

**LAPORAN PRAKTIKUM
KONSTRUKSI PERANGKAT LUNAK**

**MODUL VIII
Runtime Configuration dan Internationalization**



**Disusun Oleh :
Arzario Irsyad Al Fatih
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

1. DASAR TEORI

1. Runtime Configuration

Runtime Configuration adalah proses pengaturan perilaku aplikasi secara dinamis saat aplikasi dijalankan, tanpa perlu mengubah atau mengkompilasi ulang kode sumber. Konfigurasi ini biasanya disimpan dalam file eksternal seperti JSON, YAML, atau file lingkungan (.env), yang memungkinkan pengembang mengubah nilai parameter seperti ambang batas, biaya, atau mode bahasa dengan mudah. Dengan runtime configuration, aplikasi menjadi lebih fleksibel dan mudah disesuaikan untuk lingkungan atau kebutuhan pengguna yang berbeda, seperti perbedaan antara lingkungan pengembangan, pengujian, dan produksi.

2. Internationalization

Internationalization adalah proses perancangan dan pengembangan perangkat lunak agar dapat dengan mudah diadaptasi untuk berbagai bahasa dan wilayah tanpa perlu mengubah kode sumber aplikasi. Ini mencakup penggunaan format tanggal dan angka lokal, pesan yang dapat diterjemahkan, serta pemisahan teks antarmuka ke dalam file konfigurasi atau berkas terjemahan. Internationalization memungkinkan aplikasi mendukung banyak bahasa secara efisien dan merupakan tahap awal sebelum melakukan *localization* (110n), yaitu proses menerjemahkan dan menyesuaikan konten untuk audiens tertentu.

2. MAKSUD DAN TUJUAN

1. MAKSUD

Modul ini dibuat untuk memberikan pemahaman praktis kepada mahasiswa mengenai penerapan konsep Runtime Configuration dan Internationalization (i18n) dalam pengembangan aplikasi berbasis Node.js. Dengan pendekatan berbasis console, mahasiswa akan belajar bagaimana sebuah aplikasi dapat dibuat fleksibel dan mudah dikonfigurasi tanpa harus

memodifikasi logika program utama, serta mampu beradaptasi dengan kebutuhan pengguna dari latar belakang bahasa yang berbeda.

2. TUJUAN

Setelah mengikuti praktikum ini, mahasiswa diharapkan mampu:

- a. Membuat aplikasi berbasis console dengan Node.js yang terstruktur dan modular.
- b. Menerapkan runtime configuration menggunakan file konfigurasi eksternal (format JSON) untuk mengatur parameter aplikasi.
- c. Mengimplementasikan fitur internationalization (i18n) sederhana agar aplikasi dapat menyesuaikan bahasa tampilan berdasarkan konfigurasi.
- d. Membaca dan memuat konfigurasi menggunakan modul File System (fs) dalam Node.js.
- e. Menghubungkan konfigurasi dengan logika aplikasi, seperti perhitungan biaya transfer, pilihan metode, dan konfirmasi transaksi.
- f. Membiasakan diri dengan struktur project yang modular, memisahkan antara logika aplikasi, konfigurasi, dan data eksternal untuk kemudahan pengembangan lebih lanjut.

BAB II

IMPLEMENTASI (GUIDED)

1. BankTransferConfig.js

a. Code:

```
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;
```

2. BankTransferApp.js

a. Code

```
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;

```

3. bank_transfer_config.json

a. Code

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

4. indeks.js

a. Code:

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

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

b. Penjelasan

Program ini merupakan aplikasi berbasis Node.js yang mensimulasikan proses transfer bank dengan konfigurasi yang dapat diatur melalui file eksternal JSON. Program dimulai dari file index.js, yang menjalankan BankTransferApp. Aplikasi ini meminta pengguna untuk memasukkan jumlah uang yang ingin ditransfer, menghitung biaya berdasarkan ambang batas yang telah ditentukan (threshold), menampilkan total biaya, lalu meminta pengguna memilih metode transfer dan melakukan konfirmasi transaksi. Seluruh teks dan logika program mendukung dua bahasa (en dan id) sesuai pengaturan pada file bank_transfer_config.json, menjadikan aplikasi ini mendukung *internationalization*. Struktur modular program juga memungkinkan konfigurasi seperti bahasa, biaya, dan metode transfer dapat diubah tanpa perlu memodifikasi kode utama.

c. Output

```
PS C:\Users\toshiba\Documents\Tugas Kuliah\Semester 6\Praktikum\Kontruksi P
> node "c:\Users\toshiba\Documents\Tugas Kuliah\Semester 6\Praktikum\Kontru
ation\GUIDED\index.js"
Please insert the amount of money to transfer: 100000
Transfer fee = 6500
Total amount = 106500
Select transfer method:
1. RTO (real-time)
2. SKN
3. RTGS
4. BI FAST
Choose a method (press Enter after): 4
Please type "yes" to confirm the transaction: yes
The transfer is completed
```


BAB III

PENUGASAN (UNGUIDED)

1. Soal 1 Unguided

a. Code

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

b. Output

```
PS C:\Users\toshiba\Documents\Tugas Kuliah\Semester 6\Praktikum\Kontruksi Per
> node "c:\Users\toshiba\Documents\Tugas Kuliah\Semester 6\Praktikum\Kontruks
ation\GUIDED\index.js"
Masukkan jumlah uang yang akan di-transfer: 100000
Biaya transfer = 6500
Total biaya = 106500
Pilih metode transfer:
1. RTO (real-time)
2. SKN
3. RTGS
4. BI FAST
5. QRIS
Pilih metode (tekan Enter setelah): 5
Ketik "ya" untuk mengkonfirmasi transaksi: ya
Proses transfer berhasil
```

```
PS C:\Users\toshiba\Documents\Tugas Kuliah\Semester 6\Praktikum\Kontruksi Pe
> node "c:\Users\toshiba\Documents\Tugas Kuliah\Semester 6\Praktikum\Kontruk
ation\GUIDED\index.js"
Masukkan jumlah uang yang akan di-transfer: xyz
Input tidak valid. Masukkan angka.
```

c. Penjelasan

Kode di atas melibatkan perubahan bahasa default aplikasi ke Bahasa Indonesia, penambahan metode transfer baru (QRIS) ke dalam file konfigurasi JSON, serta penanganan input tidak valid dengan

menambahkan validasi agar aplikasi menolak input transfer yang bukan angka. Tugas ini menunjukkan penerapan runtime configuration dan internationalization secara sederhana.