

**PRAKTIKUM KONSTRUKSI PERANGKAT LUNAK
TUGAS JURNAL 08**

Runtime Configuration & Internationalization



**Telkom
University**

disusun Oleh:
Alfian Mutakim
2211104024

SE0601

**S1 REKAYASA PERANGKAT LUNAK
FAKULTAS INFORMATIKA
TELKOM UNIVERSITY
2025**

1. IMPLEMENTASI RUNTIME CONFIGURATION

Dari master/main branch dan class utama, buatlah implementasi program yang memanfaatkan teknik Runtime Configuration dengan spesifikasi sebagai berikut ini:

- a. Tambahkan class baru untuk memproses file konfigurasi dengan nama “BankTransferConfig”.
- b. File konfigurasi yang digunakan adalah “covid_config.json” dengan format sebagai berikut:

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

- c. Nilai default yang digunakan apabila file config masih belum ada yaitu:
 - CONFIG1 => “en”
 - CONFIG2 => 25000000
 - CONFIG3 => 6500
 - CONFIG4 => 15000
 - CONFIG5 => [“RTO (real-time)”, “SKN”, “RTGS”, “BI FAST”]
 - CONFIG6 => “yes”
 - CONFIG7 => “ya”
- d. Aplikasi (program utama) memiliki alur atau fungsi sebagai berikut (yang menggunakan class config “BankTransferConfig”):
 - Pada saat program dijalankan program akan menampilkan pesan sebagai berikut tergantung nilai dari CONFIG1 atau JSON untuk key “lang”:
 1. CONFIG1 == “en” => “Please insert the amount of money to transfer:”
 2. CONFIG1 == “id” => “Masukkan jumlah uang yang akan di-transfer:”
 - Setelah itu aplikasi akan menampilkan biaya transfer dan total biaya yang akan dibayarkan dengan ketentuan sebagai berikut:
 1. Apabila jumlah yang di-transfer pada bagian sebelumnya (D-i) kurang dari atau sama dengan nilai CONFIG2 atau “threshold”, maka biaya transfer adalah CONFIG3 atau “low_fee”. Akan tetapi jika lebih

dari “threshold”, maka biaya transfer diambil dari nilai CONFIG4 atau “high_fee”.

2. Total biaya yang perlu dibayarkan adalah hasil penjumlahan dari jumlah uang yang akan ditransfer dan biaya transfer. Pesan output apabila CONFIG1 atau “lang” bernilai “en” adalah “Transfer fee = <biaya_transfer>” dan “Total amount = <nominal_transfer + biaya_transfer>”.
 3. Pesan output apabila CONFIG1 atau “lang” bernilai “id” adalah “Biaya transfer = <biaya_transfer>” dan “Total biaya = <nominal_transfer + biaya_transfer>”.
- Setelah itu aplikasi akan menampilkan pesan yang bergantung dari bahasa yang dipilih di konfigurasi:
 1. EN =>” Select transfer method:”
 2. ID => “Pilih metode transfer:”
 - Kemudian juga akan dilakukan print dengan numbering untuk setiap string yang ada di JSON untuk key “methods” atau CONFIG5. Contoh output (dari nilai default):
 1. RTO (real-time)
 2. SKN
 3. RTGS
 4. BI FAST
 - Setelah menerima input pada pertanyaan sebelumnya, aplikasi akan bertanya satu pertanyaan terakhir dengan isi yang bergantung dari nilai “lang” dan “confirmation”
 1. EN => Please type "<CONFIG6>" to confirm the transaction:
 2. ID => Ketik "<CONFIG7>" untuk mengkonfirmasi transaksi:
 - Apabila input dari user sesuai dengan CONFIG6 atau CONFIG7 (tergantung nilai “lang”), maka:
 1. EN => The transfer is completed
 2. ID => Proses transfer berhasil
 - Tetapi jika input dari user tidak sesuai, maka:
 1. EN => Transfer is cancelled
 2. ID => Transfer dibatalkan

Jawab

- Source Code

- a. bank_transfer_config.json

Disini saya menggunakan Bahasa Inggris

```

{
  "lang": "en",
  "transfer": {
    "threshold": 25000000,
    "low_fee": 6500,
    "high_fee": 15000
  },
  "methods": ["RTO (real-time)", "SKN", "RTGS", "BI FAST"],
  "confirmation": {
    "en": "yes",
    "id": "ya"
  }
}

```

b. BankTransferConfig.cs

```

using System.Text.Json;
using System.IO;

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

    public static string filePath = "bank_transfer_config.json";

    public static BankTransferConfig LoadConfig()
    {
        if (!File.Exists(filePath))
        {
            var defaultConfig = new BankTransferConfig
            {
                lang = "en",
                transfer = new Transfer { threshold = 10000000, low_fee = 4000, high_fee = 10000 },
                methods = new List<string> { "BRIVA", "QRIS", "LinkAja" },
                confirmation = new Confirmation { en = "yes", id = "ya" }
            };

            string json = JsonSerializer.Serialize(defaultConfig, new JsonSerializerOptions { WriteIndented = true });
            File.WriteAllText(filePath, json);
            return defaultConfig;
        }
        else
        {
            string json = File.ReadAllText(filePath);
            return JsonSerializer.Deserialize<BankTransferConfig>(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; }
}

```

c. Program.cs

```

using System;

0 references
class Program
{
    0 references
    static void Main(string[] args)
    {
        BankTransferConfig config = BankTransferConfig.LoadConfig();

        Console.WriteLine("=== Nita Fitrotul Mar'ah - 2211104005 ===");
        Console.Write(config.lang == "en" ?
            "\nPlease insert the amount of money to transfer: " :
            "\nMasukkan jumlah uang yang akan di-transfer:");

        int amount = int.Parse(Console.ReadLine());

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

        // Fee Output
        if (config.lang == "en")
        {
            Console.WriteLine($"Transfer fee = {fee}");
            Console.WriteLine($"Total amount = {total}");
            Console.WriteLine("\nSelect transfer method:");
        }
        else
        {
            Console.WriteLine($"Biaya transfer = {fee}");
            Console.WriteLine($"Total biaya = {total}");
            Console.WriteLine("Pilih metode transfer:");
        }

        // Print methods
        for (int i = 0; i < config.methods.Count; i++)
        {
            Console.WriteLine($"{i + 1}. {config.methods[i]}");
        }

        Console.ReadLine(); // Method selection input

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

        string userConfirm = Console.ReadLine();

        if (userConfirm == confirmKeyword)
        {
            Console.WriteLine(config.lang == "en" ? "\nThe transfer is completed" : "\nProses transfer berhasil");
        }
        else
        {
            Console.WriteLine(config.lang == "en" ? "Transfer is cancelled" : "Transfer dibatalkan");
        }
    }
}

```

- Screenshot hasil run

```
Please insert the amount of money to transfer: 95000

Transfer fee = 6500
Total amount = 101500

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
```

- **Penjelasan**

Program ini menerapkan konsep runtime configuration, yaitu memungkinkan program membaca pengaturan dari file eksternal saat dijalankan. Dengan pendekatan ini, perubahan seperti bahasa tampilan, biaya transfer, atau metode transfer dapat dilakukan cukup dengan mengedit isi file `bank_transfer_config.json`, tanpa perlu memodifikasi kode program.

File konfigurasi tersebut berisi informasi penting seperti pilihan bahasa ("id" untuk Bahasa Indonesia atau "en" untuk Bahasa Inggris—dalam kasus ini digunakan Bahasa Inggris), nilai biaya transfer, daftar metode transfer (misalnya RTO, SKN, RTGS, BI FAST), serta kata kunci konfirmasi transaksi (seperti "yes" untuk menyetujui).

File `BankTransferConfig.cs` bertugas membaca dan menguraikan (melakukan parsing) isi file JSON tersebut, lalu menyimpannya dalam struktur data yang dapat digunakan oleh program. Selanjutnya, di `Program.cs`, program akan menyesuaikan alur kerja berdasarkan konfigurasi yang telah dimuat, seperti menampilkan teks sesuai bahasa yang dipilih, menghitung total biaya, dan meminta konfirmasi dari pengguna.

Seluruh perilaku program ditentukan oleh isi file JSON, sehingga tampilan dan fungsionalitas dapat disesuaikan dengan mudah hanya melalui pengeditan file konfigurasi, tanpa perlu menyentuh logika program secara langsung.