

**LAPORAN PRAKTIKUM
KONSTRUKSI PERANGKAT LUNAK**

MODUL VIII

RUNTIME CONFIGURATION DAN INTERNATIONALIZATION



Disusun Oleh :

Dimas Cahyo Margono

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

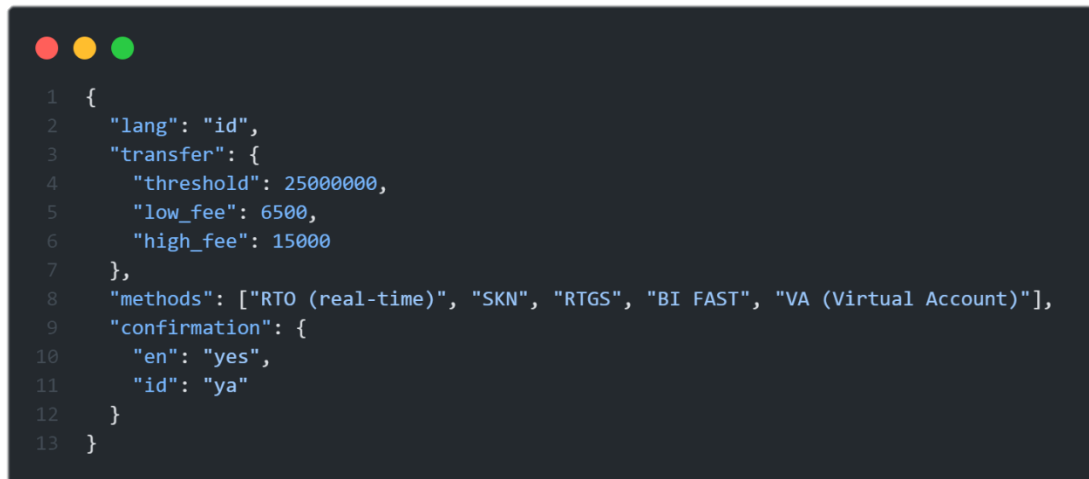
2025

BAB I

IMPLEMENTASI

Hasil Program:

1. bank_transfer_config.json



```
1  {
2    "lang": "id",
3    "transfer": {
4      "threshold": 25000000,
5      "low_fee": 6500,
6      "high_fee": 15000
7    },
8    "methods": ["RTO (real-time)", "SKN", "RTGS", "BI FAST", "VA (Virtual Account)"],
9    "confirmation": {
10     "en": "yes",
11     "id": "ya"
12   }
13 }
```

Penjelasan Kode:

File konfigurasi JSON ini digunakan untuk mengatur perilaku aplikasi transfer bank, dengan menetapkan bahasa antarmuka default ke Bahasa Indonesia ("lang": "id"), aturan biaya transfer berdasarkan ambang batas Rp25.000.000 (biaya rendah Rp6.500 dan tinggi Rp15.000), daftar metode transfer yang tersedia seperti RTO, SKN, RTGS, BI FAST, dan VA (Virtual Account), serta kata konfirmasi akhir yang harus diketik pengguna untuk menyelesaikan transaksi—"ya" untuk Bahasa Indonesia dan "yes" untuk Bahasa Inggris.

2. BankTransferConfig.js

```
1  const fs = require('fs');
2  const path = require('path');
3
4  class BankTransferConfig {
5    constructor() {
6      this.configPath = path.join(__dirname, '../data/bank_transfer_config.json');
7      this.defaultConfig = {
8        lang: "en",
9        transfer: {
10          threshold: 25000000,
11          low_fee: 6500,
12          high_fee: 15000
13        },
14        methods: ["RTO (real-time)", "SKN", "RTGS", "BI FAST"],
15        confirmation: {
16          en: "yes",
17          id: "ya"
18        }
19      };
20      this.config = this.loadConfig();
21    }
22
23    loadConfig() {
24      if (fs.existsSync(this.configPath)) {
25        const data = fs.readFileSync(this.configPath, 'utf8');
26        return JSON.parse(data);
27      } else {
28        this.saveConfig(this.defaultConfig);
29        return this.defaultConfig;
30      }
31    }
32
33    saveConfig(config) {
34      fs.writeFileSync(this.configPath, JSON.stringify(config, null, 2));
35    }
36  }
37
38  module.exports = BankTransferConfig;
```

Penjelasan Kode:

Kode JavaScript ini mendefinisikan kelas `BankTransferConfig` yang bertugas memuat dan menyimpan konfigurasi aplikasi transfer bank dari file JSON eksternal (`bank_transfer_config.json`). Di dalam konstruktor, ia menetapkan path ke file konfigurasi, mendefinisikan nilai default jika file belum ada, lalu memanggil `loadConfig()` untuk memuat konfigurasi yang tersedia. Jika file konfigurasi sudah ada, isinya dibaca dan diubah menjadi objek JavaScript menggunakan `JSON.parse`; jika belum ada, file dibuat menggunakan data default melalui metode `saveConfig()`. Kelas ini

memungkinkan pengelolaan konfigurasi aplikasi secara fleksibel dan terpisah dari logika program utama.

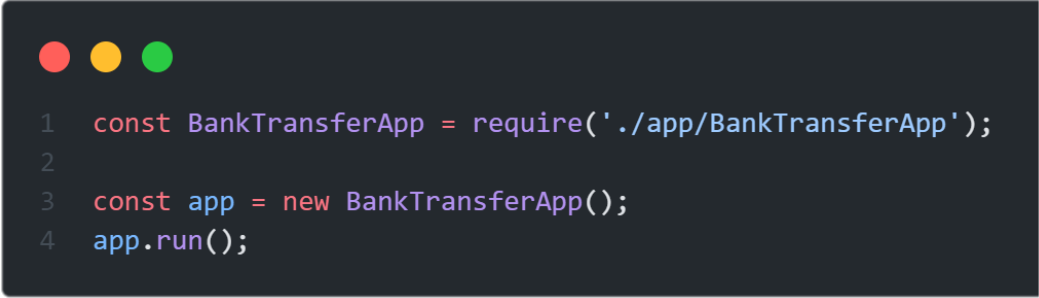
3. BankTransferApp.js

```
1  const readline = require('readline');
2  const BankTransferConfig = require('../config/BankTransferConfig');
3
4  class BankTransferApp {
5    constructor() {
6      this.config = new BankTransferConfig().config;
7      this.rl = readline.createInterface({
8        input: process.stdin,
9        output: process.stdout
10     });
11   }
12
13   askQuestion(query) {
14     return new Promise(resolve => this.rl.question(query, resolve));
15   }
16
17   async run() {
18     const lang = this.config.lang;
19
20     const promptAmount = lang === "en" ?
21       "Please insert the amount of money to transfer: " :
22       "Masukkan jumlah uang yang akan di-transfer: ";
23
24     let amount;
25     while (true) {
26       const amountStr = await this.askQuestion(promptAmount);
27       amount = parseFloat(amountStr);
28
29       if (!isNaN(amount) && amount > 0) {
30         break;
31       }
32
33       console.log(lang === "en" ? "Invalid amount. Please enter a valid number." : "Jumlah tidak valid. Masukkan angka yang benar.");
34     }
35
36     const fee = amount <= this.config.transfer.threshold
37       ? this.config.transfer.low_fee
38       : this.config.transfer.high_fee;
39
40     const total = amount + fee;
41
42     if (lang === "en") {
43       console.log(`Transfer fee = ${fee}`);
44       console.log(`Total amount = ${total}`);
45     } else {
46       console.log(`Biaya transfer = ${fee}`);
47       console.log(`Total biaya = ${total}`);
48     }
49
50     console.log(lang === "en" ? "Select transfer method:" : "Pilih metode transfer:");
51     this.config.methods.forEach((method, idx) => {
52       console.log(`${idx + 1}. ${method}`);
53     });
54
55     await this.askQuestion(lang === "en" ? "Choose a method (press Enter after): " : "Pilih metode (tekan Enter setelah): ");
56
57     const confirmationPrompt = lang === "en" ?
58       `Please type "${this.config.confirmation.en}" to confirm the transaction: ` :
59       `Ketik "${this.config.confirmation.id}" untuk mengkonfirmasi transaksi: `;
60
61     const confirmationInput = await this.askQuestion(confirmationPrompt);
62
63     if (
64       (lang === "en" && confirmationInput.trim().toLowerCase() === this.config.confirmation.en) ||
65       (lang === "id" && confirmationInput.trim().toLowerCase() === this.config.confirmation.id)
66     ) {
67       console.log(lang === "en" ? "The transfer is completed" : "Proses transfer berhasil");
68     } else {
69       console.log(lang === "en" ? "Transfer is cancelled" : "Transfer dibatalkan");
70     }
71
72     this.rl.close();
73   }
74 }
75
76 module.exports = BankTransferApp;
77
```

Penjelasan Kode:

Kode ini mendefinisikan kelas BankTransferApp yang merupakan inti dari aplikasi transfer bank berbasis terminal. Program menggunakan modul readline untuk membaca input pengguna dan memuat konfigurasi dari kelas BankTransferConfig. Saat dijalankan, aplikasi akan meminta pengguna untuk memasukkan jumlah uang yang ingin ditransfer, memverifikasi bahwa input adalah angka valid, lalu menghitung biaya transfer berdasarkan ambang batas yang ditentukan. Selanjutnya, pengguna diminta memilih metode transfer dari daftar yang tersedia dan memberikan konfirmasi akhir sebelum transaksi diproses. Bahasa antarmuka, biaya, dan kata konfirmasi semuanya disesuaikan berdasarkan pengaturan dalam file konfigurasi.

4. Index.js

A terminal window with a dark background and three colored window control buttons (red, yellow, green) in the top left corner. It contains four lines of JavaScript code with line numbers 1 through 4 on the left.

```
1  const BankTransferApp = require('./app/BankTransferApp');
2
3  const app = new BankTransferApp();
4  app.run();
```

Penjelasan Kode:

Kode ini merupakan titik masuk (entry point) program yang menjalankan aplikasi transfer bank. Baris pertama mengimpor kelas BankTransferApp dari file BankTransferApp.js yang berada di folder app. Selanjutnya, objek app dibuat dari kelas tersebut, dan metode run() dipanggil untuk memulai proses interaktif dengan pengguna melalui terminal—mulai dari input jumlah transfer, pemilihan metode, hingga konfirmasi transaksi.

Bonus Challenge:



Bonus Challenge untuk Mahasiswa

- Ubah bahasa default menjadi id lalu jalankan lagi programnya.
- Tambahkan opsi **metode transfer baru** di file JSON.
- Apa yang terjadi kalau input transfer amount bukan angka?

1. Mengubah bahasa menjadi id:

```
1 {
2   "lang": "id",
3   "transfer": {
4     "threshold": 25000000,
5     "low_fee": 6500,
6     "high_fee": 15000
7   },
8   "methods": ["RTO (real-time)", "SKN", "RTGS", "BI FAST", "VA (Virtual Account)"],
9   "confirmation": {
10    "en": "yes",
11    "id": "ya"
12  }
13 }
```

Untuk bahasa yang pada sebelumnya adalah en, itu diubah menjadi id pada bank_transfer_config.json

```

1  const readline = require('readline');
2  const BankTransferConfig = require('../config/BankTransferConfig');
3
4  class BankTransferApp {
5    constructor() {
6      this.config = new BankTransferConfig().config;
7      this.rl = readline.createInterface({
8        input: process.stdin,
9        output: process.stdout
10     });
11   }
12
13   askQuestion(query) {
14     return new Promise(resolve => this.rl.question(query, resolve));
15   }
16
17   async run() {
18     const lang = this.config.lang;
19
20     const promptAmount = lang === "en" ?
21       "Please insert the amount of money to transfer: " :
22       "Masukkan jumlah uang yang akan di-transfer: ";
23
24     let amount;
25     while (true) {
26       const amountStr = await this.askQuestion(promptAmount);
27       amount = parseFloat(amountStr);
28
29       if (!isNaN(amount) && amount > 0) {
30         break;
31       }
32
33       console.log(lang === "en" ? "Invalid amount. Please enter a valid number." : "Jumlah tidak valid. Masukkan angka yang benar.");
34     }
35
36     const fee = amount <= this.config.transfer.threshold
37       ? this.config.transfer.low_fee
38       : this.config.transfer.high_fee;
39
40     const total = amount + fee;
41
42     if (lang === "en") {
43       console.log(`Transfer fee = ${fee}`);
44       console.log(`Total amount = ${total}`);
45     } else {
46       console.log(`Biaya transfer = ${fee}`);
47       console.log(`Total biaya = ${total}`);
48     }
49
50     console.log(lang === "en" ? "Select transfer method:" : "Pilih metode transfer:");
51     this.config.methods.forEach((method, idx) => {
52       console.log(`${idx + 1}. ${method}`);
53     });
54
55     await this.askQuestion(lang === "en" ? "Choose a method (press Enter after): " : "Pilih metode (tekan Enter setelah): ");
56
57     const confirmationPrompt = lang === "en" ?
58       "Please type `${this.config.confirmation.en}` to confirm the transaction: " :
59       `Ketik "${this.config.confirmation.id}" untuk mengkonfirmasi transaksi: `;
60
61     const confirmationInput = await this.askQuestion(confirmationPrompt);
62
63     if (
64       (lang === "en" && confirmationInput.trim().toLowerCase() === this.config.confirmation.en) ||
65       (lang === "id" && confirmationInput.trim().toLowerCase() === this.config.confirmation.id)
66     ) {
67       console.log(lang === "en" ? "The transfer is completed" : "Proses transfer berhasil");
68     } else {
69       console.log(lang === "en" ? "Transfer is cancelled" : "Transfer dibatalkan");
70     }
71
72     this.rl.close();
73   }
74 }
75
76 module.exports = BankTransferApp;
77

```

Lalu menambahkan bahasa Indonesia pada file BankTransferApp.js dengan cara merevisi sedikit kodenya

2. Menambahkan opsi metode transfer baru:

```
1  {
2    "lang": "id",
3    "transfer": {
4      "threshold": 25000000,
5      "low_fee": 6500,
6      "high_fee": 15000
7    },
8    "methods": ["RTO (real-time)", "SKN", "RTGS", "BI FAST", "VA (Virtual Account)"],
9    "confirmation": {
10     "en": "yes",
11     "id": "ya"
12   }
13 }
```

Pada bagian methods, ditambahkan metode transfer baru yaitu VA (Virtual Account), yang dimana sebelumnya tidak ada.

3. Yang terjadi jika transfer amount bukan angka:

```
PS C:\Users\dimas\KPL_Dimas-Cahyo-Margono_2211104060_SE0602\Modul 08 - Runtime Configuration dan Internationalization> node index.js
Masukkan jumlah uang yang akan di-transfer: satu juta
Jumlah tidak valid. Masukkan angka yang benar.
Masukkan jumlah uang yang akan di-transfer: 
```

Jika mencoba untuk memasukkan amount dalam bentuk huruf (bukan angka), maka yang terjadi adalah “Jumlah tidak valid”, padahal angka yang dimasukkan adalah 1.000.000 (satu juta)

