

**LAPORAN PRAKTIKUM
KONSTRUKSI PERANGKAT BERGERAK**

MODUL VII

Aplikasi Bank Transfer dengan Runtime Configuration (Node.js)



Disusun Oleh :

Salman Alfa Rizzi

2211104056

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

Node.js adalah lingkungan runtime JavaScript yang dibangun di atas mesin V8 milik Google Chrome. Node.js bersifat open-source dan dirancang untuk membangun aplikasi jaringan yang scalable, terutama aplikasi berbasis server. Dengan pendekatan non-blocking dan event-driven, Node.js memungkinkan pengolahan data secara efisien dan cepat.

B. MAKSUD DAN TUJUAN

pembuatan aplikasi *Bank Transfer dengan Runtime Configuration* ini adalah untuk menerapkan konsep konfigurasi runtime dan internationalization (i18n) dalam pengembangan aplikasi Node.js. Aplikasi ini dirancang agar dapat membaca pengaturan dari file konfigurasi eksternal yang fleksibel serta menyesuaikan bahasa antarmuka berdasarkan pilihan pengguna.

Dengan adanya runtime configuration dan i18n, aplikasi menjadi lebih mudah diadaptasi tanpa perlu mengubah kode utama, baik untuk kebutuhan pengguna individu maupun lingkungan pengembangan yang berbeda.

BAB II

IMPLEMENTASI (GUIDED)

config/ 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;
```

data/ bank_transfer_config.json

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

app/ 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();
```

package.json

```
{
  "name": "modul-08---runtime-configuration-dan-internationalization",
  "version": "1.0.0",
  "main": "index.js",
  "scripts": {
    "test": "echo \\\"Error: no test specified\\\" && exit 1"
  },
  "keywords": [],
  "author": "",
  "license": "ISC",
  "description": ""
}
```

Output:

```
salmans-MacBook-Air:Modul 08 - Runtime Configuration dan Internationalization salman
alfarizzi$ node index.js
Masukkan jumlah uang yang akan di-transfer: 2000000
Biaya transfer = 6500
Total biaya = 2006500
Pilih metode transfer:
1. RT0 (real-time)
2. SKN
3. RTGS
4. BI FAST
Pilih metode (tekan Enter setelah): 4
Ketik "ya" untuk mengkonfirmasi transaksi: ya
Proses transfer berhasil
```

