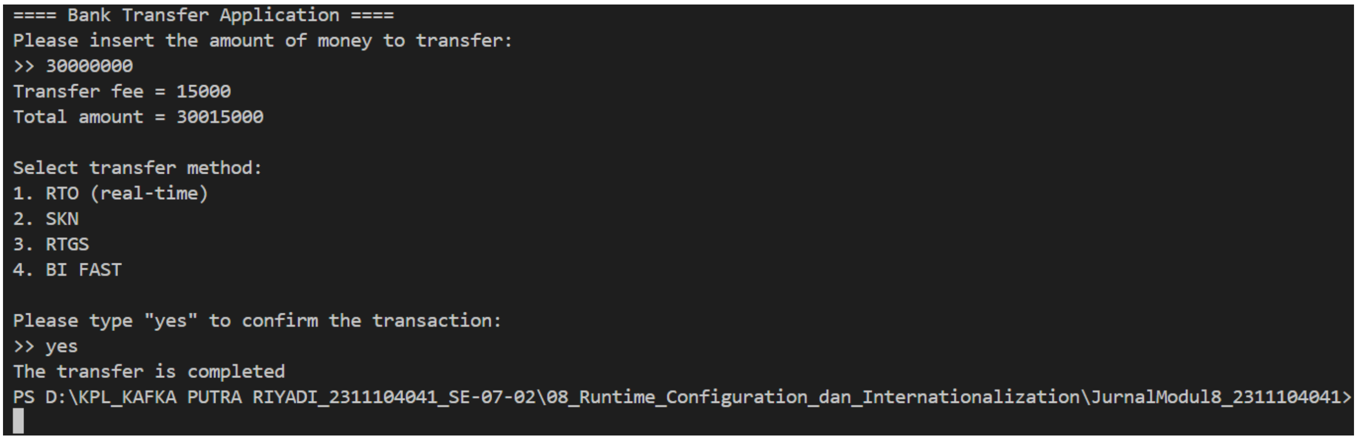
Nama : Kafka Putra Riyadi
NIM : 2311104041
Kelas : SE 07-02

link repo :
https://github.com/kafkaputrariyadi/KPL_KAFKA-PUTRA-RIYADI_2311104041_SE-07-02

outputannya :



```
==== Bank Transfer Application ====
Please insert the amount of money to transfer:
>> 30000000
Transfer fee = 15000
Total amount = 30015000

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
PS D:\KPL_KAFKA PUTRA RIYADI_2311104041_SE-07-02\08_Runtime_Configuration_dan_Internationalization\JurnalModul8_2311104041>
```

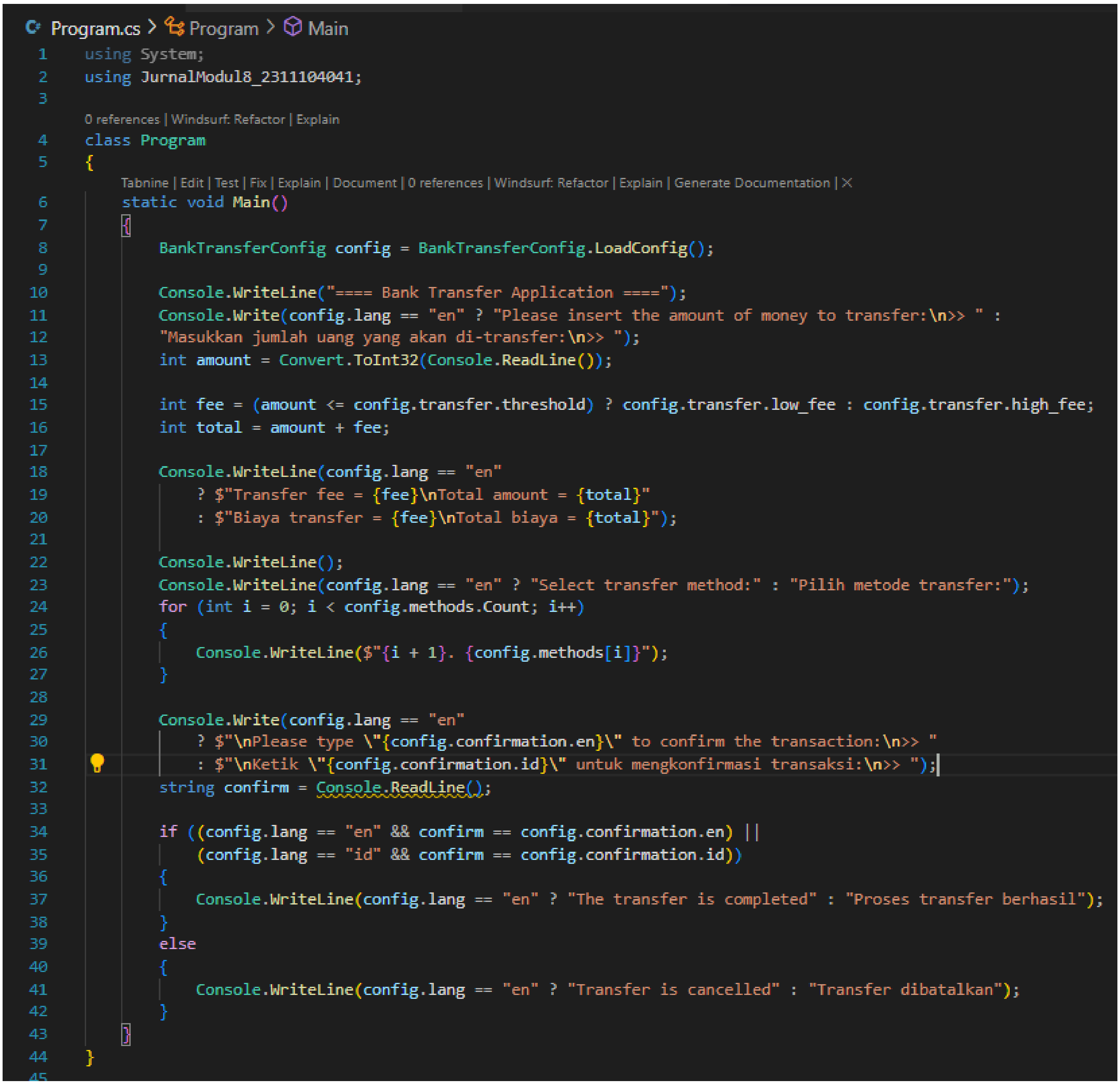
1. Program.cs

File ini berisi program utama aplikasi transfer bank.

Fungsinya yaitu :

1. Membaca konfigurasi dari file JSON menggunakan class `BankTransferConfig`.
2. Meminta input jumlah uang, lalu menghitung biaya transfer berdasarkan aturan threshold.
3. Menampilkan daftar metode transfer dan meminta konfirmasi dari user.
4. Memberikan hasil akhir apakah transfer berhasil atau dibatalkan, tergantung input user.

ScreenShoot codingan :


Program.cs > Program > Main
0 references | Windsurf: Refactor | Explain
Tabnine | Edit | Test | Fix | Explain | Document | 0 references | Windsurf: Refactor | Explain | Generate Documentation | ×

```csharp
using System;
using JurnalModul8_2311104041;

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

        Console.WriteLine("==== Bank Transfer Application ====");
        Console.Write(config.lang == "en" ? "Please insert the amount of money to transfer:\n>> " :
        "Masukkan jumlah uang yang akan di-transfer:\n>> ");
        int amount = Convert.ToInt32(Console.ReadLine());

        int fee = (amount <= config.transfer.threshold) ? config.transfer.low_fee : config.transfer.high_fee;
        int total = amount + fee;

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

        Console.WriteLine();
        Console.WriteLine(config.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.Write(config.lang == "en"
            ? $"\nPlease type \"{config.confirmation.en}\" to confirm the transaction:\n>> "
            : $"\nKetik \"{config.confirmation.id}\" untuk mengkonfirmasi transaksi:\n>> ");
        string confirm = Console.ReadLine();

        if ((config.lang == "en" && confirm == config.confirmation.en) ||
            (config.lang == "id" && confirm == config.confirmation.id))
        {
            Console.WriteLine(config.lang == "en" ? "The transfer is completed" : "Proses transfer berhasil");
        }
        else
        {
            Console.WriteLine(config.lang == "en" ? "Transfer is cancelled" : "Transfer dibatalkan");
        }
    }
}
```

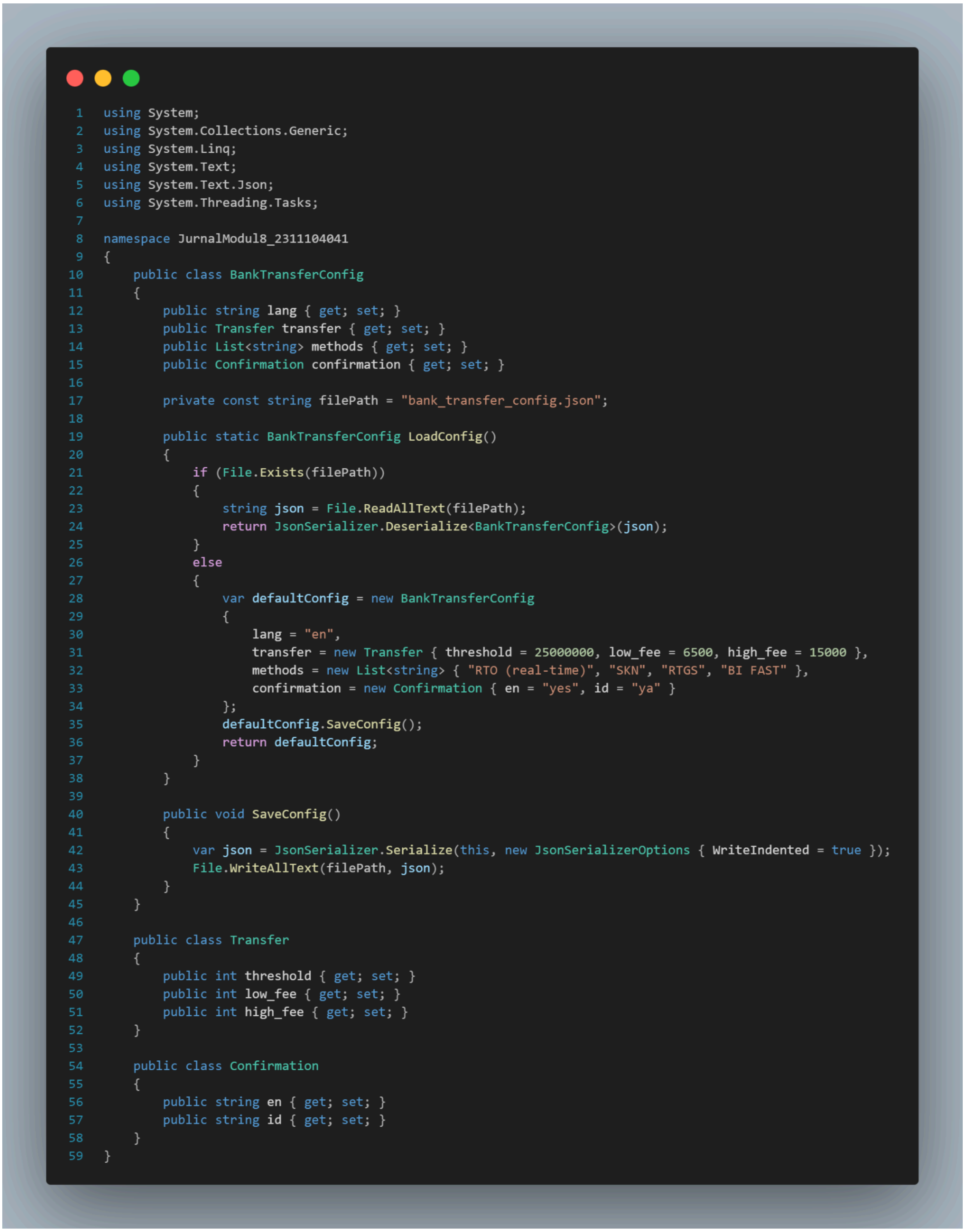
2. BankTransferConfig.cs

File ini berisi **class konfigurasi runtime**.
Fungsinya:

1. Membaca dan menyimpan pengaturan dari file `bank_transfer_config.json`.
2. Menyediakan nilai default jika file JSON belum ada.

3. Menyimpan data seperti: bahasa (`lang`), ambang batas (`threshold`), biaya transfer (`low_fee`, `high_fee`), metode transfer, dan teks konfirmasi.

Screenshoot codingan :



```csharp
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Text.Json;
using System.Threading.Tasks;

namespace JurnalModul8_2311104041
{
    public class BankTransferConfig
    {
        public string lang { get; set; }
        public Transfer transfer { get; set; }
        public List<string> methods { get; set; }
        public Confirmation confirmation { get; set; }

        private const string filePath = "bank_transfer_config.json";

        public static BankTransferConfig LoadConfig()
        {
            if (File.Exists(filePath))
            {
                string json = File.ReadAllText(filePath);
                return JsonSerializer.Deserialize<BankTransferConfig>(json);
            }
            else
            {
                var defaultConfig = new BankTransferConfig
                {
                    lang = "en",
                    transfer = new Transfer { threshold = 25000000, low_fee = 6500, high_fee = 15000 },
                    methods = new List<string> { "RTO (real-time)", "SKN", "RTGS", "BI FAST" },
                    confirmation = new Confirmation { en = "yes", id = "ya" }
                };
                defaultConfig.SaveConfig();
                return defaultConfig;
            }
        }

        public void SaveConfig()
        {
            var json = JsonSerializer.Serialize(this, new JsonSerializerOptions { WriteIndented = true });
            File.WriteAllText(filePath, json);
        }
    }

    public class Transfer
    {
        public int threshold { get; set; }
        public int low_fee { get; set; }
        public int high_fee { get; set; }
    }

    public class Confirmation
    {
        public string en { get; set; }
        public string id { get; set; }
    }
}
```