

**LAPORAN PRAKTIKUM
PERANCANGAN DAN PEMROGRAMAN WEB**

**MODUL 13
(NodeJS)**



Oleh:

Zulfa Mustafa Akhyar Iswahyudi - 2311104010

**PROGRAM STUDI S1 REKAYASA PERANGKAT LUNAK
DIREKTORAT KAMPUS PURWOKERTO
UNIVERSITAS TELKOM
2025**

BAB I

PENDAHULUAN

1.1 Dasar Teori

Node.js adalah runtime environment JavaScript yang dibangun di atas Chrome V8 JavaScript engine. Node.js memungkinkan developer untuk menggunakan JavaScript di sisi server, bukan hanya di sisi client. Dengan Node.js, developer dapat membuat aplikasi web yang scalable dan high-performance.

1.2 Tujuan

1. Memahami konsep dan arsitektur Node.js sebagai runtime environment JavaScript di server-side.
2. Mampu membuat RESTful API menggunakan Express.js untuk operasi CRUD.
3. Mampu mengintegrasikan Node.js dengan MySQL untuk pengelolaan data.
4. Mampu membuat aplikasi web full-stack dengan frontend HTML/JavaScript dan backend Node.js.
5. Memahami konsep middleware, routing, dan error handling dalam Express.js.
6. Mampu menggunakan environment variables untuk konfigurasi aplikasi.

1.3 Manfaat

1. **Skill Development** - Menambah kemampuan dalam mengembangkan aplikasi web modern dengan teknologi Node.js.
2. **Full-Stack Development** - Memahami bagaimana frontend dan backend bekerja bersama dalam satu aplikasi.
3. **Industry Relevance** - Node.js adalah teknologi yang banyak digunakan di industri untuk membangun aplikasi web scalable.
4. **Problem Solving** - Melatih kemampuan debugging dan problem solving dalam mengintegrasikan berbagai komponen.
5. **Best Practices** - Mempelajari best practices dalam struktur project, error handling, dan security.

BAB II

HASIL PRAKTIKUM

1. Mahasiswa diminta untuk membangun sebuah **aplikasi web sederhana** untuk pengelolaan data mahasiswa berbasis Node.js, Express.js, dan MySQL.

Aplikasi harus menyediakan **RESTful API** serta dapat diakses melalui browser menggunakan halaman web sederhana (HTML dan JavaScript).

Mahasiswa diperbolehkan menggunakan proyek yang telah dibuat di atas atau membuat proyek baru.

Catatan: Sertakan penjelasan tahapan pengerjaan aplikasi secara sistematis.

Output:

1.) Read

ID	NIM	Nama	Email	Jurusan	Aksi
1	2021001	Ahmad Rizki	ahmad@example.com	Teknik Informatika	<button>Edit</button> <button>Hapus</button>
2	2021002	Siti Nurhaliza	siti@example.com	Sistem Informasi	<button>Edit</button> <button>Hapus</button>
3	2021003	Budi Santoso	budi@example.com	Teknik Komputer	<button>Edit</button> <button>Hapus</button>

2.) Create

Pengelolaan Data Mahasiswa

Tambah/Edit Mahasiswa

NIM:
2311104010

Nama:
ZULFA MUSTAFA AKHYAR ISWAHYUDI

Email:
epiczakung17@gmail.com

Jurusan:
Software Architect

Daftar Mahasiswa

ID	NIM	Nama	Email	Jurusan	Aksi
1	2021001	Ahmad Rizki	ahmad@example.com	Teknik Informatika	Edit Hapus
2	2021002	Siti Nurhaliza	siti@example.com	Sistem Informasi	Edit Hapus
3	2021003	Budi Santoso	budi@example.com	Teknik Komputer	Edit Hapus

Data berhasil disimpan

Pengelolaan Data Mahasiswa

Tambah/Edit Mahasiswa

NIM:

Nama:

Email:

Jurusan:

Daftar Mahasiswa

ID	NIM	Nama	Email	Jurusan	Aksi
1	2021001	Ahmad Rizki	ahmad@example.com	Teknik Informatika	Edit Hapus
2	2021002	Siti Nurhaliza	siti@example.com	Sistem Informasi	Edit Hapus
3	2021003	Budi Santoso	budi@example.com	Teknik Komputer	Edit Hapus
5	2311104010	ZULFA MUSTAFA AKHYAR ISWAHYUDI	epiczakung17@gmail.com	Software Architect	Edit Hapus

3.) Update/Edit

Pengelolaan Data Mahasiswa

Tambah/Edit Mahasiswa

NIM:
2021003

Nama:
PALPALPAL

Email:
PAL@example.com

Jurusan:
REKAYASA AGAMA

Daftar Mahasiswa

ID	NIM	Nama	Email	Jurusan	Aksi
1	2021001	Ahmad Rizki	ahmad@example.com	Teknik Informatika	<button>Edit</button> <button>Hapus</button>
2	2021002	Siti Nurhaliza	siti@example.com	Sistem Informasi	<button>Edit</button> <button>Hapus</button>
3	2021003	Budi Santoso	budi@example.com	Teknik Komputer	<button>Edit</button> <button>Hapus</button>
5	2311104010	ZULFA MUSTAFA AKHYAR ISWAHYUDI	epiczakung17@gmail.com	Software Architect	<button>Edit</button> <button>Hapus</button>

Data berhasil disimpan

Tambah/Edit Mahasiswa

NIM:

Nama:

Email:

Jurusan:

Daftar Mahasiswa

ID	NIM	Nama	Email	Jurusan	Aksi
1	2021001	Ahmad Rizki	ahmad@example.com	Teknik Informatika	<button>Edit</button> <button>Hapus</button>
2	2021002	Siti Nurhaliza	siti@example.com	Sistem Informasi	<button>Edit</button> <button>Hapus</button>
3	2021003	PALPALPAL	PAL@example.com	REKAYASA AGAMA	<button>Edit</button> <button>Hapus</button>
5	2311104010	ZULFA MUSTAFA AKHYAR ISWAHYUDI	epiczakung17@gmail.com	Software Architect	<button>Edit</button> <button>Hapus</button>

4.) Delete

localhost:3000

localhost3000 says

Apakah Anda yakin ingin menghapus data ini?

[OK](#) [Cancel](#)

Tambah/Edit Mahasiswa

NIM:

Nama:

Email:

Jurusan:

[Simpan](#) [Reset](#)

Daftar Mahasiswa

ID	NIM	Nama	Email	Jurusan	Aksi
1	2021001	Ahmad Rizki	ahmadj@example.com	Teknik Informatika	Edit Hapus
2	2021002	Siti Nurhaliza	siti@example.com	Sistem Informasi	Edit Hapus
3	2021003	PALPALPAL	PAL@example.com	REKAYASA AGAMA	Edit Hapus
5	2311104010	ZULFA MUSTAFA AKHYAR ISWAHYUDI	epiczakung17@gmail.com	Software Architect	Edit Hapus

localhost:3000

Data berhasil dihapus

Tambah/Edit Mahasiswa

NIM:

Nama:

Email:

Jurusan:

[Simpan](#) [Reset](#)

Daftar Mahasiswa

ID	NIM	Nama	Email	Jurusan	Aksi
1	2021001	Ahmad Rizki	ahmadj@example.com	Teknik Informatika	Edit Hapus
3	2021003	PALPALPAL	PAL@example.com	REKAYASA AGAMA	Edit Hapus
5	2311104010	ZULFA MUSTAFA AKHYAR ISWAHYUDI	epiczakung17@gmail.com	Software Architect	Edit Hapus

BAB III

KESIMPULAN & SARAN

3.1 Kesimpulan

1. Node.js sebagai Server-Side Runtime - Node.js berhasil diimplementasikan sebagai runtime environment untuk menjalankan JavaScript di sisi server, memungkinkan pembuatan aplikasi web yang scalable dan efficient.
2. Express.js Framework - Express.js terbukti menjadi framework yang powerful dan mudah digunakan untuk membuat RESTful API dengan routing yang fleksibel dan middleware yang comprehensive.
3. Database Integration - Integrasi dengan MySQL menggunakan mysql2 library berhasil dilakukan dengan baik, termasuk connection pooling untuk mengelola multiple connections secara efisien.
4. Full-Stack Development - Aplikasi yang dikembangkan mendemonstrasikan konsep full-stack development dengan frontend HTML/JavaScript dan backend Node.js yang terintegrasi dengan baik.
5. CRUD Operations - Semua operasi CRUD (Create, Read, Update, Delete) berhasil diimplementasikan dan berfungsi dengan baik melalui RESTful API.
6. Best Practices - Implementasi mengikuti best practices seperti:
 - Separation of concerns (routes, config, middleware)
 - Environment variables untuk konfigurasi
 - Error handling yang proper
 - Async/await untuk asynchronous operations
 - Connection pooling untuk database efficiency

3.2 Saran

Nothing, yawwwn~ Untuk sintaks code liat aja di Github

