

Nama : Wulan Dwi Yanti

Nim : 211011401538

Kelas : TI22J

Tugas Pemrograman Berbasis Platform

LAPORAN SINGKAT MODULASI

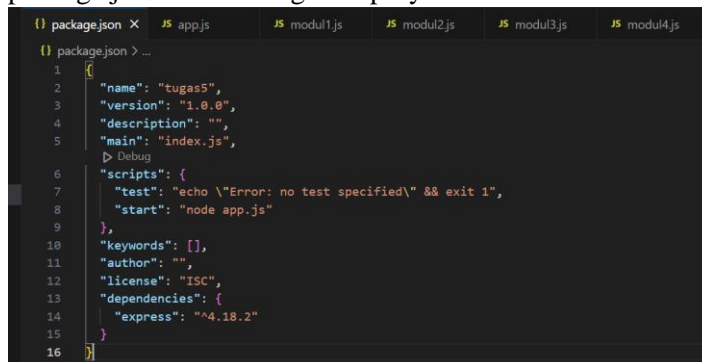
1. Konsep Modularisasi

Modularisasi adalah praktik penting dalam pengembangan perangkat lunak yang melibatkan pembagian kode menjadi modul atau file terpisah. Setiap modul berisi fungsionalitas yang berhubungan dengan tugas atau peran tertentu dalam aplikasi. Tujuan utama dari modularisasi adalah untuk meningkatkan keterbacaan, skalabilitas, dan pemeliharaan kode, serta memfasilitasi kerja tim.

2. Struktur Proyek

Dalam proyek Node.js yang mengimplementasikan modularisasi, struktur proyek akan mencakup beberapa file dan direktori:

- package.json: File konfigurasi proyek.



```
{
  "name": "tugas5",
  "version": "1.0.0",
  "description": "",
  "main": "index.js",
  "scripts": {
    "test": "echo \\\"Error: no test specified\\\" && exit 1",
    "start": "node app.js"
  },
  "keywords": [],
  "author": "",
  "license": "ISC",
  "dependencies": {
    "express": "^4.18.2"
  }
}
```

- File modul terpisah: Setiap modul berada dalam file terpisah yang dapat diimpor ke file utama.

- Modul1.js



```
function greet(name) {
  return `Hello, nama saya ${name}`;
}

module.exports = {
  greet,
};
```

- Modul2.js



```
function sapaWaktu(waktu) {
  let sapaan = 'Selamat';
  if (waktu === 'pagi') {
    sapaan += 'Pagi';
  } else if (waktu === 'sore') {
    sapaan += 'Sore';
  } else if (waktu === 'malam') {
    sapaan += 'Malam';
  }
  return `${sapaan}!`;
}

module.exports = {
  sapaWaktu,
};
```

- Modul3.js

```
JS modul3.js > cekGenapGanjil
1 function cekGenapGanjil(angka) {
2   return angka % 2 === 0 ? 'Genap' : 'Ganjil';
3 }
4 module.exports = {
5   cekGenapGanjil,
6 };
```

- Modul4.js

```
JS modul4.js > nilaiMutlak
1 function nilaiMutlak(angka) {
2   return Math.abs(angka);
3 }
4 module.exports = {
5   nilaiMutlak,
6 };
```

- File utama (app.js): File yang mengimpor dan menggunakan modul-modul tersebut.

```
JS app.js > ...
1 const modul1 = require('./modul1');
2 const modul2 = require('./modul2');
3 const modul3 = require('./modul3');
4 const modul4 = require('./modul4');
5
6 console.log(modul1.greet('Wulan'));
7 console.log(modul2.sapaWaktu('pagi'));
8 console.log(modul3.cekGenapGanjil(16));
9 console.log(modul4.nilaiMutlak(20));
10
```

3. Bagaimana Modul Bekerja Sama

Modul-modul bekerja sama dalam aplikasi Node.js dengan menggunakan dua konsep utama: `module.exports` dan `require`. Setiap modul mendefinisikan fungsi atau data yang ingin dibagikan kepada modul lain dengan menggunakan `module.exports`. Modul lain kemudian mengimpor modul tersebut menggunakan `require`.

4. Pentingnya Modularisasi

Dalam Pengembangan Perangkat Lunak Modularisasi sangat penting dalam pengembangan perangkat lunak karena:

- Keterbacaan: Memisahkan kode ke dalam modul-modul memungkinkan pengembang untuk fokus pada tugas tertentu, membuat kode lebih mudah dimengerti dan diikuti.
- Skalabilitas: Modul-modul memungkinkan aplikasi tumbuh secara modular dengan menambahkan atau menghapus modul sesuai kebutuhan tanpa mempengaruhi seluruh aplikasi.
- Pemeliharaan: Ketika ada perubahan atau masalah dalam aplikasi, perbaikan atau pembaruan hanya perlu dilakukan pada modul yang terkait, bukan seluruh aplikasi. Ini memudahkan pemeliharaan.
- Reusabilitas: Modul-modul yang terpisah dapat digunakan kembali dalam proyek-proyek lain, mempercepat pengembangan aplikasi baru.

Dengan demikian, modularisasi adalah prinsip penting dalam pengembangan perangkat lunak yang membantu menciptakan kode yang lebih terstruktur, mudah dikelola, dan lebih efisien dalam jangka panjang.