu ada bagian transfer yang berisi nilai ambang (*threshold*) dan biaya transfer (*low_fee* dan *high_fee*) tergantung jumlah uang yang akan ditransfer. Bagian *methods* menyimpan daftar pilihan metode transfer seperti SKN atau BI FAST, dan *confirmation* menyimpan kata konfirmasi yang harus diketik pengguna untuk menyelesaikan transaksi. Semua data ini akan dibaca otomatis oleh program saat

dijalankan, sehingga programmer atau pengguna tidak perlu mengubah kode program untuk mengganti bahasa atau biaya—cukup ubah isi file JSON ini saja..

3. Setelah melakukan konfigurasi data json, selanjutnya membuat class BankTransferConfig.cs yang dimana digunakan sebagai konfigurasi lanjut agar bisa dipergunakan pada projectnya. Berikut penjelasan sederhana dari syntaxnya dan juga ssannya:



```
1 using System.Text.Json;
2 using System.IO;
3 using System.Collections.Generic;
4
5 public class BankTransferConfig
6 {
7     public string lang { get; set; } = "en";
8     public Transfer transfer { get; set; } = new Transfer();
9     public List<string> methods { get; set; } = new List<string> { "RTO (real-time)", "SKN", "RTGS", "BI FAST" };
10    public Confirmation confirmation { get; set; } = new Confirmation();
11
12    public class Transfer
13    {
14        public int threshold { get; set; } = 25000000;
15        public int low_fee { get; set; } = 6500;
16        public int high_fee { get; set; } = 15000;
17    }
18
19    public class Confirmation
20    {
21        public string en { get; set; } = "yes";
22        public string id { get; set; } = "ya";
23    }
24
25    public static BankTransferConfig LoadConfig()
26    {
27        string fileName = "bank_transfer_config.json";
28        if (File.Exists(fileName))
29        {
30            string json = File.ReadAllText(fileName);
31            var config = JsonSerializer.Deserialize<BankTransferConfig>(json);
32            return config ?? new BankTransferConfig();
33        }
34        return new BankTransferConfig();
35    }
36 }
37
```

syntax pada class berikut merupakan definisi kelas BankTransferConfig.cs yang digunakan untuk mengelola konfigurasi pengaturan transfer bank dalam sebuah aplikasi. Kelas ini memiliki properti untuk menyimpan informasi seperti bahasa yang digunakan (lang), detail transfer seperti batas minimum jumlah transfer dan biaya transfer (Transfer), serta metode transfer yang tersedia (methods). Ada juga pengaturan konfirmasi dalam dua bahasa (Inggris dan Indonesia) yang disimpan dalam kelas Confirmation. Metode statis LoadConfig() digunakan untuk membaca file JSON yang berisi konfigurasi ini dan mengembalikannya sebagai objek BankTransferConfig.cs, sehingga konfigurasi transfer bank bisa diatur dengan mudah berdasarkan file yang ada. Kode ini akan sangat berguna dalam proyek sistem manajemen transfer bank, di mana pengaturan dan biaya transfer dapat diubah atau disesuaikan tanpa perlu mengubah kode secara langsung.

4. Setelah melakukan semua konfigurasi baik dari class cs dan json selanjutnya mengkonfigurasi pada file Program.cs, berikut penjelasan sederhana dan ssannya:

```

Program.cs  BankTransferConfig.cs  bank_transfer_config.json
TjModul8_2311104049
using System;

class Program
{
    static void Main(string[] args)
    {
        var config = BankTransferConfig.LoadConfig();
        string lang = config.Lang;

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

        string? inputAmount = Console.ReadLine();
        if (!int.TryParse(inputAmount, out int amount))
        {
            Console.WriteLine("Invalid amount input.");
            return;
        }

        // Step 2: Calculate Fee & Total
        int fee = amount <= config.transfer.threshold
            ? config.transfer.low_fee
            : config.transfer.high_fee;
        int total = amount + fee;

        if (lang == "en")
        {
            Console.WriteLine($"Transfer fee = {fee}");
            Console.WriteLine($"Total amount = {total}");
        }
        else
        {
            Console.WriteLine($"Biaya transfer = {fee}");
            Console.WriteLine($"Total biaya = {total}");
        }

        // Step 3: Show transfer methods
        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]}");
        }

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

        string? confirmInput = Console.ReadLine();
        bool confirmed = (lang == "en" && confirmInput == config.confirmation.en) ||
            (lang == "id" && confirmInput == config.confirmation.id);

        Console.WriteLine();
        if (confirmed)
        {
            Console.WriteLine(lang == "en"
                ? "The transfer is completed"
                : "Proses transfer berhasil");
        }
        else
        {
            Console.WriteLine(lang == "en"
                ? "Transfer is cancelled"
                : "Transfer dibatalkan");
        }
    }
}

```

syntax main program di atas adalah program C# untuk melakukan simulasi transfer uang dengan menggunakan pengaturan dari file konfigurasi. Program dimulai dengan memuat data konfigurasi, lalu meminta pengguna memasukkan jumlah uang yang ingin ditransfer. Setelah itu, program menghitung biaya transfer berdasarkan jumlah uang—jika jumlahnya di bawah atau sama dengan batas tertentu, maka biaya rendah yang dikenakan, kalau lebih maka biaya tinggi. Program kemudian menampilkan total biaya, menawarkan pilihan metode transfer, dan meminta konfirmasi dari pengguna. Jika pengguna mengetik konfirmasi yang sesuai (dalam bahasa Inggris atau Indonesia tergantung pengaturan), maka transaksi dianggap berhasil, jika tidak maka dibatalkan. Semua proses ini ditampilkan dalam bahasa yang ditentukan di file konfigurasi, jadi program ini fleksibel digunakan untuk pengguna berbahasa Inggris atau Indonesia.

kemudian setelah selesai melakukan konfigurasi semua class dan berhasil berjalan dengan sempurna, dilakukannya commit seperti gambar berikut yang di mulai dari git add ., git commit -m “aplikasi bank transfer dengan runtime configuration”.

5. Setelah menyelesaikan semua dari ke-3 branch yang sudah dilakukan selanjutnya kita akan pull request dan juga merge menyatukan semua syntax yang ada pada branch 1 hingga 3 ke branch master yang dimana ketika sudah disatukan semuanya maka semua class akan disatukan kedalam branch master seperti berikut :

KPL_Zaenarif-Putra-Ainurdin_2311104049_S1SE-07-02 / 07_Grammar-Based_Input_Processing_Parsing / TJ / TjModul7_Kelompok7 /

● zaenarifputra menyelesaikan conflict merge putra dan putra2

Name	Last commit message
..	
DataMahasiswa2311104049.cs	menyelesaikan conflict merge putra dan putra2
GlossaryItem2311104049.cs	menambahkan class GlossaryItem2311104049
Program.cs	menyelesaikan conflict merge putra dan putra2
TeamMembers2311104049.cs	menyelesaikan conflict merge putra dan putra2
TJModul7_Kelompok7.csproj	menambahkan class DataMahasiswa2311104049
jurnal7_1_2311104049.json	menyelesaikan conflict merge putra dan putra2
jurnal7_2_2311104049.json	menyelesaikan conflict merge putra dan putra2
jurnal7_3_2311104049.json	menambahkan class GlossaryItem2311104049

Setelah semua program dan sudah merge menuju branch master modul7 sudah berhasil di selesaikan

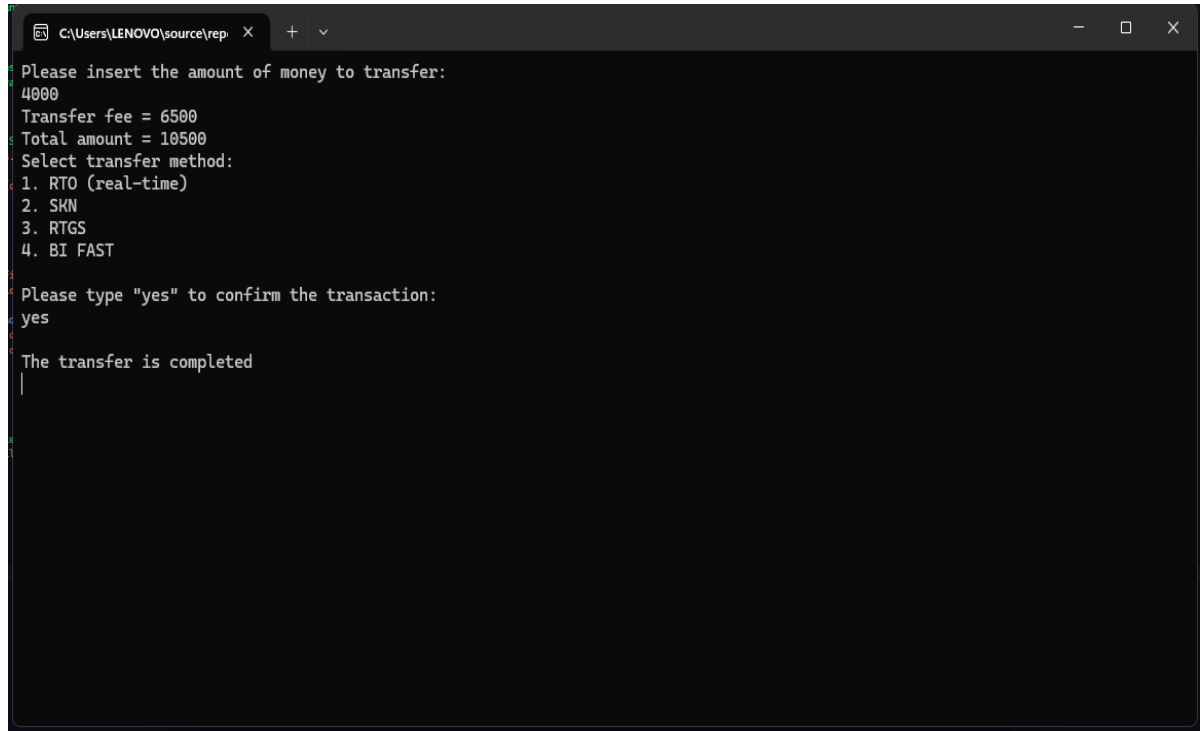
KPL_Zaenarif-Putra-Ainurdin_2311104049_S1SE-07-02 / 08_Runtime_Configuration_dan_Internationalization / TJ / TjModul8_2311104049 /

● zaenarifputra aplikasi bank transfer dengan runtime configuration

Name	Last commit message
..	
BankTransferConfig.cs	aplikasi bank transfer dengan runtime configuration
Program.cs	aplikasi bank transfer dengan runtime configuration
TJModul8_2311104049.csproj	aplikasi bank transfer dengan runtime configuration
bank_transfer_config.json	aplikasi bank transfer dengan runtime configuration

III. Hasil Running

1. Hasil Running Project modul8



```
C:\Users\LENOVO\source\rep x + v
Please insert the amount of money to transfer:
4000
Transfer fee = 6500
Total amount = 10500
Select transfer method:
1. RTO (real-time)
2. SKN
3. RTGS
4. BI FAST

Please type "yes" to confirm the transaction:
yes

The transfer is completed
|
```

Kesimpulan dari program TjModul8_2311104049 yang mensimulasikan sistem transfer bank dengan konfigurasi runtime menggunakan file JSON. File bank_transfer_config.json berfungsi untuk menyimpan pengaturan utama seperti bahasa, ambang batas transfer, biaya transfer, metode transfer, dan konfirmasi, sehingga memudahkan perubahan pengaturan tanpa mengubah kode program. Selanjutnya, class BankTransferConfig.cs dibuat untuk membaca dan mengelola file konfigurasi tersebut dengan efisien. Program utama yang terdapat di Program.cs memanfaatkan konfigurasi ini untuk menjalankan proses transfer berdasarkan input pengguna, dengan tampilan yang dinamis sesuai bahasa yang dipilih. Setelah semua komponen berhasil dikembangkan dan berfungsi dengan baik, proses commit dilakukan menggunakan Git, dan semua perubahan dari tiga branch digabungkan (merge) ke dalam branch master melalui proses pull request. Dengan demikian, seluruh proses pengembangan modul ini telah diselesaikan dengan baik dan sesuai dengan alur pengembangan perangkat lunak berbasis sistem kontrol versi.