

**LAPORAN PRAKTIKUM
KONSTRUKSI PERANGKAT LUNAK**

**MODUL VI
Node.js**



**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. Node.js

Node.js adalah sebuah runtime environment yang dibangun di atas mesin JavaScript V8 milik Google Chrome. Node.js memungkinkan pengembang untuk menjalankan kode JavaScript di luar browser, yaitu pada sisi server. Dengan demikian, Node.js banyak digunakan dalam pengembangan aplikasi web berbasis server, API, serta berbagai alat bantu (tools) dalam pengembangan perangkat lunak modern.

Keunggulan Node.js antara lain:

- Non-blocking I/O: Node.js menggunakan arsitektur event-driven dan asynchronous, yang memungkinkan proses berjalan tanpa harus menunggu proses sebelumnya selesai.
- NPM (Node Package Manager): Sebuah ekosistem pustaka (library) terbesar yang memudahkan pengelolaan dependensi proyek.
- Cepat dan ringan: Karena dibangun di atas V8 dan bersifat non-blocking, Node.js cocok untuk aplikasi real-time seperti chat, game online, dan streaming.

2. JSON

JSON adalah format penyimpanan dan pertukaran data yang ringan dan mudah dibaca oleh manusia maupun mesin. JSON umumnya digunakan untuk menyimpan konfigurasi, data statis, serta dalam komunikasi antara klien dan server (terutama dalam REST API). Format ini terdiri dari pasangan *key-value*, mirip dengan objek dalam JavaScript.

2. MAKSUD DAN TUJUAN

1. MAKSUD

Modul ini bertujuan untuk memberikan pemahaman dan pengalaman praktis kepada pengembang (terutama pemula) mengenai cara:

- a. Membuat dan menggunakan modul di Node.js.
- b. Membaca dan memproses file JSON sebagai sumber data.
- c. Membangun aplikasi backend sederhana yang menampilkan informasi dari file JSON.
- d. Menggunakan Git sebagai version control system.
- e. Berkolaborasi dan berbagi kode melalui GitHub menggunakan fitur branch dan pull request.

2. TUJUAN

Setelah mempelajari dan mempraktikkan modul ini, peserta diharapkan mampu:

- a. Mengembangkan kemampuan dasar Node.js. Memberikan latihan kepada peserta dalam menggunakan modul core Node.js seperti fs dan path.
- b. Memahami proses membaca dan menampilkan data dari file JSON. Menunjukkan cara mengambil data dari file JSON dan menggunakannya dalam aplikasi JavaScript.
- c. Melatih praktik modularisasi kode. Mengajarkan cara menyusun kode agar lebih terstruktur dengan memisahkan logika ke dalam file modul terpisah.

BAB II

IMPLEMENTASI (GUIDED)

1. data_kuliah

a. Code:

```
{
  "student": {
    "name": "Ahmad Fadhil"
  },
  "lecturerCode": "D123",
  "courses": [
    {
      "courseCode": "IF101",
      "courseName": "Pemrograman Dasar"
    },
    {
      "courseCode": "IF102",
      "courseName": "Struktur Data"
    },
    {
      "courseCode": "IF103",
      "courseName": "Pemrograman Web"
    },
    {
      "courseCode": "IF104",
      "courseName": "Pemrograman Berorientasi Objek"
    }
  ]
}
```

2. package.json

a. Code

```
{
  "name": "nodejs-json-project",
  "version": "1.0.0",
  "main": "index.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "keywords": [],
  "author": "",
  "license": "ISC",
  "description": ""
}
```

3. KuliahData.js

a. Code

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

class KuliahData {
  constructor() {
    this.data = null;
  }

  readJSON() {
    const filePath = path.join(__dirname, 'data_kuliah.json');
    this.data = JSON.parse(fs.readFileSync(filePath, 'utf8'));
  }

  printInfo() {
    if (!this.data) return console.log("Data belum dibaca.");

    console.log(`Nama Mahasiswa: ${this.data.student.name}`);
    console.log(`Kode Dosen: ${this.data.lecturerCode}`);

    if (Array.isArray(this.data.courses)) {
      console.log("Daftar Mata Kuliah:");
      this.data.courses.forEach((course, index) => {
        console.log(`MK ${index + 1} ${course.courseCode} - ${course.courseName}`);
      });
    } else {
      console.log("Tidak ada data mata kuliah.");
    }
  }
}

module.exports = KuliahData;
```

b. Penjelasan

Kode tersebut mendefinisikan sebuah kelas bernama `KuliahData` yang digunakan untuk membaca dan menampilkan data kuliah dari file JSON. Dengan menggunakan modul `fs` dan `path` dari Node.js, metode `readJSON()` akan membaca file `data_kuliah.json` secara sinkron, lalu mengubah isinya menjadi objek JavaScript yang disimpan dalam properti `this.data`. Metode `printInfo()` akan menampilkan nama mahasiswa, kode dosen, serta daftar mata kuliah jika data telah berhasil dibaca. Jika data belum dibaca atau tidak tersedia, akan ditampilkan pesan peringatan yang

sesuai. Kelas ini diekspor menggunakan `module.exports` agar bisa digunakan di file JavaScript lainnya.

4. indeks.js

a. Code

```
const KuliahData = require('./KuliahData');

const kuliahData = new KuliahData();
kuliahData.readJSON();
kuliahData.printInfo();
```

b. Output

```
Nama Mahasiswa: Ahmad Fadhil
Kode Dosen: D123
Daftar Mata Kuliah:
MK 1 IF101 - Pemrograman Dasar
MK 2 IF102 - Struktur Data
MK 3 IF103 - Pemrograman Web
MK 4 IF104 - Pemrograman Berorientasi Objek
```

c. Penjelasan

Kode tersebut digunakan untuk mengimpor dan menggunakan kelas `KuliahData` yang sebelumnya telah didefinisikan dalam file `KuliahData.js`. Pada baris pertama, `require('./KuliahData')` memuat modul `KuliahData`, kemudian membuat objek baru dari kelas tersebut menggunakan `new KuliahData()`. Setelah objek dibuat, metode `readJSON()` dipanggil untuk membaca file `data_kuliah.json` dan memuat isinya ke dalam properti `data` di dalam objek. Terakhir, metode `printInfo()` dipanggil untuk menampilkan informasi mahasiswa, kode dosen, dan daftar mata kuliah yang telah dibaca dari file JSON ke konsol.