

**LAPORAN PRAKTIKUM  
KONSTRUKSI PERANGKAT BERGERAK**

**MODUL IX**

**Membuat API DI NODE.JS**



**Disusun Oleh :**

**Arwin Nabel Arroffif**

**2211104057**

**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. Latar Belakang**

Seiring berkembangnya teknologi web, komunikasi antara aplikasi menjadi sangat penting. Salah satu cara untuk menghubungkan aplikasi adalah melalui API (Application Programming Interface). API memungkinkan pertukaran data antara server dan client secara efisien. Node.js menjadi pilihan populer dalam pembuatan API karena bersifat ringan, cepat, dan mendukung pemrograman non-blok (non-blocking).

### **B. TUJUAN**

Tujuan dari pembuatan API ini adalah untuk menyediakan layanan pertukaran data antara aplikasi client dan server menggunakan Node.js dan Express.js secara efisien dan terstruktur.

## BAB II IMPLEMENTASI

### app.js

```
const express = require('express');
const app = express();
const port = 3000;

app.use(express.json());

// Simpan data mahasiswa di array static
let mahasiswa = [
  { nama: "Arwin Nabiel Arrofif", nim: "2211104057" },
  { nama: "sudibyo", nim: "2211104056" },
  { nama: "iis Tarwiyah", nim: "1302000993" }
];

// GET semua mahasiswa
app.get('/api/mahasiswa', (req, res) => {
  res.json(mahasiswa);
});

// GET mahasiswa berdasarkan index
app.get('/api/mahasiswa/:index', (req, res) => {
  const index = parseInt(req.params.index);
  if (index >= 0 && index < mahasiswa.length) {
    res.json(mahasiswa[index]);
  } else {
    res.status(404).json({ message: 'Data tidak ditemukan' });
  }
});

// POST mahasiswa baru
app.post('/api/mahasiswa', (req, res) => {
  const { nama, nim } = req.body;
  mahasiswa.push({ nama, nim });
  res.status(201).json({ message: 'Data berhasil ditambahkan' });
});

// DELETE mahasiswa berdasarkan index
app.delete('/api/mahasiswa/:index', (req, res) => {
  const index = parseInt(req.params.index);
  if (index >= 0 && index < mahasiswa.length) {
    mahasiswa.splice(index, 1);
    res.json({ message: 'Data berhasil dihapus' });
  } else {
    res.status(404).json({ message: 'Data tidak ditemukan' });
  }
});
```

```
app.listen(port, () => {  
  console.log(`Server berjalan di http://localhost:${port}`);  
});
```

### **.gitignore**

```
node_modules/  
.env
```

# DOKUMENTASI POSTMAN







