LAPORAN PRAKTIKUM TEKNOLOGI CLOUD COMPUTING

WEB SERVICE

PRAKTIKUM TEKNOLOGI CLOUD COMPUTING IF-A

Disusun oleh

Nama : Rahmadi Priambudi Riadi

NIM : 123200155

PROGRAM STUDI INFORMATIKA

JURUSAN INFORMATIKA

FAKULTAS TEKNIK INDUSTRI

UNIVERSITAS PEMBANGUNAN NASIONAL "VETERAN"

YOGYAKARTA

2025

HALAMAN PENGESAHAN
LAPORAN PRAKTIKUM

WEB SERVICE

PRAKTIKUM TEKNOLOGI CLOUD COMPUTING IF-A

Disusun Oleh :

Rahmadi Priambudi Riadi 123200155

Telah diperiksa dan disetujui oleh Asisten Praktikum : Muhammad Rafli, Aditya Prayoga

Pada tanggal : 3 Maret 2025

.

Menyetujui.

Asisten Praktikum	**Asisten Praktikum**
Muhammad Rafli	**Aditya Prayoga**
NIM 123210078	**NIM 123210098**

KATA PENGANTAR

Puji syukur penulis panjatkan ke hadirat Tuhan Yang Maha Esa atas berkat dan rahmat-Nya, sehingga laporan praktikum ini dapat diselesaikan dengan baik. Laporan ini disusun sebagai bagian dari tugas praktikum yang telah dilaksanakan.

Penulis mengucapkan terima kasih yang sebesar-besarnya kepada:

1. Bapak/Ibu Dosen Teori yang telah memberikan bimbingan dan ilmu pengetahuan selama perkuliahan.
2. Asisten Praktikum yang telah memberikan bantuan dan arahan selama pelaksanaan praktikum.
3. Semua pihak yang telah mendukung dan membantu dalam penyelesaian laporan ini.

Penulis menyadari bahwa laporan ini masih jauh dari sempurna. Oleh karena itu, penulis mengharapkan kritik dan saran yang membangun untuk perbaikan di masa yang akan datang.

Akhir kata, penulis berharap semoga laporan ini dapat bermanfaat bagi semua pihak yang membacanya.

Yogyakarta, 2 Maret 2025

Penulis

DAFTAR ISI

DAFTAR GAMBAR

BAB I
PENDAHULUAN

1.1 Latar Belakang

Dalam era digital yang semakin kompleks, kebutuhan untuk mengelola dan menyimpan catatan secara efisien menjadi semakin mendesak. Banyak individu dan organisasi yang menggunakan aplikasi catatan untuk mengorganisir informasi penting, namun sering kali mengalami kendala dalam mengakses catatan tersebut secara mudah dan cepat. Permasalahan yang umum terjadi antara lain adalah kurangnya integrasi antara frontend dan backend, sehingga pengguna kesulitan dalam melakukan operasi seperti menambah, mengubah, dan menghapus catatan. Selain itu, banyak aplikasi catatan yang tidak memiliki antarmuka pengguna yang intuitif, menyulitkan pengguna dalam menemukan informasi yang dibutuhkan. Oleh karena itu, diperlukan sebuah solusi berupa web service yang dapat membantu pengguna dalam mengelola catatan mereka dengan lebih baik.

Solusi yang diusulkan dalam penugasan ini adalah membangun sebuah web service yang mencakup fungsi backend dan frontend untuk manajemen catatan. Pada sisi backend, aplikasi ini harus memuat fitur CRUD (Create, Read, Update, Delete) yang terintegrasi melalui RESTful API. Hal ini akan memungkinkan komunikasi yang efisien antara frontend dan backend, memastikan data dapat diakses dan dipermodifikasi dengan cepat. Di sisi frontend, tampilan antarmuka harus dirancang dengan memperhatikan aspek usability, sehingga pengguna dapat dengan mudah mengakses dan mengelola catatan mereka tanpa kebingungan. Dengan pendekatan ini, pengguna dapat secara efektif mengelola informasi penting mereka dengan cara yang aman dan terstruktur.

Teknologi yang dipilih untuk membangun web service ini merupakan kombinasi stack yang teruji dan efisien dalam pengembangan aplikasi web. Penggunaan framework yang mendukung pengembangan frontend dan backend secara terpisah akan meningkatkan fleksibilitas dan kemampuan skalabilitas aplikasi. Selain itu, penerapan RESTful API dalam komunikasi data akan memastikan kecepatan dan kehandalan dalam pemrosesan permintaan pengguna.

Solusi ini tidak hanya menjawab permasalahan yang telah diidentifikasi, tetapi juga memberikan pengalaman pengguna yang lebih baik. Dengan memanfaatkan teknologi modern, diharapkan aplikasi ini dapat menjadi solusi yang tepat dan relevan dalam manajemen catatan di era digital saat ini.

1.2 Rumusan Masalah

1. Bagaimana cara merancang dan mengembangkan sebuah web service dengan tema notes (catatan) yang efektif dan efisien?
    a. Kriteria yang harus dipenuhi meliputi desain front-end yang menarik dan responsif serta backend yang berfungsi dengan baik.
2. Apa saja komponen yang diperlukan untuk memastikan bahwa tampilan front-end dan backend dapat berfungsi secara optimal?
    a. Pertimbangan mencakup penggunaan teknologi web yang tepat serta integrasi antara front-end dan backend.
3. Bagaimana implementasi fitur CRUD (Create, Read, Update, Delete) dan RESTful dalam backend?
    a. Penting untuk menentukan bagaimana fitur ini diimplementasikan dan diuji untuk memastikan semua fungsi bekerja dengan baik.

1.3 Tujuan

Untuk membuat sebuah web service dengan tema catatan, langkah pertama adalah mengembangkan bagian front-end dan back-end. Tampilan front-end harus dibuat dengan teknologi web yang sesuai dan harus bebas dalam desain. Selanjutnya, pastikan bahwa bagian back-end dapat menangani operasi CRUD (Create, Read, Update, Delete) dan mengikuti prinsip RESTful untuk pengembangan API. Selain itu, penting untuk memastikan bahwa front-end dan back-end dapat berkomunikasi dengan baik agar fungsionalitas aplikasi berjalan lancar.

1.4 Manfaat

Berikut adalah manfaat yang dapat diuraikan berkaitan dengan usulan pembuatan web service untuk catatan:

Manfaat Teknis:

- **Peningkatan Aksesibilitas:** Memberikan pengguna akses yang lebih mudah terhadap catatan penting mereka melalui antarmuka web yang responsif.
- **Kemudahan Pengelolaan Data:** Backend yang mendukung CRUD (Create, Read, Update, Delete) mempermudah pengguna dalam mengelola catatan mereka.
- **Integrasi dan Konektivitas:** Kemampuan aplikasi untuk menghubungkan front-end dan back-end secara efektif meningkatkan interoperabilitas sistem.
- **Penggunaan Teknologi Modern:** Memanfaatkan teknologi terkini dalam web development yang memungkinkan pengembangan yang lebih cepat dan efisien.

Manfaat Non-Teknis:

- **Meningkatkan Produktivitas:** Pengguna dapat menyimpan dan mengelola catatan dengan lebih efisien, mengurangi waktu yang dibutuhkan untuk mengatur informasi.
- **Peningkatan Pengalaman Pengguna:** Desain antarmuka yang ramah pengguna akan membuat interaksi lebih menyenangkan dan mengurangi kurva belajar.
- **Implementasi Kolaborasi:** Memungkinkan kolaborasi antara pengguna untuk berbagi dan mengedit catatan secara bersamaan.
- **Dukungan untuk Pengambilan Keputusan:** Data yang tersusun rapi dan mudah diakses akan mendukung pengguna dalam membuat keputusan yang lebih baik berdasarkan informasi terkini.

BAB II

TINJAUAN LITERATUR

2.1 Pengembangan Aplikasi Web dengan Node.js dan Express.js

Node.js adalah runtime environment berbasis JavaScript yang memungkinkan pengembangan aplikasi server-side dengan performa tinggi. Express.js adalah framework web untuk Node.js yang menyederhanakan pembuatan API dan routing. Dalam proyek ini, Node.js dan Express.js digunakan untuk membangun back-end aplikasi catatan yang menyediakan endpoint API untuk operasi CRUD (Create, Read, Update, Delete).

Express.js memungkinkan pengembang untuk membuat API RESTful dengan mudah, sementara Node.js menyediakan lingkungan yang efisien untuk menangani banyak permintaan secara bersamaan. Kombinasi ini sangat cocok untuk aplikasi catatan sederhana yang membutuhkan kecepatan dan skalabilitas.

2.2 MySQL sebagai Database Relasional

MySQL adalah sistem manajemen basis data relasional (RDBMS) yang banyak digunakan untuk menyimpan dan mengelola data terstruktur. Dalam proyek ini, MySQL digunakan untuk menyimpan data catatan, seperti judul dan isi catatan.

Keunggulan MySQL meliputi:

- Kemampuan untuk menangani data dalam jumlah besar.
- Dukungan untuk transaksi ACID (Atomicity, Consistency, Isolation, Durability).
- Integrasi yang mudah dengan aplikasi berbasis Node.js melalui library seperti mysql2.

BAB III
METODOLOGI

3.1 Analisis Permasalahan

Aplikasi catatan (Notes App) dirancang untuk memudahkan pengguna dalam membuat, membaca, mengupdate, dan menghapus catatan. Permasalahan yang dihadapi adalah:

1. **Manajemen Data**: Bagaimana menyimpan dan mengelola data catatan secara efisien.
2. **Antarmuka Pengguna**: Bagaimana menyediakan antarmuka yang intuitif untuk operasi CRUD.
3. **Integrasi Back-end dan Front-end**: Bagaimana memastikan komunikasi yang lancar antara back-end (Node.js) dan front-end (HTML, CSS, JavaScript).

Tujuan dari proyek ini adalah membangun aplikasi catatan yang memungkinkan pengguna untuk:

- Menambahkan catatan baru.
- Melihat daftar catatan yang sudah ada.
- Mengedit catatan yang sudah ada.
- Menghapus catatan.

3.2 Perancangan Solusi

Solusi yang diusulkan meliputi:

1. **Back-end**:
    - Menggunakan Node.js dan Express.js untuk membuat API RESTful.
    - Menggunakan MySQL untuk menyimpan data catatan.
    - Membuat endpoint API untuk operasi CRUD:
        - GET /api/notes: Mengambil semua catatan.
        - POST /api/notes: Menambahkan catatan baru.
        - PUT /api/notes/:id: Mengupdate catatan.
        - DELETE /api/notes/:id: Menghapus catatan.

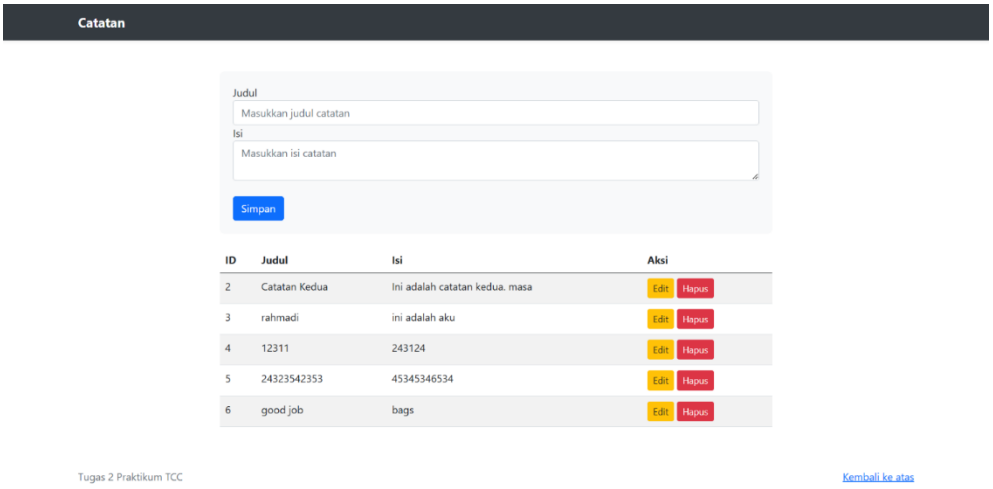
2. **Front-end**:
    - Menggunakan HTML, CSS, dan JavaScript untuk membuat antarmuka pengguna.
    - Menggunakan Bootstrap untuk styling dan layout.
    - Menggunakan JavaScript (dengan Fetch API) untuk berkomunikasi dengan back-end.
3. **Integrasi**:
    - Front-end mengirim request ke back-end menggunakan Fetch API.
    - Back-end memproses request dan mengembalikan respons dalam format JSON.

BAB IV
HASIL DAN PEMBAHASAN

4.1 Hasil

Berikut adalah hasil implementasi dari proyek aplikasi catatan:

Tampilan Aplikasi



Gambar 0.1 Tampilan web aplikasi catatan

Fungsi CRUD

- **Create**: Pengguna dapat menambahkan catatan baru melalui form.
- **Read**: Daftar catatan ditampilkan dalam tabel.
- **Update**: Pengguna dapat mengedit catatan yang sudah ada.
- **Delete**: Pengguna dapat menghapus catatan.

4.2 Pembahasan

1. Tujuan Tercapai:
    a. Aplikasi catatan berhasil dibangun dengan fitur CRUD yang berfungsi dengan baik.
    b. Back-end dan front-end terintegrasi dengan lancar.
2. Metode yang Digunakan:
    a. Node.js dan Express.js terbukti efektif untuk membangun API RESTful.
    b. MySQL cocok untuk menyimpan data catatan yang terstruktur.

3. Pemahaman Baru:
    a. Penggunaan Fetch API memudahkan komunikasi antara front-end dan back-end.
    b. Bootstrap menyederhanakan pembuatan antarmuka pengguna yang responsif.

BAB V
PENUTUP

5.1 Kesimpulan

Aplikasi catatan (Notes App) berhasil dibangun dengan fitur CRUD yang memungkinkan pengguna untuk menambah, membaca, mengedit, dan menghapus catatan. Back-end menggunakan Node.js, Express.js, dan MySQL, sementara front-end menggunakan HTML, CSS, JavaScript, dan Bootstrap. Aplikasi ini layak digunakan untuk manajemen catatan sederhana.

5.2 Saran

1. Pengembangan Fitur:
    a. Menambahkan fitur pencarian dan filter untuk catatan.
    b. Menambahkan autentikasi pengguna untuk keamanan.
2. Peningkatan Performa:
    a. Menggunakan caching untuk meningkatkan kecepatan akses data.
    b. Mengoptimalkan query database untuk menangani data dalam jumlah besar.
3. Penggunaan Teknologi Lain:
    a. Menggunakan React.js atau Vue.js untuk front-end yang lebih dinamis.
    b. Menggunakan MongoDB sebagai alternatif database NoSQL.

DAFTAR PUSTAKA

1. **Node.js Documentation**. (2023). *Node.js Official Documentation*. Diakses dari https://nodejs.org/en/docs/
2. **Express.js Documentation**. (2023). *Express.js Official Documentation*. Diakses dari https://expressjs.com/
3. **MySQL Documentation**. (2023). *MySQL Official Documentation*. Diakses dari https://dev.mysql.com/doc/
4. **Bootstrap Documentation**. (2023). *Bootstrap Official Documentation*. Diakses dari https://getbootstrap.com/docs/
5. **MDN Web Docs**. (2023). *Fetch API*. Diakses dari https://developer.mozilla.org/en-US/docs/Web/API/Fetch_API
6. **Haverbeke, M.** (2018). *Eloquent JavaScript: A Modern Introduction to Programming*. No Starch Press.
7. **Subramanian, V.** (2019). *Pro MERN Stack: Full Stack Web App Development with Mongo, Express, React, and Node*. Apress.
8. **W3Schools**. (2023). *JavaScript Tutorial*. Diakses dari https://www.w3schools.com/js/
    - .
9. **FreeCodeCamp**. (2023). *Learn Node.js and Express*. Diakses dari https://www.freecodecamp.org/
10. **JavaScript Info**. (2023). *JavaScript Fetch API*. Diakses dari https://javascript.info/fetch