

Link Repository Gtithub

https://github.com/IsabelleArd/KPL_ISABELLE-PUTRI-ARDINI_2311104030_SE0701/tree/master/08_RuntimeConfiguration_Internasionalization/Jurnal

BankTransferConfig.js

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

class BankTransferConfig {
  constructor() {
    this.configFile = 'bank_transfer_config.json';

    if (!fs.existsSync(this.configFile)) {
      this.lang = "en";
      this.transfer = {
        threshold: 25000000,
        low_fee: 6500,
        high_fee: 15000
      };
      this.methods = ["RTO (real-time)", "SKN", "RTGS", "BI FAST"];
      this.confirmation = {
        en: "yes",
        id: "ya"
      };
      this.saveConfig();
    } else {
      this.loadConfig();
    }
  }

  loadConfig() {
    const data = fs.readFileSync(this.configFile);
    const config = JSON.parse(data);
    this.lang = config.lang;
    this.transfer = config.transfer;
    this.methods = config.methods;
    this.confirmation = config.confirmation;
  }

  saveConfig() {
    const config = {
      lang: this.lang,
      transfer: this.transfer,
      methods: this.methods,
      confirmation: this.confirmation
    };
    fs.writeFileSync(this.configFile, JSON.stringify(config, null, 2));
  }
}
```

```
module.exports = BankTransferConfig;
```

Keterangan:

Code	Keterangan
<pre>constructor() { this.configFile = 'bank_transfer_config.json'; ... }</pre>	<p>Mengecek apakah file bank_transfer_config.json ada.</p> <ul style="list-style-type: none">- Jika tidak, membuat konfigurasi default dan menyimpannya ke file.- Jika ada, akan membaca dan memuat konfigurasinya.
<pre>loadConfig()</pre>	<p>Membaca dan membuat konfigurasi dari file.</p>
<pre>saveConfig()</pre>	<p>Menyimpan konfigurasi ke file jika terjadi inisialisasi awal.</p>

bank_transfer_config.json

```
{
  "lang": "id",
  "transfer": {
    "threshold": 20000000,
    "low_fee": 5000,
    "high_fee": 12000
  },
  "methods": ["SKN", "RTGS", "BI FAST", "Online Transfer"],
  "confirmation": {
    "en": "yes",
    "id": "ya"
  }
}
```

Keterangan:

Code	Keterangan
"lang": "id"	Bahasa antarmuka menggunakan Bahasa Indonesia.
"threshold"	Batas jumlah transfer: \leq Rp20.000.000 akan dikenakan low_free, selebihnya high_fee.
methods	Pilihan metode transfer
confirmation	Kata konfirmasi yang harus diketik pengguna

app.js

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

const rl = readline.createInterface({
  input: process.stdin,
  output: process.stdout
});

const config = new BankTransferConfig();

function askAmount() {
  if (config.lang === "en") {
    rl.question('Please insert the amount of money to transfer: ', handleAmount);
  } else {
    rl.question('Masukkan jumlah uang yang akan di-transfer: ', handleAmount);
  }
}

function handleAmount(amountInput) {
  const amount = parseFloat(amountInput);
  let transferFee = 0;

  if (amount <= config.transfer.threshold) {
    transferFee = config.transfer.low_fee;
  } else {
    transferFee = config.transfer.high_fee;
  }

  const total = amount + transferFee;

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

  chooseMethod();
}
```

```

function chooseMethod() {
  if (config.lang === "en") {
    console.log("Select transfer method:");
  } else {
    console.log("Pilih metode transfer:");
  }

  config.methods.forEach((method, index) => {
    console.log(`${index + 1}. ${method}`);
  });

  rl.question('Choose your transfer method (input number): ', confirmTransaction);
}

function confirmTransaction() {
  const confirmationText = (config.lang === "en") ? config.confirmation.en : config.confirmation.id;

  if (config.lang === "en") {
    rl.question(`Please type "${confirmationText}" to confirm the transaction: `, handleConfirmation);
  } else {
    rl.question(`Ketik "${confirmationText}" untuk mengkonfirmasi transaksi: `, handleConfirmation);
  }
}

function handleConfirmation(input) {
  const confirmationText = (config.lang === "en") ? config.confirmation.en : config.confirmation.id;

  if (input.trim().toLowerCase() === confirmationText.toLowerCase()) {
    if (config.lang === "en") {
      console.log("The transfer is completed");
    } else {
      console.log("Proses transfer berhasil");
    }
  } else {
    if (config.lang === "en") {
      console.log("Transfer is cancelled");
    } else {
      console.log("Transfer dibatalkan");
    }
  }
}

```

```

    rl.close();
  }

  askAmount();

```

Keterangan:

Code	Keterangan
const config = new BankTransferConfig();	Membuat instance dari konfigurasi, sehingga bisa dipakai di seluruh alur.

<code>rl.question(..., handleAmount);</code>	Menyesuaikan pertanyaan berdasarkan bahasa (ID atau EN).
<code>if (amount <= config.transfer.threshold) { transferFee = config.transfer.low_fee; } else { transferFee = config.transfer.high_fee; }</code>	Menentukan biaya transfer berdasarkan jumlah uang.
<code>console.log(`Biaya transfer = \${transferFee}`); console.log(`Total biaya = \${total}`);</code>	Menampilkan total biaya
<code>config.methods.forEach((method, index) => { console.log(`\${index + 1}. \${method}`); });</code>	Menampilkan daftar metode transfer yang tersedia dari file konfigurasi.
<code>rl.question(`Ketik "\${confirmationText}" untuk mengkonfirmasi...`, handleConfirmation);</code>	Menunggu konfirmasi dari user apakah ingin melanjutkan transaksi.
<code>if (input.trim().toLowerCase() === confirmationText.toLowerCase()) { console.log("Proses transfer berhasil"); } else { console.log("Transfer dibatalkan"); }</code>	<ul style="list-style-type: none"> - Jika input pengguna sesuai dengan kata konfirmasi, maka transaksi berhasil. - Jika tidak sesuai transaksi dibatalkan.