

LAPORAN PRAKTIKUM TEKNOLOGI CLOUD COMPUTING

**CRUD NOTES WEB SERVICE DAN RESTFUL API
PRAKTIKUM TEKNOLOGI CLOUD COMPUTING IF-F**



Disusun oleh:

Nama : Annas Sovianto

NIM : 123220045

**PROGRAM STUDI INFORMATIKA
JURUSAN INFORMATIKA
FAKULTAS TEKNIK INDUSTRI
UNIVERSITAS PEMBANGUNAN NASIONAL "VETERAN" YOGYAKARTA
2025**

HALAMAN PENGESAHAN

LAPORAN PRAKTIKUM

CRUD NOTES WEB SERVICE DAN RESTFUL API PRAKTIKUM TEKNOLOGI CLOUD COMPUTING IF-F

Disusun Oleh:

Annas Sovianto

123220045

Telah diperiksa dan disetujui oleh Asisten Praktikum Teknologi Cloud Computing

Pada tanggal :

Menyetujui.

Asisten Praktikum

Asisten Praktikum

Berlyandhica Alam F

NIM. 123210060

Faustina Chelloana T

NIM. 123210139

KATA PENGANTAR

Segala puji dan syukur ke hadirat Tuhan Yang Maha Esa atas segala rahmat dan karunia-Nya sehingga penulis dapat menyelesaikan laporan praktikum Teknologi Cloud Computing dengan judul "CRUD Notes Web Service dan RESTful API". Laporan ini disusun sebagai bentuk dokumentasi dan evaluasi atas pelaksanaan praktikum serta sebagai salah satu syarat pemenuhan mata kuliah Teknologi Cloud Computing IF-F.

Penyusunan laporan ini tidak terlepas dari bimbingan, dukungan, serta kontribusi banyak pihak. Oleh karena itu, pada kesempatan ini penulis ingin menyampaikan rasa terima kasih kepada:

1. Bapak/Ibu Dosen Tim Pengampu Teknologi Cloud Computing yang telah memberikan arahan, motivasi, dan dukungan akademik selama pelaksanaan praktikum.
2. Berlyandhica Alam F dan Faustina Chelloana T, selaku Asisten Praktikum Teknologi Cloud Computing yang telah membimbing dan memberikan masukan selama kegiatan praktikum.
3. Rekan-rekan satu tim praktikum yang telah bekerja sama dalam menyelesaikan tugas dan tantangan selama praktikum.
4. Seluruh pihak yang terlibat dalam penyelenggaraan praktikum Teknologi Cloud Computing IF-F yang telah mendukung jalannya kegiatan ini hingga selesai.

Laporan ini diharapkan dapat memberikan gambaran yang jelas mengenai proses implementasi CRUD Notes Web Service dan RESTful API, termasuk aspek perancangan dan pengembangannya. Penulis menyadari bahwa laporan ini masih memiliki kekurangan. Oleh karena itu, penulis mengharapkan kritik dan saran untuk perbaikan di masa mendatang.

Semoga laporan ini dapat memberikan manfaat bagi pembaca dan menjadi referensi dalam pembelajaran teknologi cloud computing pada pengembangan layanan berbasis RESTful API.

Yogyakarta, 01 Maret 2025

Penulis

DAFTAR ISI

HALAMAN JUDUL	i
HALAMAN PENGESAHAN	ii
KATA PENGANTAR.....	iii
DAFTAR ISI	iv
DAFTAR GAMBAR.....	v
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	1
1.3 Tujuan.....	2
1.4 Manfaat.....	2
BAB II TINJAUAN LITERATUR	3
2.1 <i>Client (Front-end)</i>	3
2.2 <i>Server (Back-end)</i>	3
BAB III METODOLOGI	5
3.1 Analisis Permasalahan.....	5
3.2 Perancangan Solusi.....	5
BAB IV HASIL DAN PEMBAHASAN.....	7
4.1 Hasil.....	7
4.2 Pembahasan	10
BAB V PENUTUP	11
5.1 Kesimpulan.....	11
5.2 Saran.....	11
DAFTAR PUSTAKA.....	12

DAFTAR GAMBAR

Gambar 1. Tampilan Antarmuka Notes	7
Gambar 2. Formulir Input	8
Gambar 3. Tabel Notes	8
Gambar 4. Tabel Notes pada Database	9

BAB I

PENDAHULUAN

1.1 Latar Belakang

Di era digital yang semakin berkembang, pengelolaan data informasi menjadi salah satu tantangan utama bagi individu maupun organisasi. Banyak catatan penting yang tersebar dan tidak terorganisir dengan baik sehingga menyulitkan akses dan pemanfaatannya. Permasalahan tersebut menuntut adanya solusi pengelolaan data yang lebih efisien dan terstruktur agar informasi dapat diakses dengan mudah dan tepat waktu.

Solusi untuk mengatasi masalah tersebut adalah mengembangkan sebuah aplikasi catatan digital berbasis web. Aplikasi ini menggunakan kombinasi teknologi HTML, CSS, dan JavaScript pada sisi klien dan Express sebagai server dengan ORM Sequelize untuk mengelola database secara terstruktur. Aplikasi ini membuat mudah penggunaannya untuk melakukan penyimpanan dan pengelolaan data, serta meningkatkan interaksi melalui antarmuka yang intuitif.

Teknologi yang dipilih merupakan solusi tepat karena masing-masing memiliki keunggulan yang mendukung penyelesaian masalah. HTML, CSS, dan JavaScript menawarkan fleksibilitas dan interaktivitas pada tampilan, sedangkan Express menyediakan kerangka kerja yang efisien dalam mengatur routing. Sequelize membantu pengelolaan data secara aman dan terstruktur yang dapat menjadikan keseluruhan sistem sebagai solusi ideal untuk mengatasi permasalahan pengelolaan catatan yang tidak terstruktur.

1.2 Rumusan Masalah

Berdasarkan latar belakang di atas, masalah yang akan diselesaikan dalam pengembangan aplikasi ini dirumuskan sebagai berikut:

1. Bagaimana mengembangkan aplikasi catatan digital yang efektif untuk mengelola data secara terstruktur?
2. Bagaimana merancang antarmuka pengguna yang intuitif dan responsif agar memudahkan interaksi dalam pengelolaan catatan?
3. Bagaimana mengintegrasikan teknologi front-end dan back-end secara optimal untuk mendukung keamanan dan keandalan sistem pengelolaan data?

1.3 Tujuan

Tujuan dari pengembangan aplikasi catatan digital ini adalah:

1. Untuk mengetahui cara mengembangkan aplikasi catatan digital yang efektif dalam mengelola data secara terstruktur.
2. Untuk mengetahui cara merancang antarmuka pengguna yang intuitif dan responsif guna memudahkan interaksi dalam pengelolaan catatan.
3. Untuk mengetahui cara mengintegrasikan teknologi front-end dan back-end secara optimal demi mendukung keamanan dan keandalan sistem pengelolaan data.

1.4 Manfaat

Manfaat yang diharapkan dari pengembangan aplikasi ini meliputi:

1. Meningkatkan pemahaman dan penerapan teknologi pengembangan web dalam mengelola data secara terstruktur serta efektif.
2. Memberikan kemudahan dan efisiensi bagi pengguna melalui antarmuka yang intuitif dan responsif, sehingga proses pengelolaan catatan menjadi lebih cepat dan mudah.
3. Menjadi referensi dan solusi praktis bagi pengembang lain dalam mengintegrasikan teknologi front-end dan back-end untuk menciptakan sistem yang aman dan andal.

BAB II

TINJAUAN LITERATUR

2.1 *Client (Front-end)*

Sisi klien dalam pengembangan aplikasi berbasis web bertanggung jawab atas interaksi pengguna dan penyajian informasi secara dinamis. Teknologi utama yang digunakan dalam pengembangan antarmuka meliputi HTML (HyperText Markup Language), CSS (Cascading Style Sheets), dan JavaScript.

HTML berfungsi sebagai struktur dasar dalam penyusunan elemen-elemen pada halaman web. CSS digunakan untuk mengatur tata letak, warna, serta estetika tampilan guna meningkatkan pengalaman pengguna. JavaScript, sebagai bahasa pemrograman berbasis skrip, berperan dalam meningkatkan interaktivitas melalui manipulasi Document Object Model (DOM) serta pengelolaan event pengguna.

Hal yang digunakan untuk memastikan pengalaman pengguna yang optimal, responsivitas antarmuka menjadi faktor utama dalam perancangan sisi klien. Teknik seperti media query dalam CSS serta pendekatan event-driven programming dalam JavaScript digunakan untuk menyesuaikan tampilan dan fungsi sistem pada berbagai perangkat. Selain itu, pengelolaan data pada sisi klien dilakukan melalui mekanisme AJAX (Asynchronous JavaScript and XML) guna memungkinkan pertukaran data tanpa perlu melakukan pemuatan ulang halaman secara keseluruhan.

2.2 *Server (Back-end)*

Sisi server bertanggung jawab terhadap pemrosesan data, pengelolaan basis data, serta implementasi arsitektur REST (Representational State Transfer) dalam pengembangan API. Server dikembangkan menggunakan Node.js, sebuah lingkungan eksekusi berbasis JavaScript yang memungkinkan pemrosesan asinkron dan bersifat non-blocking, sehingga meningkatkan efisiensi dalam menangani permintaan dari klien.

Komunikasi antara klien dan server menggunakan protokol HTTP (Hypertext Transfer Protocol) dengan metode standar seperti GET, POST, PUT, PATCH, dan DELETE. Data yang dikirimkan dikemas dalam format JSON (JavaScript Object Notation) untuk memastikan interoperabilitas serta kemudahan dalam manipulasi data di sisi klien maupun server.

Basis data yang digunakan dalam sistem ini adalah MySQL, sebuah Relational Database Management System (RDBMS) yang menawarkan keandalan tinggi dalam

pengelolaan data terstruktur. Operasi dalam basis data mencakup fungsi CRUD (Create, Read, Update, Delete) yang dieksekusi melalui perintah SQL (Structured Query Language).

Pengujian API dilakukan dengan menggunakan Postman untuk memastikan setiap endpoint berfungsi sesuai spesifikasi serta mampu menangani berbagai skenario penggunaan, termasuk validasi input, pengelolaan otorisasi pengguna, serta efisiensi dalam pengelolaan permintaan berskala besar.

BAB III

METODOLOGI

3.1 Analisis Permasalahan

Perancangan sistem catatan digital ini bertujuan untuk menyediakan platform yang memungkinkan pengguna menyimpan, mengelola, serta mengakses catatan secara efisien. Permasalahan utama yang dihadapi adalah kebutuhan akan sistem yang mampu menyediakan antarmuka responsif di sisi klien, komunikasi yang andal dengan server, serta manajemen data yang aman dan terstruktur.

Dari sisi klien, tantangan utama meliputi perancangan antarmuka yang responsif dan interaktif agar dapat digunakan pada berbagai perangkat. Selain itu, interaksi dengan server harus dilakukan secara asinkron untuk meningkatkan efisiensi dalam pertukaran data.

Dari sisi server, permasalahan yang dihadapi mencakup pengelolaan data pengguna secara aman, penerapan standar komunikasi melalui arsitektur REST, serta integrasi dengan basis data relasional untuk memastikan efisiensi dalam penyimpanan dan pemrosesan data. Oleh karena itu, metode yang digunakan harus dapat menjamin kestabilan sistem, mengoptimalkan kecepatan akses data, serta menjaga keamanan informasi yang tersimpan.

3.2 Perancangan Solusi

Solusi yang diusulkan dalam pengembangan sistem catatan digital ini melibatkan perancangan dan implementasi yang mencakup dua komponen utama, yaitu sisi klien (front-end) dan sisi server (back-end).

Pada sisi klien, solusi dikembangkan dengan menerapkan teknologi dasar web, yakni HTML, CSS, dan JavaScript. Teknik media query digunakan dalam CSS untuk memastikan tampilan sistem tetap responsif pada berbagai ukuran layar. Interaksi pengguna dengan sistem ditingkatkan melalui penerapan event-driven programming dalam JavaScript, yang memungkinkan pengguna melakukan berbagai operasi seperti menambahkan, mengedit, dan menghapus catatan tanpa harus melakukan pemuatan ulang halaman. Komunikasi antara klien dan server dilakukan secara asinkron menggunakan AJAX, sehingga proses pengambilan dan pengiriman data dapat berlangsung lebih cepat dan efisien.

Di sisi server, solusi dirancang menggunakan Node.js untuk menangani permintaan dari klien dan mengelola data secara efisien. Struktur sistem mengikuti prinsip RESTful API, yang memungkinkan komunikasi yang standar dan terorganisir antara klien dan server. Basis data yang digunakan adalah MySQL, yang diakses menggunakan perintah SQL untuk

operasi CRUD (Create, Read, Update, Delete). Validasi input diterapkan pada server guna mencegah serangan seperti SQL Injection dan memastikan bahwa data yang diterima sesuai dengan format yang diharapkan.

Dalam implementasi API, setiap endpoint dikembangkan sesuai standar REST dengan metode HTTP seperti GET, POST, PUT, dan DELETE. Proses pengujian dilakukan menggunakan Postman untuk memastikan bahwa setiap endpoint berfungsi dengan benar dan mengembalikan respons yang sesuai. Selain itu, langkah-langkah pengamanan seperti validasi data, enkripsi kata sandi, dan pengelolaan sesi pengguna juga diterapkan untuk menjaga integritas dan keamanan sistem.

BAB IV

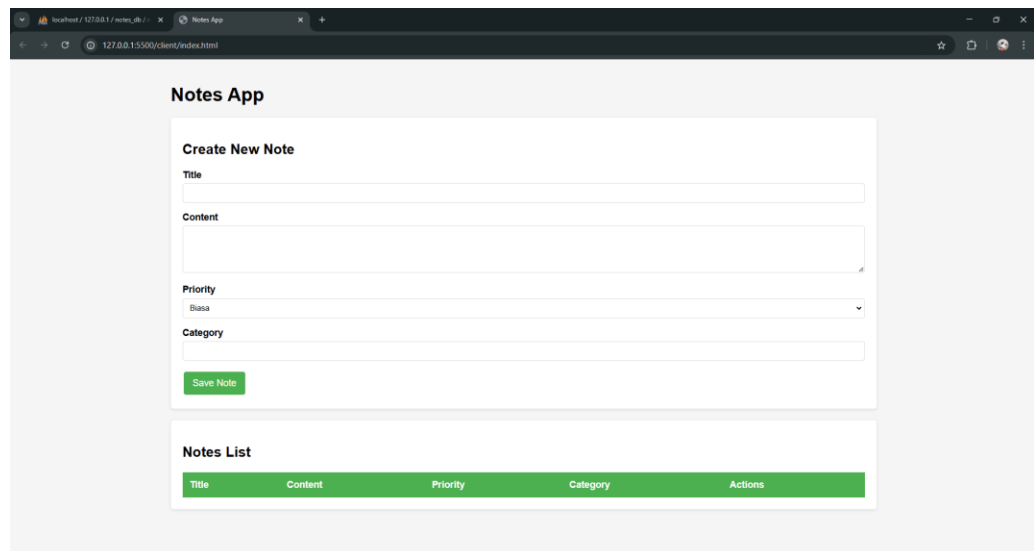
HASIL DAN PEMBAHASAN

4.1 Hasil

Implementasi sistem catatan digital yang telah dikembangkan menghasilkan antarmuka pengguna yang responsif serta komunikasi yang terstruktur antara sisi klien dan server. Hasil pengujian menunjukkan bahwa setiap fitur utama telah berhasil diimplementasikan sesuai dengan perencanaan.

Pada tahap implementasi sisi klien, tampilan antarmuka utama telah dirancang menggunakan HTML, CSS, dan JavaScript. Pengguna dapat menambahkan catatan baru, mengedit catatan yang sudah ada, serta menghapus catatan secara dinamis tanpa perlu melakukan pemuatan ulang halaman. AJAX digunakan untuk memastikan bahwa setiap perubahan yang dilakukan oleh pengguna langsung tersinkronisasi dengan server. Berikut ini merupakan dokumentasi-dokumentasi yang sudah dikembangkan:

a) Tampilan Antarmuka Pengguna (User Interface)



Gambar 1. Tampilan Antarmuka Notes

Gambar ini menunjukkan bahwa pengguna dapat melihat daftar catatan yang telah dibuat dengan informasi seperti judul, isi, kategori, dan prioritas catatan.

b) Formulir Input Notes

Create New Note

Title
Ringkasan Rapat

Content
Rapat yang diadakan pada 1 Maret 2025 membahas progres pengembangan aplikasi, kendala teknis, dan penjadwalan milestone berikutnya. Tim telah menyelesaikan 70% fitur utama, namun masih menghadapi masalah seperti latensi sinkronisasi data dan performa API yang kurang optimal. Solusi yang diusulkan meliputi penerapan caching di sisi klien, optimasi database, dan load testing API. Target berikutnya adalah debugging hingga 5 Maret, pengujian beta pada 10 Maret, serta finalisasi dokumentasi sebelum deployment. Rapat lanjutan dijadwalkan pada 8 Maret untuk mengevaluasi progres lebih lanjut.

Priority
Penting

Category
Kerja

Save Note

Gambar 2. Formulir Input

Formulir ini terdapat inputan pengguna untuk memasukkan data catatan, termasuk judul, isi, kategori, dan tingkat prioritas.

c) Tabel Notes

Title	Content	Priority	Category	Actions
Updated Meeting Notes	Revised project timeline and milestones	Penting	Work	<button>Edit</button> <button>Delete</button>
satu	satu satu	Rendah	sekolahsd	<button>Edit</button> <button>Delete</button>
asd	asd	Biasa	aasdaskdjaksjd	<button>Edit</button> <button>Delete</button>
Ringkasan Rapat	Rapat yang diadakan pada 1 Maret 2025 membahas pro...	Penting	Kerja	<button>Edit</button> <button>Delete</button>

Gambar 3. Tabel Notes

Tabel ini merupakan list-list catatan yang memuat Title, Content, Priority, Category, dan Actions untuk edit dan delete.

d) Pengujian API dengan REST (Testing)

```

1. GET http://localhost:5000/notes
2.
3. ###
4. GET http://localhost:5000/notes/1
5.
6. ###
7. POST http://localhost:5000/notes/
8. Content-Type: application/json
9.
10. {
11.     "title": "Golang",
12.     "content": "milestones",
13.     "priority": "Biasa",
14.     "category": "Work"

```

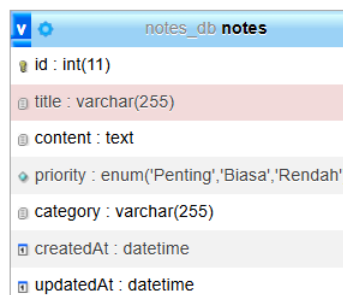
```

15. }
16.
17. ###
18. // Untuk priority hanya bisa diisi dengan "Penting",
    "Biasa", atau "Rendah"
19. PUT http://localhost:5000/notes/2
20. Content-Type: application/json
21.
22. {
23.     "title": "Updated Meeting Notes",
24.     "content": "Revised project timeline and
    milestones",
25.     "priority": "Penting",
26.     "category": "Work"
27. }
28.
29. ###
30. PATCH http://localhost:5000/notes/3
31. Content-Type: application/json
32.
33. {
34.     "title": "Partially Updated Notes",
35.     "priority": "Rendah"
36. }
37.
38. ###
39. DELETE http://localhost:5000/notes/1

```

Code ini sudah di test pada extension REST dan endpoint API yang telah dikembangkan mampu menangani berbagai permintaan dari klien dengan memberikan respons yang sesuai.

e) Struktur Database MySQL



notes_db	notes
id	int(11)
title	varchar(255)
content	text
priority	enum('Penting','Biasa','Rendah')
category	varchar(255)
createdAt	datetime
updatedAt	datetime

Gambar 4. Tabel Notes pada Database

Struktur database menunjukkan bahwa sistem menggunakan tabel yang dirancang untuk menyimpan informasi catatan secara terstruktur dengan skema yang sesuai.

4.2 Pembahasan

Pembahasan ini bertujuan untuk mengevaluasi sejauh mana tujuan tugas ini telah tercapai serta menganalisis kelebihan dan keterbatasan dari metode yang digunakan. Berikut ini merupakan kelebihan dan keterbatasannya:

a) Evaluasi kesesuaian dengan tujuan awal

Tujuan utama dari sistem ini adalah untuk menyediakan platform pencatatan digital yang efisien, aman, dan mudah diakses. Dari hasil implementasi yang diperoleh, sistem telah memenuhi kebutuhan tersebut dengan menyediakan antarmuka yang interaktif serta mendukung operasi CRUD secara optimal.

b) Evaluasi metode yang digunakan

Penggunaan AJAX dalam komunikasi antara klien dan server terbukti meningkatkan efisiensi sistem dengan memungkinkan pertukaran data secara asinkron. Selain itu, penerapan REST API memberikan struktur komunikasi yang standar sehingga memudahkan pengembangan lebih lanjut di masa depan.

c) Pemahaman baru dan hubungan dengan literatur

Hasil implementasi ini mengonfirmasi bahwa pendekatan berbasis REST API dan penggunaan MySQL sebagai database merupakan metode yang efektif dalam pengembangan aplikasi pencatatan digital. Namun, dalam literatur terkait, disebutkan bahwa untuk meningkatkan performa, dapat dipertimbangkan penggunaan teknologi caching atau database NoSQL jika sistem dihadapkan pada skala penggunaan yang lebih besar.

Berdasarkan hasil yang diperoleh, beberapa pengembangan yang dapat dilakukan di masa mendatang meliputi:

a) Optimasi kinerja melalui teknik caching atau pemilihan database yang lebih efisien jika sistem berkembang ke skala yang lebih besar.

b) Peningkatan keamanan, termasuk implementasi autentikasi pengguna dan enkripsi data sensitif.

c) Pengembangan fitur tambahan, seperti pencarian catatan, filter berdasarkan kategori, serta integrasi dengan layanan cloud untuk penyimpanan yang lebih fleksibel.

BAB V

PENUTUP

5.1 Kesimpulan

Berdasarkan hasil implementasi dan pembahasan yang telah dilakukan, dapat disimpulkan bahwa sistem pencatatan digital yang dikembangkan telah berhasil memenuhi tujuan utama, yaitu menyediakan platform pencatatan yang interaktif, responsif, serta mendukung operasi CRUD dengan baik.

Melalui penerapan AJAX dan REST API, komunikasi antara klien dan server dapat berjalan secara efisien tanpa perlu memuat ulang halaman, sehingga meningkatkan pengalaman pengguna. Selain itu, penggunaan MySQL sebagai basis data telah memungkinkan penyimpanan informasi yang terstruktur dan mudah dikelola.

Namun, meskipun sistem ini telah berfungsi sesuai dengan spesifikasi yang dirancang, masih terdapat beberapa aspek yang dapat ditingkatkan, terutama dalam hal keamanan dan skalabilitas. Oleh karena itu, diperlukan pengembangan lebih lanjut untuk memastikan sistem dapat digunakan secara optimal dalam skala yang lebih besar.

5.2 Saran

Berdasarkan hasil yang telah dicapai, terdapat beberapa aspek yang dapat ditingkatkan untuk memperbaiki serta mengembangkan sistem ini di masa mendatang, antara lain:

a) **Peningkatan Keamanan**

Diperlukan validasi tambahan di sisi server, serta mekanisme autentikasi dan otorisasi pengguna untuk mencegah akses tidak sah.

b) **Optimasi Performa dan Skalabilitas**

Jika sistem digunakan dengan jumlah data yang besar, performa bisa menjadi tantangan. Oleh karena itu, perlu dilakukan optimasi database melalui teknik indexing atau penggunaan database NoSQL jika diperlukan. Selain itu, penggunaan mekanisme caching dapat membantu meningkatkan efisiensi dalam menangani permintaan data.

DAFTAR PUSTAKA

- [1] W3Schools, "*HyperText Markup Language (HTML)*," W3Schools, 2025. [Online]. Available: <https://www.w3schools.com/html/default.asp>
- [2] W3Schools, "*Cascading Style Sheets (CSS)*," W3Schools, 2025. [Online]. Available: <https://www.w3schools.com/css/default.asp>
- [3] W3Schools, "*Javascript Learn*," W3Schools, 2025. [Online]. Available: <https://www.w3schools.com/js/default.asp>
- [4] Sequelize, "*Sequelize Documentation*," Sequelize, 2025. [Online]. Available: <https://sequelize.org/>
- [5] Jurusan Informatika, "*Modul Praktikum Teknologi Cloud Computing IF-F*," Universitas Pembangunan Nasional “Veteran” Yogyakarta, 2025.
- [6] Wibowo, A., Delima, R., & Mau, W. B. (2023). Implementasi JQuery AJAX Untuk Fitur Pendataan Petani pada Website Dutatani. *Jurnal Terapan Teknologi Informasi (JUTEI)*, 7(2), 99-105.
- [7] Nugroho, M. F., Primajaya, A., & Jajuli, M. (2023). Rancang Bangun REST API Aplikasi Manajemen Toko Menggunakan Nodejs pada Cantika Paint. *JATI (Jurnal Mahasiswa Teknik Informatika)*, 7(6), 3904-3910.
- [8] Qaisa, R. S., Putri, A., & Maghfirah, H. (2023). Perancangan Aplikasi ToDo Menggunakan Node. js dan REST API. *The Indonesian Journal of Computer Science*, 12(6).