LAPORAN PRAKTIKUM
KONSTRUKSI PERANGKAT BERGERAK

MODUL VIII

RUNTIME CONFIGURATION DAN INTERNATIONALIZATION

Disusun Oleh :

Dhiya Ulhaq R

S1SE-06-02

Asisten Praktikum :

Muhamad Taufiq Hidayat

Dosen Pengampu :

Riyan Dwi Yulian Prakoso, S.Kom., M.Kom.

PROGRAM STUDI S1 REKAYASA PERANGKAT LUNAK

DIREKTORAT TELKOM KAMPUS PURWOKERTO

2025

BAB I
PENDAHULUAN

A. DASAR TEORI

1. Runtime Configuration

Runtime Configuration adalah metodologi pengembangan software yang mengutamakan pemisahan antara kode program dan pengaturan aplikasi. Pendekatan ini memungkinkan perubahan perilaku aplikasi tanpa perlu memodifikasi kode sumbernya. Dalam konteks aplikasi transfer perbankan yang sedang dibahas, semua parameter konfigurasi disimpan dalam berkas eksternal bernama bank_transfer_config.json. Berkas konfigurasi ini menyimpan berbagai aspek penting aplikasi seperti:

- Bahasa aplikasi (lang(id)),
- Batas biaya transfer dan besaran biaya (threshold, low_fee, high_fee),
- Metode transfer yang tersedia (methods),
- Kata konfirmasi dalam berbagai bahasa (confirmation).

2. Internationalization

Internationalization (disingkat i18n) adalah strategi pengembangan yang memungkinkan aplikasi beroperasi secara efektif di berbagai konteks budaya dan linguistik. Implementasi i18n dalam aplikasi transfer bank ini mengadopsi pendekatan pragmatis dengan beberapa fitur utama:

- Pengorganisasian elemen teks dalam berbagai bahasa yang tersimpan rapi dalam file konfigurasi JSON.
- Pemanfaatan parameter lang sebagai penanda bahasa aktif yang menentukan pengalaman pengguna.
- Menyesuaikan teks yang ditampilkan berdasarkan bahasa yang dipilih (misalnya, prompt, label, konfirmasi transaksi).

B. TUJUAN

Setelah mengikuti praktikum ini, mahasiswa diharapkan mampu:

1. Membuat aplikasi berbasis console dengan Node.js yang terstruktur dan modular.
2. Menerapkan runtime configuration menggunakan file konfigurasi eksternal (format JSON) untuk mengatur parameter aplikasi.
3. Mengimplementasikan fitur internationalization (i18n) sederhana agar aplikasi dapat menyesuaikan bahasa tampilan berdasarkan konfigurasi.
4. Membaca dan memuat konfigurasi menggunakan modul File System (fs) dalam Node.js.

BAB II
(GUIDED DAN CHALLENGE)

Ubah bahasa default menjadi id lalu jalankan lagi programnya

Tambahkan opsi metode transfer baru di file JSON

Code

bank_transfer_config.json

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

BankTransferConfig.js

```
const fs = require('fs');
const path = require('path');

class BankTransferConfig {
  constructor() {
    this.configPath = path.join(__dirname,
'../data/bank_transfer_config.json');
    this.defaultConfig = {
      lang: "en",
      transfer: {
        threshold: 25000000,
        low_fee: 6500,
        high_fee: 15000
      },
      methods: ["RTO (real-time)", "SKN", "RTGS", "BI FAST"],
      confirmation: {
        en: "yes",
        id: "ya"
      }
    };
    this.config = this.loadConfig();
  }

  loadConfig() {
    if (fs.existsSync(this.configPath)) {
      const data = fs.readFileSync(this.configPath, 'utf8');
      return JSON.parse(data);
    } else {
      this.saveConfig(this.defaultConfig);
      return this.defaultConfig;
    }
  }

  saveConfig(config) {
    fs.writeFileSync(this.configPath, JSON.stringify(config, null, 2));
  }
}

module.exports = BankTransferConfig;
```

BankTransferApp.js

```
const readline = require('readline');
const BankTransferConfig = require('../config/BankTransferConfig');

class BankTransferApp {
  constructor() {
    this.config = new BankTransferConfig().config;
    this.rl = readline.createInterface({
      input: process.stdin,
      output: process.stdout
    });
  }

  askQuestion(query) {
    return new Promise(resolve => this.rl.question(query, resolve));
  }

  async run() {
    const lang = this.config.lang;

    const promptAmount = lang === "en" ?
      "Please insert the amount of money to transfer: " :
      "Masukkan jumlah uang yang akan di-transfer: ";

    const amountStr = await this.askQuestion(promptAmount);
    const amount = parseFloat(amountStr);

    const fee = amount <= this.config.transfer.threshold
      ? this.config.transfer.low_fee
      : this.config.transfer.high_fee;

    const total = amount + fee;

    if (lang === "en") {
      console.log(`Transfer fee = ${fee}`);
      console.log(`Total amount = ${total}`);
    } else {
      console.log(`Biaya transfer = ${fee}`);
      console.log(`Total biaya = ${total}`);
    }

    console.log(lang === "en" ? "Select transfer method:" : "Pilih metode
transfer:");
    this.config.methods.forEach((method, idx) => {
      console.log(`${idx + 1}. ${method}`);
    });

    await this.askQuestion(lang === "en" ? "Choose a method (press Enter
after): " : "Pilih metode (tekan Enter setelah): ");

    const confirmationPrompt = lang === "en" ?
      `Please type "${this.config.confirmation.en}" to confirm the
transaction: ` :
```

```
      `Ketik "${this.config.confirmation.id}" untuk mengkonfirmasi transaksi:
`;

    const confirmationInput = await this.askQuestion(confirmationPrompt);

    if (
      (lang === "en" && confirmationInput.trim().toLowerCase() ===
this.config.confirmation.en) ||
      (lang === "id" && confirmationInput.trim().toLowerCase() ===
this.config.confirmation.id)
    ) {
      console.log(lang === "en" ? "The transfer is completed" : "Proses
transfer berhasil");
    } else {
      console.log(lang === "en" ? "Transfer is cancelled" : "Transfer
dibatalkan");
    }

    this.rl.close();
  }
}

module.exports = BankTransferApp;
```

index.js

```
const BankTransferApp = require('./app/BankTransferApp');

const app = new BankTransferApp();
app.run();
```

Output 1

```
D:\bersama berkarya\SEMESTER 6\Praktikum KPL\KPL_Dhiya_Ulhaq_Ramadhan_2211104053_SE0602\08_Runtime Configuration dan Internationalization>node .
Masukkan jumlah uang yang akan di-transfer: 12000
Biaya transfer = 6500
Total biaya = 18500
Pilih metode transfer:
1. RTO (real-time)
2. SKN
3. RTGS
4. BI FAST
5. Mobile Banking
Pilih metode (tekan Enter setelah): 5
Ketik "ya" untuk mengkonfirmasi transaksi: ya
Proses transfer berhasil
```

Apa yang terjadi kalau input transfer amount bukan angka?

Output 2

```
D:\bersama berkarya\SEMESTER 6\Praktikum KPL\KPL_Dhiya_Ulhaq_Ramadhan_2211104053_SE0602\08_Runtime Configuration dan Internationalization>node .
Masukkan jumlah uang yang akan di-transfer: Dhiya
Biaya transfer = 15000
Total biaya = NaN
Pilih metode transfer:
1. RTO (real-time)
2. SKN
3. RTGS
4. BI FAST
5. Mobile Banking
Pilih metode (tekan Enter setelah): 4
Ketik "ya" untuk mengkonfirmasi transaksi: ya
Proses transfer berhasil
```

Deskripsi Code

Dalam proses transfer uang pada aplikasi ini, pengguna diminta memasukkan jumlah yang ingin ditransfer melalui interface command line. Sistem membaca input tersebut menggunakan modul readline, yang secara default menyimpan semua input sebagai tipe data string dalam variabel amountStr. Selanjutnya, aplikasi berusaha mengkonversi string tersebut menjadi nilai numerik menggunakan fungsi parseFloat(). Permasalahan timbul saat pengguna tidak memasukkan format angka yang sesuai. Beberapa skenario yang sering terjadi antara lain pengguna memasukkan nominal dengan kata-kata (misalnya "dua ratus ribu"). Pada semua kasus tersebut, fungsi parseFloat() gagal melakukan konversi dan mengembalikan nilai khusus bernama NaN (Not a Number).

Meskipun aplikasi tidak mengalami kegagalan teknis (crash), namun dari sisi pengalaman pengguna, ini merupakan kegagalan fungsional yang signifikan. Ketika informasi biaya transfer dan total biaya ditampilkan sebagai NaN, pengguna awam yang tidak familiar dengan konsep NaN dalam pemrograman akan kebingungan. Mereka mungkin menginterpretasikan NaN sebagai suatu kode error atau bahkan sebagai bagian dari proses normal, yang berpotensi menyebabkan kesalahpahaman dan keputusan yang tidak tepat dalam melanjutkan atau membatalkantransaksi.