

LAPORAN PRAKTIKUM JURNAL MODUL 8

NIM / Nama: 2311104072 – Jauhar Fajar Zuhair

Bagian I: Pendahuluan

Laporan ini menjelaskan implementasi program simulasi transfer bank yang konfigurasinya diatur melalui file JSON eksternal. Program ini memungkinkan pengaturan bahasa, biaya transfer, metode transfer, dan pesan konfirmasi yang dapat dimuat dan digunakan saat runtime.

Bagian II: File Konfigurasi `bank_transfer_config.json`

File ini berfungsi sebagai sumber data konfigurasi untuk aplikasi transfer bank.

Isi `bank_transfer_config.json`:

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

Penjelasan:

- `lang`: Bahasa default yang digunakan aplikasi ("en" untuk English, "id" untuk Indonesian).
- `transfer`: Objek yang berisi konfigurasi biaya transfer.
 - `threshold`: Batas jumlah transfer untuk menentukan biaya (`low_fee` atau `high_fee`).
 - `low_fee`: Biaya transfer jika jumlahnya di bawah atau sama dengan `threshold`.
 - `high_fee`: Biaya transfer jika jumlahnya di atas `threshold`.

- **methods:** Daftar metode transfer yang tersedia (sebagai array string).
 - **confirmation:** Teks konfirmasi yang valid untuk setiap bahasa.
 - **en:** Teks konfirmasi dalam Bahasa Inggris.
 - **id:** Teks konfirmasi dalam Bahasa Indonesia.
-

Bagian III: Desain Kelas `BankTransferConfig.cs`

Kelas ini memodelkan struktur data dari file JSON dan menyediakan fungsionalitas untuk memuat konfigurasi.

Struktur Kelas Utama dan Properti:

```
public class BankTransferConfig
{
    public string Lang { get; set; } = "en";
    public TransferSettings Transfer { get; set; } = new TransferSettings();

    public List<string> Methods { get; set; } = new List<string>
    {
        "RTO (real-time)", "SKN", "RTGS", "BI FAST"
    };

    public ConfirmationSettings Confirmation { get; set; } = new
    ConfirmationSettings();

    // ... Method LoadConfig() ...
}
```

Penjelasan:

- Kelas `BankTransferConfig` merepresentasikan keseluruhan file JSON.
- Properti `Lang`, `Transfer`, `Methods`, dan `Confirmation` sesuai dengan kunci level atas di JSON.
- Nilai default diberikan untuk memastikan objek tidak null bahkan jika file JSON tidak ada.

Kelas Nested untuk Struktur Data:

```
public class TransferSettings
{
    public long Threshold { get; set; } = 25000000;
    public int LowFee { get; set; } = 6500;
    public int HighFee { get; set; } = 15000;
}

public class ConfirmationSettings
{
    public string En { get; set; } = "yes";
    public string Id { get; set; } = "ya";
}
```

Penjelasan:

- Kelas `TransferSettings` dan `ConfirmationMessages` digunakan untuk memodelkan objek JSON bersarang (transfer dan confirmation).
 - Nama properti di C# sengaja dibuat sama persis (termasuk case) dengan kunci di JSON untuk mempermudah deserialisasi standar `System.Text.Json`.
-

Bagian IV: Memuat Konfigurasi (LoadConfig Method)

Method ini bertanggung jawab membaca file `bank_transfer_config.json` dan membuat instance `BankTransferConfig`.

Kode Method `LoadConfig`:

```
private const string ConfigFileName = "bank_transfer_config.json";

public static BankTransferConfig LoadConfig()
{
    if (File.Exists(ConfigFileName))
    {
        try
        {
            string jsonString = File.ReadAllText(ConfigFileName);
            var options = new JsonSerializerOptions
            {
                PropertyNameCaseInsensitive = true
            };
            return JsonSerializer.Deserialize<BankTransferConfig>(jsonString, options) ?? GetDefaultConfig();
        }
        catch (Exception ex)
        {
            Console.WriteLine($"Error loading config: {ex.Message}");
            return GetDefaultConfig();
        }
    }
    else
    {
        var defaultConfig = GetDefaultConfig();
        defaultConfig.SaveConfig();
        return defaultConfig;
    }
}
```

Penjelasan:

- Method statis `LoadConfig` membaca file JSON dari `FilePath`.
 - Menggunakan `JsonSerializer.Deserialize` untuk mengubah JSON string menjadi objek `BankTransferConfig`. Opsi `PropertyNameCaseInsensitive = true` ditambahkan untuk fleksibilitas (meskipun dalam kasus ini tidak wajib karena nama properti sudah cocok).
 - Jika terjadi error (misalnya file tidak ada), konfigurasi default dibuat dan disimpan kembali ke `bank_transfer_config.json` agar file selalu ada untuk eksekusi berikutnya.
-

Bagian V: Penggunaan Konfigurasi di Program.cs

Program utama memanfaatkan konfigurasi yang telah dimuat untuk menjalankan simulasi transfer bank.

Memuat Konfigurasi dan Memilih Bahasa:

```
var config = BankTransferConfig.LoadConfig();

Console.WriteLine("== Bank Transfer Application ==");
Console.WriteLine("Choose your language / Pilih bahasa:");
Console.WriteLine("1. English");
Console.WriteLine("2. Indonesian");
Console.Write("Enter your choice (1 or 2): ");

string langChoice = Console.ReadLine();

if (langChoice == "1")
{
    config.Lang = "en";
}
else if (langChoice == "2")
{
    config.Lang = "id";
}
else
{
    Console.WriteLine("Invalid choice. Using default language (English).");
    config.Lang = "en";
}
```

Penjelasan:

- Objek `config` dimuat menggunakan `BankTransferConfig.LoadConfig()`.
- Pengguna diminta memilih bahasa (English/Indonesian).
- Properti `config.Lang` diperbarui sesuai pilihan pengguna, yang akan mempengaruhi pesan-pesan selanjutnya.

Menggunakan Nilai Konfigurasi dalam Simulasi:

```
// Step 1: Menampilkan pesan sesuai bahasa yang dipilih
string promptAmount = (config.Lang == "id") ?
    "Masukkan jumlah uang yang akan di-transfer:" :
    "Please insert the amount of money to transfer:";
Console.WriteLine(promptAmount);
decimal amount = Convert.ToDecimal(Console.ReadLine());

// Step 2: Menghitung biaya transfer berdasarkan threshold
int transferFee;
if (amount <= config.Transfer.threshold)
{
    transferFee = config.Transfer.low_fee;
}
else
{
    transferFee = config.Transfer.high_fee;
}

string feeMsg = (config.Lang == "id") ?
    $"Biaya transfer = {transferFee}" :
    $"Transfer fee = {transferFee}";
Console.WriteLine(feeMsg);
Console.WriteLine($"Total amount = {amount + transferFee}");

// Step 3: Memilih metode transfer (menggunakan daftar dari config)
Console.WriteLine((config.Lang == "id") ? "Pilih metode transfer:" : "Select
transfer method:");
for (int i = 0; i < config.Methods.Count; i++)
{
    Console.WriteLine($"{i + 1}. {config.Methods[i]}");
}
// ... (logika input pilihan metode) ...

// Step 4: Konfirmasi transfer menggunakan teks dari config
string confirmPrompt = (config.Lang == "id") ?
    $"Ketik \"{config.Confirmation.id}\" untuk mengkonfirmasi transaksi:" :
    $"Type \"{config.Confirmation.en}\" to confirm the transaction:";
Console.Write(confirmPrompt + " ");
string confirmationInput = Console.ReadLine();

string expectedConfirmation = (config.Lang == "id") ? config.Confirmation.id
: config.Confirmation.en;

if (confirmationInput == expectedConfirmation)
{
    Console.WriteLine((config.Lang == "id") ? "Proses transfer berhasil" :
"The transfer is completed");
}
else
{
    Console.WriteLine((config.Lang == "id") ? "Transfer dibatalkan" :
"Transfer cancelled");
}
```

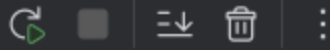
Penjelasan:

- Pesan yang ditampilkan ke pengguna (misalnya, `promptAmount`, `feeMsg`, `confirmPrompt`) dipilih berdasarkan nilai `config.Lang`.
 - Perhitungan biaya transfer (`transferFee`) menggunakan nilai `config.Transfer.threshold`, `config.Transfer.low_fee`, dan `config.Transfer.high_fee`.
 - Daftar metode transfer yang ditampilkan berasal dari `config.Methods`.
 - Teks konfirmasi yang diharapkan (`expectedConfirmation`) diambil dari `config.Confirmation.en` atau `config.Confirmation.id` sesuai bahasa yang dipilih.
-

Bagian VI: Hasil Eksekusi (Contoh Tampilan Konsol)

Berikut adalah contoh interaksi pengguna jika memilih Bahasa Indonesia:

Run jmmodul8_2311104072 (1) ×



=== Bank Transfer Application ===

Choose your language / Pilih bahasa:

1. English
2. Indonesian

Enter your choice (1 or 2): 2

Masukkan jumlah uang yang akan di-transfer:

5000000

Biaya transfer = 6,500

Total biaya = 5,006,500

Pilih metode transfer:

1. RTO (real-time)
2. SKN
3. RTGS
4. BI FAST

4

Metode yang dipilih: BI FAST

Ketik "ya" untuk mengkonfirmasi transaksi:

ya

Proses transfer berhasil
