

**TUGAS JURNAL  
KONSTRUKSI PERANGKAT LUNAK**

**MODUL VIII  
RUNTIME CONFIGURATION &  
INTERNATIONALIZATION**



**Disusun Oleh :  
Atika Aji Hadiyani**

**2211104003**

**SE06-01**

**Dosen Pengampu :  
Yudha Islami Sulistya, S.Kom., M.Cs.**

**PROGRAM STUDI S1 SOFTWARE ENGINEERING FAKULTAS  
INFORMATIKA**

**TELKOM UNIVERSITY PURWOKERTO**

**2025**

## TUGAS JURNAL

### 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 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.
  3. Pesan output apabila CONFIG1 atau “lang” bernilai “en” adalah “Transfer fee = <biaya\_transfer>” dan “Total amount = <nominal\_transfer + biaya\_transfer>.”
  4. 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. 1. RTO (real-time)
  2. 2. SKN
  3. 3. RTGS
  4. 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;

5 references
public class BankTransferConfig
{
    7 references
    public string lang { get; set; }
    4 references
    public Transfer transfer { get; set; }
    3 references
    public List<string> methods { get; set; }
    3 references
    public Confirmation confirmation { get; set; }

    public static string filePath = "bank_transfer_config.json";

    1 reference
    public static BankTransferConfig LoadConfig()
    {
        if (!File.Exists(filePath))
        {
            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" }
            };

            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);
    }
}

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

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

### c. Program.cs

```
using System;

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

        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" ?
        $"Please type \"{confirmKeyword}\" to confirm the transaction: " :
        $"Ketik \"{confirmKeyword}\" untuk mengkonfirmasi transaksi:");

    string userConfirm = Console.ReadLine();

    if (userConfirm == confirmKeyword)
    {
        Console.WriteLine(config.lang == "en" ? "The transfer is completed" : "Proses 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
2

Please type "yes" to confirm the transaction: yes

The transfer is completed
```

- **Penjelasan:**

Program ini menggunakan konsep *runtime configuration*, yang memungkinkan aplikasi membaca pengaturan dari file eksternal saat dijalankan. Dengan pendekatan ini, kita tidak perlu mengubah kode setiap kali ingin mengganti pengaturan seperti bahasa, biaya transfer, atau metode transfer. Cukup ubah isi file `bank_transfer_config.json`, dan program akan menyesuaikan secara otomatis. File konfigurasi tersebut memuat informasi seperti opsi bahasa (contohnya "id" untuk Bahasa Indonesia atau "en" untuk Bahasa Inggris — dalam kasus ini digunakan "en"), nominal biaya transfer, daftar metode transfer (seperti RTO, SKN, RTGS, dan BI FAST), serta kata kunci untuk konfirmasi transaksi (misalnya "setuju").

Sementara itu, file `BankTransferConfig.cs` bertugas membaca dan menguraikan data dari file JSON, lalu menyimpannya dalam bentuk yang dapat dimanfaatkan oleh program utama. Di bagian `Program.cs`, pengaturan yang telah dibaca akan digunakan untuk menampilkan pertanyaan kepada pengguna, menghitung biaya transfer sesuai pengaturan, dan meminta konfirmasi pengguna. Seluruh proses dan teks yang ditampilkan akan mengikuti konfigurasi yang telah ditetapkan di file JSON. Dengan demikian, tampilan dan perilaku program dapat diubah dengan mudah hanya melalui pengeditan file konfigurasi, tanpa perlu menyentuh kode sumber utama.

- E. Ubah nilai default pada json file “bank\_transfer\_config.json” dengan nilai yang berbeda dan tunjukkan ke asisten praktikum hasil run dari aplikasi. Nilai dari input kedua di bagian C harus kurang dari nilai CONFIG2

**Jawab:**

- **Source code**

a. bank\_transfer\_config.json

Disini saya menggunakan Bahasa Indonesia

```
{
  "lang": "id",
  "transfer": {
    "threshold": 10000000,
    "low_fee": 4000,
    "high_fee": 10000
  },
  "methods": [ "BRIVA", "QRIS", "LinkAja" ],
  "confirmation": {
    "en": "yes",
    "id": "setuju"
  }
}
```

b. BankTransferConfig.cs

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

5 references
public class BankTransferConfig
{
    7 references
    public string lang { get; set; }
    4 references
    public Transfer transfer { get; set; }
    3 references
    public List<string> methods { get; set; }
    3 references
    public Confirmation confirmation { get; set; }

    public static string filePath = "bank_transfer_config.json";

    1 reference
    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);
    }
}

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

2 references
public class Confirmation
{
    2 references
    public string en { get; set; }
    2 references
    public string id { get; set; }
}

```

### c. Program.cs

```

using System;

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

        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.WriteLine(config.lang == "en" ?
        $"{confirmKeyword} to confirm the transaction: " :
        $"Ketik \"{confirmKeyword}\" untuk mengkonfirmasi transaksi:");

    string userConfirm = Console.ReadLine();

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

```

#### - Screenshot hasil run

```

Masukkan jumlah uang yang akan di-transfer:65000

Biaya transfer = 4000
Total biaya = 69000
Pilih metode transfer:
1. BRIVA
2. QRIS
3. LinkAja
2

Ketik "setuju" untuk mengkonfirmasi transaksi:setuju

Proses transfer berhasil

```

#### - Penjelasan

Program ini menggunakan konsep *runtime configuration*, yang memungkinkan aplikasi membaca pengaturan dari file eksternal saat dijalankan. Dengan pendekatan ini, kita tidak perlu mengubah kode setiap kali ingin mengganti pengaturan seperti bahasa, biaya transfer, atau metode transfer. Cukup ubah isi file `bank_transfer_config.json`, dan program akan menyesuaikan secara otomatis. File konfigurasi tersebut memuat informasi seperti opsi bahasa (contohnya "id" untuk Bahasa Indonesia atau "en" untuk Bahasa Inggris — dalam kasus ini digunakan "en"), nominal biaya transfer, daftar metode transfer (seperti RTO, SKN, RIDS, dan BI FAST), serta kata kunci untuk konfirmasi transaksi (misalnya "setuju").

Sementara itu, file `BankTransferConfig.cs` bertugas membaca dan menguraikan data dari file JSON, lalu menyimpannya dalam bentuk yang dapat dimanfaatkan oleh program utama. Di bagian `Program.cs`, pengaturan yang telah dibaca akan digunakan untuk menampilkan pertanyaan kepada pengguna, menghitung biaya transfer sesuai pengaturan, dan meminta konfirmasi pengguna. Seluruh proses dan teks yang ditampilkan akan mengikuti konfigurasi yang telah ditetapkan di file JSON. Dengan demikian, tampilan dan perilaku program dapat diubah dengan mudah hanya melalui pengeditan file konfigurasi, tanpa perlu menyentuh kode sumber utama.