

LAPORAN PRAKTIKUM
JURNAL MODUL 8



Nama :

Aulia Jasifa Br Ginting 2311104060

S1SE-07-02

Dosen :

Yudha Islami Sulistya, S.Kom., M.Kom

PROGRAM STUDI S1 REKAYASA PERANGKAT LUNAK

FAKULTAS INFORMATIKA

TELKOM UNIVERSITY PURWOKERTO

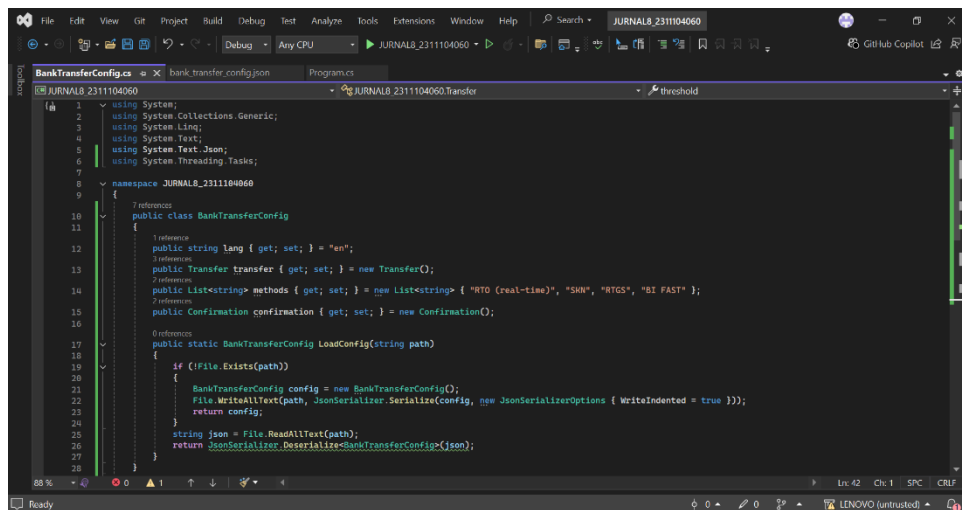
2025

I. Link GitHub

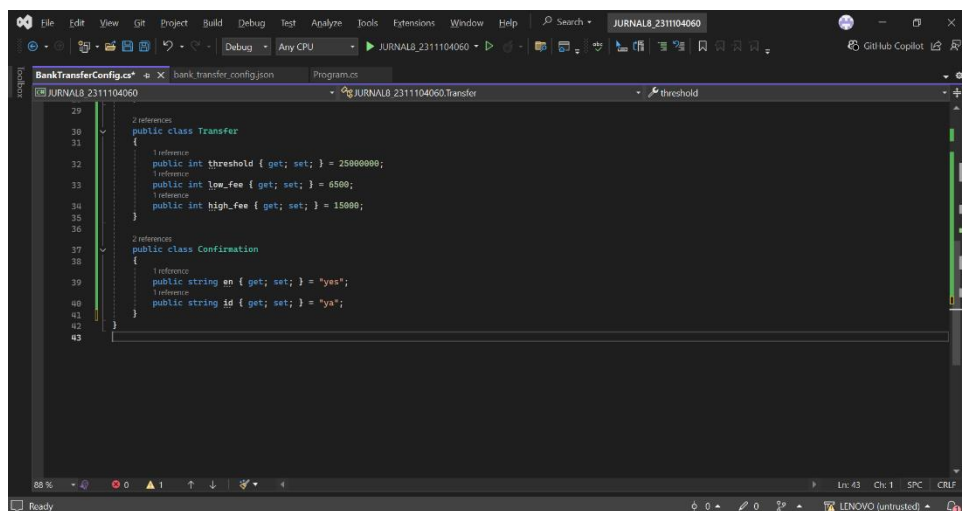
https://github.com/auliajsf06/KPL_Aulia-Jasifa-Br-Ginting_2311104060_SE-07-02

II. IMPLEMENTASI RUNTIME CONFIGURATION

A. Tambahkan class baru untuk memproses file konfigurasi dengan nama “BankTransferConfig”.

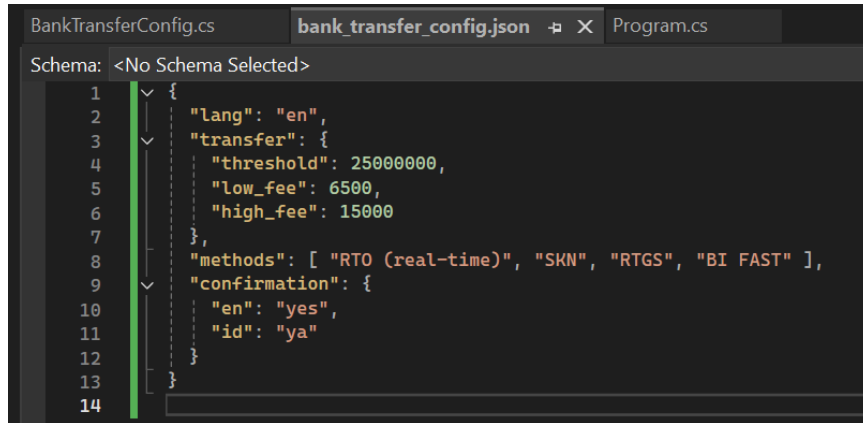


```
1 using System;
2 using System.Collections.Generic;
3 using System.Linq;
4 using System.Text;
5 using System.Text.Json;
6 using System.Threading.Tasks;
7
8 namespace JURNAL8_2311104060
9 {
10     7 references
11     public class BankTransferConfig
12     {
13         1 reference
14         public string lang { get; set; } = "en";
15         1 reference
16         public Transfer transfer { get; set; } = new Transfer();
17         2 references
18         public List<string> methods { get; set; } = new List<string> { "RT0 (real-time)", "SKN", "RTGS", "BI FAST" };
19         2 references
20         public Confirmation confirmation { get; set; } = new Confirmation();
21
22         0 references
23         public static BankTransferConfig LoadConfig(string path)
24         {
25             if (!File.Exists(path))
26             {
27                 BankTransferConfig config = new BankTransferConfig();
28                 File.WriteAllText(path, JsonSerializer.Serialize(config, new JsonSerializerOptions { WriteIndented = true }));
29                 return config;
30             }
31             string json = File.ReadAllText(path);
32             return JsonSerializer.Deserialize<BankTransferConfig>(json);
33         }
34     }
35 }
```



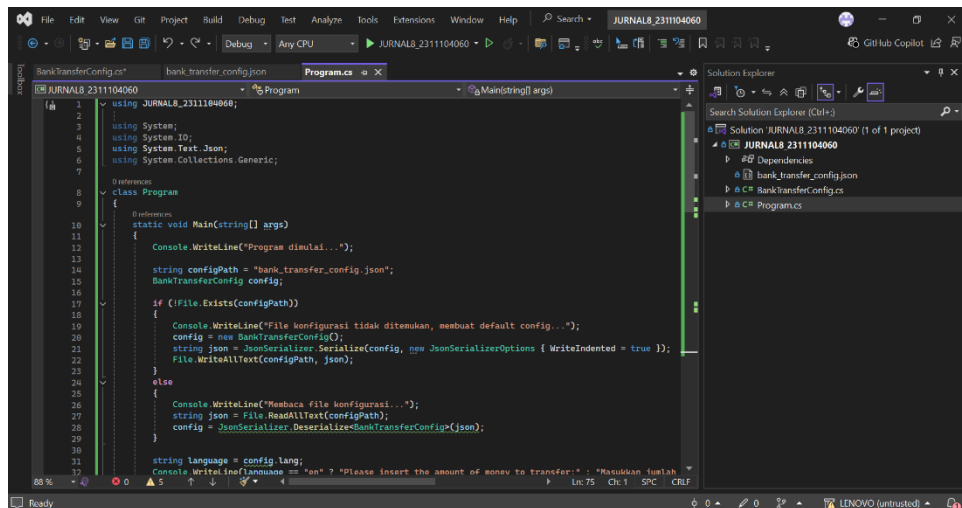
```
29
30     2 references
31     public class Transfer
32     {
33         1 reference
34         public int threshold { get; set; } = 25000000;
35         1 reference
36         public int low_fee { get; set; } = 6500;
37         1 reference
38         public int high_fee { get; set; } = 15000;
39     }
40
41     2 references
42     public class Confirmation
43     {
44         1 reference
45         public string an { get; set; } = "yes";
46         1 reference
47         public string id { get; set; } = "ya";
48     }
49 }
```

B. File konfigurasi yang digunakan adalah “bank_transfer_config.json” dengan format sebagai berikut ini:

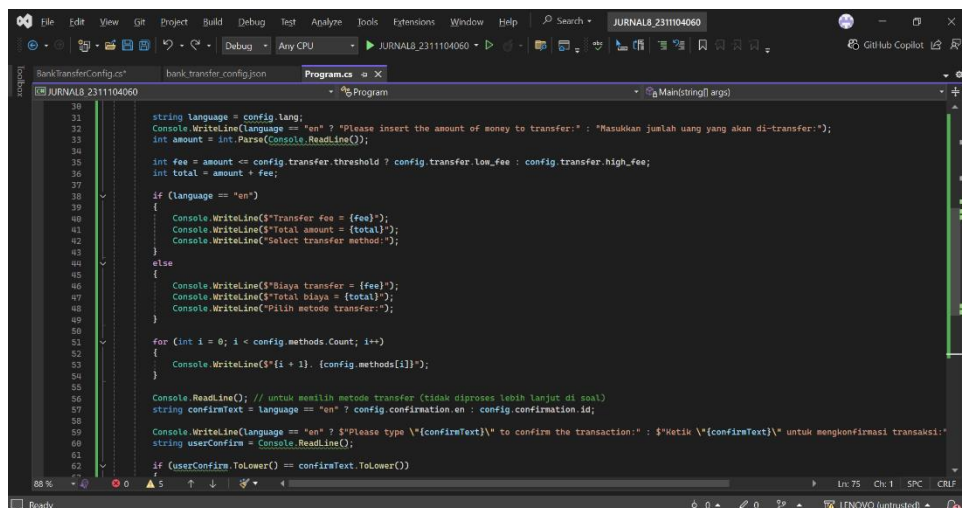


```
1 {
2     "lang": "en",
3     "transfer": {
4         "threshold": 25000000,
5         "low_fee": 6500,
6         "high_fee": 15000
7     },
8     "methods": [ "RTO (real-time)", "SKN", "RTGS", "BI FAST" ],
9     "confirmation": {
10        "en": "yes",
11        "id": "ya"
12    }
13 }
14 }
```

C. Class “Program.cs”



```
1 using JURNAL8_2311104060;
2 using System;
3 using System.IO;
4 using System.Text.Json;
5 using System.Collections.Generic;
6
7
8 class Program
9 {
10     static void Main(string[] args)
11     {
12         Console.WriteLine("Program dimulai...");
13
14         string configPath = "bank_transfer_config.json";
15         BankTransferConfig config;
16
17         if (!File.Exists(configPath))
18         {
19             Console.WriteLine("File konfigurasi tidak ditemukan, membuat default config...");
20             config = new BankTransferConfig();
21             string json = JsonSerializer.Serialize(config, new JsonSerializerOptions { WriteIndented = true });
22             File.WriteAllText(configPath, json);
23         }
24         else
25         {
26             Console.WriteLine("Membaca file konfigurasi...");
27             string json = File.ReadAllText(configPath);
28             config = JsonSerializer.Deserialize<BankTransferConfig>(json);
29         }
30
31         string language = config.lang;
32         Console.WriteLine(language == "en" ? "Please insert the amount of money to transfer:" : "Masukkan jumlah uang yang akan di-transfer:");
33     }
34 }
```



```
31 string language = config.lang;
32 Console.WriteLine(language == "en" ? "Please insert the amount of money to transfer:" : "Masukkan jumlah uang yang akan di-transfer:");
33 int amount = int.Parse(Console.ReadLine());
34
35 int fee = amount <= config.transfer.threshold ? config.transfer.low_fee : config.transfer.high_fee;
36 int total = amount + fee;
37
38 if (language == "en")
39 {
40     Console.WriteLine($"Transfer fee = {fee}");
41     Console.WriteLine($"Total amount = {total}");
42     Console.WriteLine("Select transfer method:");
43 }
44 else
45 {
46     Console.WriteLine($"Biaya transfer = {fee}");
47     Console.WriteLine($"Total biaya = {total}");
48     Console.WriteLine("Pilih metode transfer:");
49 }
50
51 for (int i = 0; i < config.methods.Count; i++)
52 {
53     Console.WriteLine($"{i + 1}. {config.methods[i]}");
54 }
55
56 Console.ReadLine(); // untuk memilih metode transfer (tidak diproses lebih lanjut di soal)
57 string confirmText = language == "en" ? config.confirmation.en : config.confirmation.id;
58
59 Console.WriteLine(language == "en" ? $"Please type \"{confirmText}\" to confirm the transaction:" : $"Ketik \"{confirmText}\" untuk mengkonfirmasi transaksi:");
60 string userConfirm = Console.ReadLine();
61
62 if (userConfirm.ToLower() == confirmText.ToLower())
```


tampilannya tanpa mengubah kode program, cukup dengan mengubah nilai pada file konfigurasi.

2. Class Konfigurasi (BankTransferConfig.cs)

Class `BankTransferConfig` dan class-class turunannya (`Transfer`, `Confirmation`) berfungsi untuk merepresentasikan struktur data dari file konfigurasi JSON ke dalam bentuk objek C#. Saat program dijalankan, file JSON dibaca dan di-deserialize menjadi objek dari class ini, sehingga data dapat digunakan langsung oleh program. Nilai default juga diberikan di properti class untuk berjaga-jaga jika file belum tersedia—sehingga aplikasi tetap dapat berjalan dengan konfigurasi awal.

3. Class Program Utama (Program.cs)

`Program.cs` adalah titik awal eksekusi aplikasi. Program ini pertama-tama memastikan file konfigurasi tersedia. Jika belum ada, maka dibuat secara otomatis dengan nilai default. Setelah file dibaca, program meminta pengguna untuk memasukkan nominal transfer. Berdasarkan nilai tersebut dan ambang batas (threshold), program menghitung biaya transfer (`low_fee` atau `high_fee`) dan total biaya. Kemudian, program menampilkan daftar metode transfer dan meminta konfirmasi dari pengguna untuk menyelesaikan transaksi. Bahasa dan pesan yang ditampilkan mengikuti nilai lang dari file konfigurasi. Aplikasi menjadi fleksibel dan dapat beradaptasi sesuai kebutuhan tanpa perlu memodifikasi kode sumber secara langsung.

III. KESIMPULAN

Praktikum ini memberikan pemahaman praktis tentang bagaimana sebuah aplikasi dapat menggunakan konfigurasi runtime eksternal (file JSON) untuk mengatur perilaku program tanpa perlu mengubah kode sumber. Dengan memisahkan konfigurasi dari logika program, aplikasi menjadi lebih fleksibel dan mudah disesuaikan. Selain itu, praktikan juga diajarkan penggunaan Git dan GitHub sebagai alat version control serta dokumentasi source code. Praktikum ini sangat berguna untuk membekali mahasiswa dengan konsep pemrograman yang baik, seperti modularisasi, penggunaan file konfigurasi, dan pengelolaan proyek menggunakan sistem versioning.