

## **IMPLEMENTATION OF JSON WEB TOKEN IN THE DEVELOPMENT OF VILLAGE MONOGRAPH DATABASE BASED ON RESTAPI**

Akhmad Ramadhan<sup>\*1</sup>, Sri Mulyati<sup>2</sup>

<sup>1,2</sup>Informatics, Industrial Engineering Faculty, Universitas Islam Indonesia  
Email: <sup>1</sup>[20523049@students.uui.ac.id](mailto:20523049@students.uui.ac.id), <sup>2</sup>[mulya@uui.ac.id](mailto:mulya@uui.ac.id)

(Article received: October 17, 2024; Revision: November 6, 2024; published: December 29, 2024)

### **Abstract**

Currently, both central and local government institutions have been utilizing technological advancements to optimize services, strengthen partnerships, and maximize management efficiency. However, this has led to discrepancies in data across various government platforms due to the absence of an integrated database that can be linked with multiple institutional platforms. The development of an integrated database has become crucial to ensure data consistency and prevent discrepancies across different platforms. This study aims to develop an integrated database for village monographs by leveraging data transfer technology based on REST API and implementing security using JSON Web Token (JWT). The database development was carried out using the prototype development method, and testing was conducted using the black-box testing method. The results show that this research successfully developed a database capable of operating effectively on two different platforms, and the system developed adheres to the security standard of JSON Web Token. Therefore, this study can improve the efficiency of managing village monograph data while ensuring data protection during the transfer process between platforms.

**Keywords:** Back end, JSON Web Token, Monograph, REST API

## **IMPLEMENTASI JSON WEB TOKEN DALAM PENGEMBANGAN BASIS DATA MONOGRAFI DESA BERBASIS REST API**

### **Abstrak**

Saat ini lembaga pemerintahan baik dipusat maupun desa telah memanfaatkan perkembangan teknologi untuk mengoptimalkan pelayanan, memperkuat kemitraan, dan memaksimalkan efisiensi manajemen. Namun, hal ini menyebabkan munculnya perbedaan data di berbagai platform lembaga pemerintahan karena belum adanya basis data terpadu yang dapat diintegrasikan dengan berbagai platform lembaga. pengembangan basis data yang dapat terintegrasi menjadi sangat penting untuk memastikan konsistensi data dan mencegah terjadinya perbedaan informasi di berbagai platform lembaga. Penelitian ini bertujuan untuk mengembangkan basis data terintegrasi untuk monografi desa dengan memanfaatkan teknologi transfer data berbasis REST API serta menerapkan keamanan menggunakan JSON Web Token (JWT). Pengembangan basis data dilakukan dengan metode pengembangan prototype dan pengujian dilakukan menggunakan metode black box testing. Hasil menunjukkan bahwa penelitian ini mampu mengembangkan basis data yang mampu beroperasi secara efektif pada dua platform yang berbeda dan sistem yang dikembangkan telah menggunakan standar keamanan JSON Web Token. Sehingga, penelitian ini dapat meningkatkan efisiensi dalam pengelolaan data monografi desa yang menjamin perlindungan data dalam proses transfer antar platform.

**Kata kunci:** Back end, JSON Web Token, Monografi, REST API

### **1 PENDAHULUAN**

Perkembangan teknologi merupakan suatu hal yang tidak bisa dihindari, termasuk dalam lingkup pemerintahan desa [1]. Banyak lembaga pemerintahan telah memanfaatkan perkembangan teknologi untuk mengoptimalkan pelayanan, memperkuat kemitraan, dan memaksimalkan efisiensi manajemen [2]. Situasi ini justru

menyebabkan munculnya banyak perbedaan data di berbagai platform lembaga pemerintahan di Indonesia [3]. Menurut Ristiano dalam jurnal yang ditulis oleh Susniwati perbedaan ini disebabkan oleh perbedaan metode pengumpulan data yang diterapkan oleh masing-masing lembaga pemerintahan [3]. Perbedaan data tentu saja menimbulkan ambiguitas dalam penyebaran

informasi kepada masyarakat, yang pada akhirnya dapat berdampak negatif bagi perkembangan dan kesejahteraan desa. Pengumpulan data yang dilakukan oleh berbagai lembaga pemerintahan dengan menggunakan metode yang berbeda menunjukkan adanya pengambilan data yang berulang dan tidak efisien. Hal ini mengindikasikan bahwa proses pengumpulan data yang terfragmentasi ini dapat mengakibatkan duplikasi usaha serta menghambat efektivitas pengelolaan informasi.

Dalam konteks ini, pengembangan basis data yang dapat terintegrasi menjadi sangat penting untuk memastikan konsistensi data dan mencegah terjadinya perbedaan informasi di berbagai platform lembaga. Integrasi basis data memungkinkan pemerintah menyediakan layanan yang lebih responsif, efisien, dan berkualitas tinggi melalui pemantauan *real-time*, optimalisasi operasi, pengambilan keputusan berbasis data yang ilmiah, serta pengawasan cerdas yang meningkatkan akurasi, kecepatan, dan akuntabilitas pelayanan publik [4]. Selain itu, Basis data yang terintegrasi mampu mengoptimalkan distribusi data serta pemrosesan query dengan memanfaatkan keunggulan dari berbagai platform, termasuk perangkat *mobile*, *web*, *desktop*, *server*, *cloud*, hingga teknologi *post-cloud*, sehingga mendukung efisiensi, kecepatan, dan fleksibilitas dalam pengelolaan serta akses informasi secara menyeluruh [5]. Integrasi basis data dalam pemerintahan desa bertujuan mempercepat proses pengumpulan dan pengelolaan data ekonomi desa, serta memudahkan masyarakat mengakses informasi terkait ekonomi desa [6]. Sistem basis data desa yang terintegrasi juga dapat membantu dalam perencanaan dan pengendalian pembangunan desa, dari pengajuan anggaran hingga pelaporan [7].

Kementerian Komunikasi dan Informatika Indonesia menyatakan bahwa setidaknya terdapat tujuh kriteria informasi yang harus disertakan dalam platform publikasi pemerintah daerah yaitu, Ikhtisar, Pemerintahan Daerah, Geografi, Peta dan Sumber Daya Daerah, Peraturan dan Kebijakan Daerah, Berita dan Buku Tamu [8]. Kriteria informasi ini sejalan dengan data monografi desa yang diatur dalam Peraturan Menteri Dalam Negeri No.13 Tahun 2012 tentang monografi desa dan kelurahan. Monografi mencakup berbagai jenis data yaitu, data umum, data personal, data kewenangan, data keuangan, dan data kelembagaan [9]. Berdasarkan keselarasan antara kedua pedoman ini, pengembangan basis data monografi desa yang terintegrasi berpotensi menjadi solusi efektif dalam mengatasi permasalahan inkonsistensi data yang sering muncul di berbagai platform lembaga pemerintahan. Dengan adanya integrasi ini, data dari setiap desa akan lebih mudah dikelola dan diakses secara konsisten oleh berbagai pihak yang membutuhkan.

REST API merupakan salaf satu contoh *Application Programming Interface* yang mendukung sistem basis data yang dapat terintegrasi untuk monografi desa. REST API memungkinkan pengembang menggunakan basis data yang sama dalam pengembangan aplikasi dengan platform yang berbeda [10]. Hal tersebut terjadi karena REST API memanfaatkan aturan HTTP, yang menyederhanakan pengelolaan data serta memudahkan integrasi dengan berbagai platform dan kerangka kerja, sehingga meningkatkan efisiensi alur kerja pengelolaan data [11]. Penggunaan REST API memungkinkan adanya kolaborasi dalam memanfaatkan basis data terintegrasi dalam pengembangan publikasi monografi. Meskipun hanya dalam cakupan desa, jumlah data yang masuk dalam database dapat sangat besar. Dalam penelitian yang dilakukan oleh Vega-Grojo, mengindikasikan bahwa REST API memiliki kemampuan untuk mengelola dan mengakses data dalam jumlah besar [12]. Membangun basis data terintegrasi untuk monografi desa memerlukan tingkat keamanan yang tinggi. Data penduduk dalam basis data harus dilindungi dari ancaman pihak yang tidak bertanggung jawab. Pengguna REST API memungkinkan penerapan autentikasi dan otorisasi antara *server* dan pengguna dengan menggunakan JSON Web Token, sehingga keamanan data dapat terjaga secara efektif dan aman [13]. Dengan menggunakan JWT, aplikasi dapat memverifikasi identitas pengguna melalui pengiriman token dari klien ke server sebagai bukti autentikasi dan otorisasi, server kemudian memverifikasi token tersebut dan memberikan akses ke sumber daya yang diminta jika token valid [14]. Otentikasi berbasis token memiliki keunggulan dibandingkan otentikasi tradisional karena bersifat *stateless*, sehingga dapat menggantikan peran *cookie* dan *session*. Hal ini mengurangi beban kerja server dengan menghilangkan kebutuhan untuk mempertahankan *session* pada layanan web tradisional [15].

Beberapa peneliti sebelumnya telah mengembangkan basis data menggunakan REST API. Salah satunya adalah penelitian yang dilakukan oleh Astowo, berjudul *Penerapan JSON Web Token sebagai Strategi Pengamanan Data pada Aplikasi MultiMasjid* [14]. Dalam penelitiannya, Astowo mengembangkan basis data untuk aplikasi multi masjid dengan menggunakan teknologi REST API sebagai penghubung komunikasi antara aplikasi dan basis data. Untuk aspek keamanan, Astowo menerapkan JSON Web Token (JWT) dalam setiap pertukaran data guna memastikan keamanan data dalam aplikasi tersebut. Hasil penelitian ini menunjukkan bahwa pemanfaatan REST API sebagai alat pertukaran data, ditambah penerapan JWT sebagai mekanisme keamanan, memberikan kinerja optimal dalam hal respons data dari server. Selain itu, pendekatan ini juga menjamin keamanan

data dan melindungi informasi pengguna dari akses yang tidak sah.

Penelitian selanjutnya dilakukan oleh Febriansyah dengan judul *Perancangan Sistem Informasi E-Arsip Berbasis Website Pada Desa Wateswinangun* [16]. Dalam penelitian ini, Febriansyah mengembangkan sistem informasi berbasis web untuk pengarsipan elektronik dalam lingkup pemerintahan desa. Pada sistem tersebut, Febriansyah menggunakan REST API sebagai sarana komunikasi antara server basis data dan server situs web. Namun, sistem ini tidak menggunakan layanan keamanan JSON Web Token untuk melindungi data desa. Hasil dari sistem ini menunjukkan bahwa penerapan REST API pada basis data situs web dapat meningkatkan efektivitas dalam proses pertukaran data.

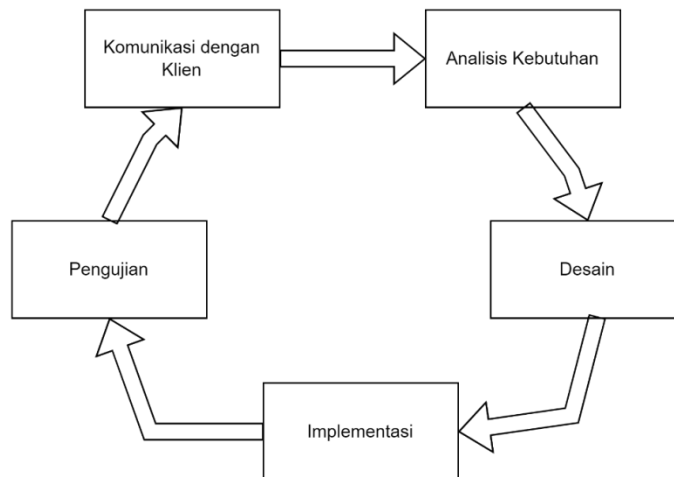
Berdasarkan pemaparan penelitian terdahulu yang memiliki kesamaan unsur variabel, peneliti kemudian mengangkat tema dengan unsur kebaruan, yaitu pengembangan basis data terintegrasi untuk monografi desa dengan menggunakan metode REST API serta metode autentikasi JSON Web Token (JWT). Penelitian ini bertujuan untuk membangun basis data monografi yang dapat diintegrasikan dengan berbagai platform, menggunakan REST API sebagai penghubung, serta menerapkan JSON Web Token untuk autentikasi dan otorisasi. Dengan demikian, penyebaran informasi dapat dilakukan melalui berbagai platform, sehingga meningkatkan efektivitas penyebaran informasi monografi desa. Melalui penggunaan REST API, penelitian ini

berfokus pada optimalisasi akses dan distribusi data antar platform yang berbeda, memungkinkan interoperabilitas yang lebih baik dan efisiensi dalam pengelolaan serta penyebaran informasi. Hal ini diharapkan dapat memberikan kontribusi yang signifikan terhadap upaya peningkatan transparansi, akurasi, dan kecepatan dalam penyampaian informasi terkait monografi desa kepada masyarakat luas dan pihak-pihak yang berkepentingan.

## 2 METODE PENELITIAN

### 2.1 Metode Pengembangan

Metode prototipe dipilih karena memungkinkan pengembang memahami kebutuhan pengguna secara mendalam melalui pendekatan iteratif yang responsif terhadap umpan balik. Dalam pengembangan basis data monografi desa berbasis REST API, prototipe menjadi relevan karena sistem ini harus memenuhi kebutuhan spesifik pengguna terkait akses data yang fleksibel, dinamis, dan mudah digunakan. Dengan prototipe, pengembang dapat membuat model awal sistem yang diuji oleh pengguna, sehingga setiap revisi mencerminkan kebutuhan nyata dan spesifikasi teknis secara akurat. Pendekatan ini meminimalkan risiko kesalahan desain, memperkuat komunikasi antara pengembang dan pengguna, serta menghasilkan sistem yang efektif dan sesuai dengan ekspektasi operasional di lapangan.



Gambar 1 Metode Pengembangan

Tahap-tahap yang digunakan dalam pengembangan yang digunakan dalam metode prototype ditunjukkan pada Gambar 1. Peneliti menggunakan diskusi sebagai pendekatan utama untuk berkomunikasi dengan klien, membahas kebutuhan, masalah, dan harapan dalam pengembangan basis data monografi desa terpadu berbasis REST API, guna memastikan sistem memenuhi kebutuhan operasional dan dapat

diimplementasikan secara efektif. Dalam proses pengembangan, peneliti memilih metode prototipe karena metode ini memungkinkan klien untuk melihat representasi awal dari sistem yang dikembangkan dan memberikan masukan secara iteratif, sehingga kebutuhan yang mungkin tidak teridentifikasi sejak awal dapat diakomodasi dengan lebih baik. Pemilihan metode ini relevan dengan kebutuhan sistem yang dinamis dan spesifik, seperti

basis data monografi desa, di mana sistem harus dapat menyesuaikan diri dengan kebutuhan pengguna yang beragam dan memastikan fungsi-fungsi utama berjalan sesuai ekspektasi klien. Berdasarkan hasil komunikasi ini, peneliti menganalisis kebutuhan pengguna dalam dua tahap: analisis kebutuhan pengguna untuk memahami kebutuhan spesifik, dan analisis kebutuhan fungsional yang berfokus pada identifikasi fitur sistem yang akan dikembangkan. Hasil analisis tersebut kemudian dijadikan dasar dalam merancang arsitektur sistem REST API dengan menggunakan JSON Web Tokens (JWT) sebagai metode otentikasi dan otorisasi untuk memastikan keamanan dan keandalan komunikasi antar komponen sistem. Desain yang telah diimplementasikan diuji melalui metode *black box testing* sesuai kebutuhan fungsional, meliputi autentikasi, operasi CRUD, dan multiplatform, dengan menggunakan perangkat lunak Postman sebagai alat uji.

## 2.2 Komunikasi

Peneliti mengimplementasikan metode diskusi sebagai pendekatan utama dalam tahap komunikasi dengan klien. Terdapat dua klien utama yang terlibat dalam penelitian ini, yaitu pemerintahan desa sebagai pengguna akhir dan pengembang IT sebagai pihak yang akan memanfaatkan sistem basis data yang dibuat oleh peneliti. Melalui diskusi ini, peneliti secara mendalam membahas berbagai aspek penting, termasuk kebutuhan, masalah, dan harapan yang berkaitan dengan pengembangan basis data monografi desa terpadu yang menggunakan REST API sebagai kerangka kerja utama. Tujuan utama diskusi ini adalah untuk memastikan bahwa sistem yang akan dibangun dapat memenuhi kebutuhan operasional klien dan diimplementasikan secara efektif oleh pengembang IT.

## 2.3 Analisis Kebutuhan

Pada tahap ini, peneliti akan melakukan analisis mendalam mengenai kebutuhan pengguna berdasarkan permasalahan yang telah dirumuskan pada tahap komunikasi dengan klien. Analisis ini dilakukan dalam dua tahap utama, yaitu analisis kebutuhan pengguna dan analisis kebutuhan fungsional sistem. Analisis kebutuhan pengguna bertujuan untuk memahami apa saja kebutuhan spesifik dari pengguna. Pada Tabel 1 merupakan analisis kebutuhan terhadap pemerintahan desa, sedangkan pada Tabel 2 merupakan analisis kebutuhan terhadap pengembang IT.

Tabel 1 Kebutuhan Pemerintah

| No | Kebutuhan   |
|----|---|
| 1  | Pemerintahan desa membutuhkan sebuah platform yang berguna sebagai sarana informasi terpadu |
| 2  | Pemerintah dapat melakukan integrasi platform dengan pihak lain yang diajak kerja sama.     |
| 3  | Pemerintah dapat melakukan pengisian data secara dinamis.                                   |

Tabel 2 Kebutuhan Pengembang IT

| No | Kebutuhan   |
|----|---|
| 1  | Pengembang memerlukan sebuah basis data relational yang skalable            |
| 2  | Pengembang dapat mengelola data sesuai dengan kebutuhan                     |
| 3  | Pengembang memerlukan standar keamanan dalam proses pengelolaan data        |
| 4  | Pengembang memerlukan standar pertukaran data dalam proses pengelolaan data |

Analisis Kebutuhan Fungsional dilakukan untuk merancang fungsionalitas sistem agar sesuai dengan kebutuhan pengguna. Melalui analisis ini, diharapkan masalah yang dihadapi oleh pengguna dapat diselesaikan dengan efektif. Kebutuhan fungsionalitas sistem yang diperlukan untuk membangun basis data terintegrasi untuk monografi desa mencakup Tabel 3 yang ditujukan untuk pemerintahan desa dan Tabel 4 yang ditujukan untuk pengembang IT.

Tabel 3 Kebutuhan Fungsional Terhadap Pemerintah

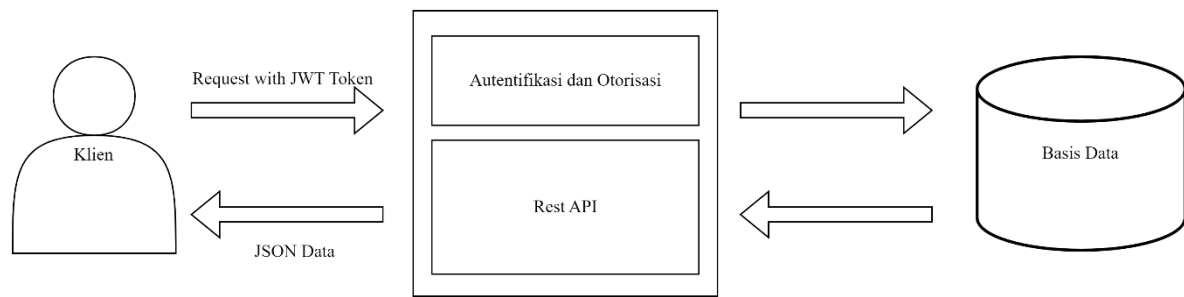
| No | Kebutuhan  |
|----|--|
| 1  | Sistem menyediakan basis data yang dapat menampung berbagai jenis data monografi desa, seperti data umum, data personil, data kewenangan, data keuangan, dan data kelembagaan. |
| 2  | Sistem harus dapat terintegrasi dengan berbagai platform dan memastikan konsistensi dan keakuratan data yang terintegrasi dengan berbagai platform.                            |
| 3  | Sistem menyediakan akses data rekapitulasi monografi.  |

Tabel 4 Kebutuhan Fungsional Terhadap Pengembang IT

| No | Kebutuhan  |
|----|--|
| 1  | Sistem menyediakan layanan register dan login pengguna yang berguna untuk mendapatkan token autentikasi. |
| 2  | Sistem menggunakan format JSON dalam penyajian dan penginputan data.                                     |
| 3  | Sistem menyediakan JSON Web Token sebagai alat autentikasi dan otorisasi dalam proses pengolahan data.   |
| 4  | Sistem menyediakan akses untuk memasukkan data kedalam basis data.                                       |
| 5  | Sistem menyediakan akses untuk mengubah data yang berada dalam basis data.                               |

## 2.4 Desain

Dalam tahap ini peneliti merancang dua sistem yang akan diterapkan dalam sistem. Gambar 2 merupakan rancangan arsitektur sistem Restfull API yang mendasari alur penggunaan sistem ini. Pada Gambar 3 merupakan rancangan alur autentikasi dan otorisasi JSON Web Token, sistem keamanan yang digunakan pada saat pertukaran data Restfull API.



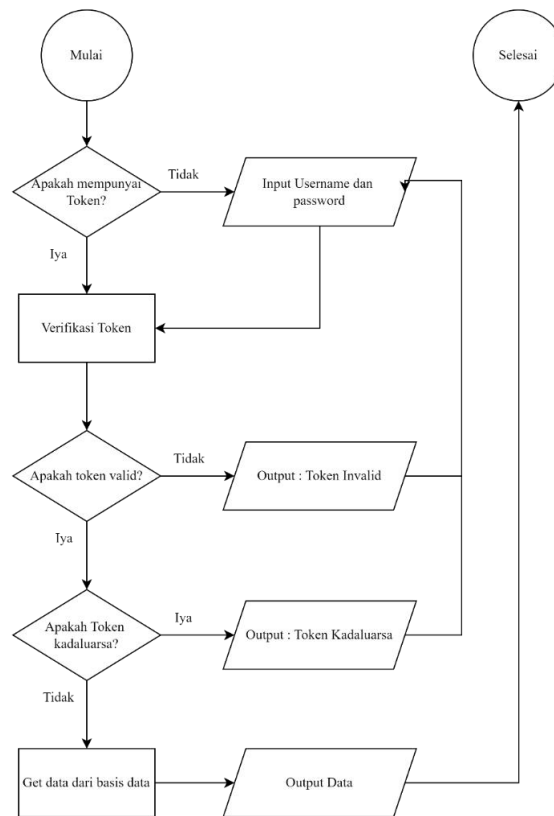
Gambar 2 Arsitektur REST API

Arsitektur sistem REST API terdiri dari tiga komponen utama, seperti yang ditunjukkan pada Gambar 2. Komponen pertama adalah klien, yang merupakan entitas yang berinteraksi langsung dengan *server* API. Klien ini dapat berupa aplikasi web, mobile, atau desktop. Komponen kedua adalah *server* API, yang mencakup modul Autentikasi dan Otorisasi yang bertanggung jawab untuk memverifikasi klien menggunakan standar JSON Web Token. *Server* API ini juga mencakup REST API yang berfungsi sebagai antarmuka untuk menangani permintaan HTTP (*Get*, *Put*, *Post*, *Delete*) sesuai dengan sumber daya yang ada di basis data. Komponen ketiga adalah basis data, yang berfungsi sebagai tempat penyimpanan data dan melayani proses CRUD (*Create*, *Read*, *Update*, *Delete*) yang diperintahkan oleh Restful API melalui protokol HTTP.

Alur sistem REST API, seperti yang ditunjukkan pada Gambar 2, dimulai dengan permintaan dari klien. Klien mengirim permintaan kepada REST API disertai dengan token JWT pada setiap header permintaan. Permintaan dari klien dapat berupa *Get*, *Put*, *Post*, atau *Delete*. Setiap permintaan yang diterima oleh REST API akan diperiksa oleh modul autentikasi dan otorisasi untuk memvalidasi keabsahan token JWT. Jika token yang disertakan pada header permintaan valid, permintaan

akan dilanjutkan ke REST API. REST API kemudian memproses permintaan sesuai dengan protokol HTTP yang digunakan, dan berkomunikasi dengan basis data untuk memasukkan atau mengubah data yang diperlukan. Rest API akan merespons kembali kepada klien dengan data yang diminta, status *server*, dan pesan yang relevan.

Pada Gambar 3 dijelaskan mengenai alur verifikasi JSON Web Token pada *server* API. Proses verifikasi dimulai ketika klien mengirimkan permintaan kepada *server*. *Server* kemudian memeriksa apakah klien menyertakan token pada header permintaan. Jika klien tidak memiliki token, klien akan diminta untuk login menggunakan username dan password guna mendapatkan token. Setelah klien memperoleh token, *server* akan memverifikasi token yang terdapat dalam header. Proses verifikasi token mencakup pemeriksaan tanda tangan (*signature*) dan masa berlaku token. Jika token tidak valid, *server* akan merespons dengan "Token Invalid". Jika token valid tetapi telah kedaluwarsa, *server* akan merespons dengan "Token Kedaluwarsa". Apabila token valid dan belum kedaluwarsa, *server* akan melanjutkan dengan meminta data dari basis data sesuai permintaan klien. Setelah verifikasi token berhasil, sistem akan mengirimkan data dalam format JSON kepada klien.



Gambar 3 Alur JSON Web Token

### 3 HASIL DAN PEMBAHASAN

Dalam penelitian ini, peneliti menggunakan metode pengujian black box textng. Pengujian black box teksting didasarkan pada kebutuhan fungsional. Pengujian dibagi menjadi tiga bagian yaitu, pengujian autentifikasi, pengujian CRUD, pengujian MultiPlatform.

#### 3.1 Pengujian Autentifikasi

Dalam pengujian autentikasi, peneliti menguji fungsionalitas *register* pengguna, *login* pengguna, dan autentikasi *JSON Web Token* dari sistem yang telah dikembangkan. Pengujian ini dilakukan menggunakan *software Postman* sebagai alat pengujian. Daftar pengujian autentifikasi menggunakan metode *blackbox* ditampilkan pada Tabel 5, Tabel 6, dan Tabel 7

Tabel 5 Pengujian Register

| Id   | Deskripsi Pengujian  | Perkiraan Hasil  | Hasil Pengujian   | Kecepatan Respon | Kesimpulan |
|------|--|--|---|------------------|------------|
| 1A01 | Pengguna menggunakan endpoint <code>./user/sign-up</code> dan mengisi <code>userName</code> dan password dengan format JSON di body                      | Sistem akan membuat user baru dan merespon dengan message "User created" | Sistem membuat user baru dan merespon dengan message "User created" | 136 ms           | Berhasil   |
| 1A02 | Pengguna menggunakan endpoint <code>./user/sign-up</code> dan mengisi <code>userName</code> yang telah digunakan dan password dengan format JSON di body | Sistem akan merespon dengan message "User already exist"                 | Sistem merespon dengan message "User already exist"                 | 47 ms            | Berhasil   |

Pengujian fungsionalitas pertama pada Tabel 5 ditandai dengan ID 1A01. Pengujian ini bertujuan untuk menguji proses pendaftaran pengguna yang diperlukan untuk mendapatkan token akses. Melalui endpoint `localhost:4000/user/sign-up`, peneliti memasukkan username "akhmad" dan password "123456" dalam format JSON. Jika username belum terdaftar, sistem akan melakukan pendaftaran dengan mencatat username dan password, yang kemudian di-hash dalam sistem. Selanjutnya, sistem akan merespons dengan pesan "user created" beserta

data pengguna seperti username, password yang telah di-hash, serta `updatedAt` dan `createdAt`. Berdasarkan pengujian yang telah dilakukan sistem berhasil memenuhi rancangan yang telah ditetapkan dengan kecepatan respon 136 ms

Pengujian dengan ID 1A02 pada Tabel 5 menguji verifikasi username yang sudah terdaftar. Melalui operasi *POST* dengan endpoint `localhost:4000/user/sign-up`, peneliti memasukkan username "rama" yang telah terdaftar sebelumnya dan password dalam format *JSON*. Sistem kemudian

akan mendeteksi bahwa username tersebut sudah terdaftar dan memberikan respons dengan pesan "User Already Exist". Berdasarkan pengujian yang telah dilakukan, sistem dapat berjalan sesuai dengan

perkiraan hasil. Kecepatan respon dalam pengujian mencapai 47 ms.

Tabel 6 Pengujian Login

| Id   | Deskripsi Pengujian   | Perkiraan Hasil  | Hasil Pengujian   | Kecepatan Respon | Kesimpulan |
|------|---|--|---|------------------|------------|
| 1B01 | Pengguna menggunakan endpoint <code>./user/login</code> dan mengisi <code>userName</code> dan <code>password</code> dengan format <code>JSON</code> di <code>body</code> yang telah didaftarkan | Sistem akan merespon dengan message "authentication succes" dan <code>JSON Web Token</code> yang telah <i>digenerate</i> | Sistem merespon dengan message "authentication succes" dan <code>JSON Web Token</code> yang telah <i>digenerate</i> | 106 ms           | Berhasil   |
| 1B02 | Pengguna menggunakan endpoint <code>./user/login</code> dan mengisi <code>userName</code> dan <code>password</code> yang salah dengan format <code>JSON</code> di <code>body</code>             | Sistem akan merespon dengan message "UserName or password incorect"  | Sistem merespon dengan message "UserName or password incorect"  | 15 ms            | Berhasil   |

Pengujian 1B01 pada Tabel 6 menguji fungsionalitas login pengguna untuk mendapatkan akses *JSON Web Token*. Melalui operasi *POST* dengan endpoint `localhost:4000/user/login`, peneliti memasukkan `username` "rama" dan `password` "123456" yang telah didaftarkan dalam format *JSON* pada bagian `body`. Sistem akan melakukan verifikasi terhadap `username` dan `password` yang dimasukkan. Apabila `username` dan `password` sesuai dengan data yang ada di sistem, maka sistem akan merespons dengan pesan "authentication success" dan *JSON Web Token* yang telah di-generate oleh sistem. Hasil dari pengujian yang telah dilakukan, sistem dapat memenuhi perkiraan hasil dengan kecepatan 106 ms.

Pengujian 1B02 pada Tabel 6 bertujuan untuk menguji verifikasi `username` dan `password` ketika salah satu data yang digunakan tidak sesuai dengan data yang ada di basis data. Melalui operasi *POST* dengan endpoint `localhost:4000/user/login`, peneliti memasukkan `username` "ramaz" dan `password` "123456" yang tidak terdaftar di basis data dalam format *JSON* pada bagian `body`. Sistem kemudian akan memverifikasi `username` dan `password` tersebut, dan memberikan respons dengan pesan "username or password incorrect". Hasil dari pengujian ini menunjukkan sistem dapat memverifikasi login pengguna yang tidak sesuai dengan kecepatan respon 15 ms.

Tabel 7 Pengujian *JSON Web Token*

| Id   | Deskripsi Pengujian   | Perkiraan Hasil                                     | Hasil Pengujian                                | Kecepatan Respon | Kesimpulan |
|------|---|---|--|------------------|------------|
| 1C01 | Pengguna melakukan request data di endpoint yang sesuai, dan menyertakan <i>JSON Web Token</i> di header  | Sistem akan merespon sesuai dengan request pengguna | Sistem merespon sesuai dengan request pengguna | 15 ms            | Berhasil   |
| 1C02 | Pengguna melakukan request data di endpoint yang sesuai, dan menyertakan <i>JSON Web Token</i> salah di header atau tidak menyertakan <i>JSON Web Token</i> | Sistem akan merespon dengan message "Invalid Token" | Sistem merespon dengan message "Invalid Token" | 8 ms             | Berhasil   |
| 1C03 | Pengguna melakukan request data di endpoint yang sesuai, dan menyertakan <i>JSON Web Token</i> yang telah kadaluarsa di header                              | Sistem akan merespon dengan message "jwt expired"   | Sistem merespon dengan message "jwt expired"   | 8 ms             | Berhasil   |

Pengujian dengan ID 1C01 pada Tabel 7 bertujuan untuk menguji verifikasi *JSON Web Token* yang diberikan saat login. Melalui operasi *GET* pada salah satu endpoint yang telah disediakan, yaitu `localhost:4000/dataPenduduk`, peneliti memasukkan token yang telah diperoleh ke dalam bagian *Authorization*. Sistem akan memverifikasi token yang diinputkan oleh peneliti, dan jika token valid, sistem akan merespons dengan memanggil data sesuai dengan permintaan peneliti. Hasil dari pengujian sistem dapat merespon sesuai dengan respon pengguna dengan kecepatan 15 ms.

Pengujian dengan ID 1C02 pada Tabel & bertujuan untuk menguji kemampuan sistem dalam memverifikasi token yang tidak cocok dengan token yang telah dihasilkan sebelumnya. Pengujian dilakukan dengan menggunakan operasi *GET* pada

salah satu *endpoint* yang tersedia, yaitu `localhost:4000/dataPenduduk`. Dalam pengujian ini, peneliti memasukkan token yang tidak sesuai pada bagian *Authorization*. Sistem kemudian akan melakukan proses verifikasi terhadap token tersebut. Jika token tidak valid, sistem akan memberikan respons berupa pesan "Invalid Token" kepada pengguna. Berdasarkan hasil pengujian sistem dapat memverifikasi token yang tidak valid dan mengembalikan respon yang sesuai dengan kecepatan 8 ms.

Pengujian terakhir dalam bagian Autentifikasi adalah pengujian dengan ID 1C03. Pada Tabel 7 Tujuan dari pengujian ini adalah untuk memverifikasi token yang telah kadaluarsa. Sistem yang dikembangkan membatasi penggunaan token hingga 60 detik, dan jika token telah melewati batas

waktu tersebut, sistem akan menganggapnya kadaluarsa. Pengujian dilakukan dengan menggunakan operasi *GET* pada salah satu *endpoint* yang tersedia, yaitu *localhost:4000/dataPenduduk*. Dalam pengujian ini, peneliti memasukkan token yang sudah berumur lebih dari 60 detik pada bagian *Authorization*. Sistem kemudian melakukan verifikasi dan memberikan respons kepada pengguna dengan pesan "invalid token" dan "jwt expired". Berdasarkan hasil pengujian sistem dapat memverifikasi token yang telah kadaluarsa dengan kecepatan 8ms.

Melalui pengujian autentikasi yang telah dilakukan, sistem berhasil memenuhi semua kebutuhan fungsional yang telah dirumuskan oleh peneliti sebelumnya. Dalam pelaksanaan proses autentikasi, sistem berhasil menjalankan pendaftaran pengguna, memanfaatkan fitur login, dan menghasilkan token yang diperlukan untuk menentukan hak akses dalam pengelolaan basis data. Keberhasilan ini menunjukkan bahwa sistem dapat

beroperasi dengan baik dalam mendukung fungsi-fungsi yang telah ditetapkan.

### 3.2 Pengujian CRUD

Pada tahap pengujian CRUD, peneliti menguji fungsi utama dalam pengelolaan basis data, yaitu Create, Read, Update, dan Delete. Fungsi *Create* digunakan untuk mencatat data baru ke dalam basis data dan dilakukan menggunakan operasi *POST* sesuai aturan HTTP. Fungsi *Read* berperan untuk membaca data yang sudah ada di basis data, yang diakses melalui operasi *GET* sesuai aturan HTTP, sehingga data tersebut dapat diberikan kepada pengguna. Fungsi *Update* digunakan untuk mengubah data yang tersimpan dalam basis data, yang dilakukan dengan operasi *PATCH* sesuai aturan HTTP. Terakhir, fungsi *Delete* berfungsi untuk menghapus data dalam basis data, yang dilaksanakan melalui operasi *DELETE* sesuai aturan HTTP. Daftar pengujian beserta hasilnya disajikan dalam Tabel 8.

Tabel 8 Pengujian CRUD

| <b>Id</b> | <b>Deskripsi Pengujian</b>  | <b>Perkiraan Hasil</b>  | <b>Hasil Pengujian</b>   | <b>Kecepatan Respon</b> | <b>Kesimpulan</b> |
|-----------|---|---|--|-------------------------|-------------------|
| 2A01      | Pengguna melakukan request operasi post sesuai dengan endpoint yang akan di akses, menyertakan data sesuai dengan field yang disediakan JSON Web Token yang telah di dapat            | Sistem akan mencatat data dan merespon dengan message "Data Created Successfully" | Sistem mencatat data dan merespon dengan message "Data Created Successfully" | 115 ms                  | Berhasil          |
| 2B01      | Pengguna melakukan request operasi GET sesuai dengan endpoint yang akan diakses, dan menyertakan JSON Web Token yang telah di dapat   | Sistem akan merespon dengan mengambil data yang diminta oleh pengguna             | Sistem merespon dengan mengambil data yang diminta oleh pengguna             | 18 ms                   | Berhasil          |
| 2C01      | Pengguna melakukan request operasi PATCH sesuai dengan endpoint dan id data yang akan diubah, menyertakan data sesuai dengan field yang disediakan JSON Web Token yang telah di dapat | Sistem akan merespon dengan message "Data Updated Successfully"                   | Sistem merespon dengan message "Data Updated Successfully"                   | 49 ms                   | Berhasil          |
| 2D01      | Pengguna melakukan request operasi DELETE sesuai dengan endpoint dan id data yang akan dihapus  | Sistem akan merespon dengan message "Data Has Been Deleted"                       | Sistem merespon dengan message "Data Has Been Deleted"                       | 19 ms                   | Berhasil          |

Pengujian *Create* dengan ID 2A01 pada Tabel 8 bertujuan untuk menguji fungsionalitas sistem dalam mencatat data baru. Pengujian dilakukan melalui salah satu *endpoint* yang telah disediakan, yaitu *localhost:4000/dataPenduduk*, dengan operasi *POST*. Pada bagian *body*, peneliti memasukkan data berupa nomorIndukPenduduk, nama, tempatLahir, tglLahir, namaIbu, jenisKelamin\_id, pekerjaan\_id, pendidikanTerakhir\_id, agama\_id, status\_id, shdk\_id, dusun\_id, rukunWarga\_id, rukunTetangga\_id serta menyertakan *JSON Web Token* di bagian *Authorization*. Setelah permintaan dikirim ke API, sistem akan mencatat data yang dimasukkan oleh pengguna ke dalam basis data. Setelah proses pencatatan selesai, sistem akan memberikan respons berupa pesan "Data Created Successfully" dan mengembalikan data yang telah dicatat kepada pengguna. Hasil dari pengujian menunjukkan bahwa sistem dapat melakukan fungsi create dan merespon kembali dengan kecepatan 115 ms.

Pengujian dengan ID 2B01 pada Tabel 8 bertujuan untuk menguji fungsionalitas sistem dalam membaca data dari basis data dan mengembalikannya kepada pengguna dalam format *JSON*. Pengujian dilakukan melalui salah satu *endpoint* yang tersedia, yaitu *localhost:4000/dataPenduduk*, menggunakan operasi *GET* yang disertai *JSON Web Token* pada bagian *Authorization*. Setelah permintaan dikirim, sistem akan membaca data dari basis data dan mengembalikannya kepada pengguna. Hasil dari pengujian menunjukkan bahwa sistem dapat menjalankan fungsi *READ* dengan kecepatan 18ms

Pengujian dengan ID 2C01 pada Tabel 8 bertujuan untuk menguji fungsionalitas sistem dalam melakukan perubahan data di dalam basis data. Peneliti melakukan pengujian menggunakan operasi *PATCH* pada salah satu *endpoint*, yaitu *localhost:4000/dataPenduduk/51*. Angka 51 di akhir menunjukkan identifikasi dari ID data yang akan diubah. Peneliti mengubah data dengan menuliskan



seluruh atribut yang ada dan menyesuaikannya sesuai kebutuhan. Dalam pengujian ID 2C01, peneliti mengganti nama penduduk dari "AILA DELINARA UNTOROI" menjadi "AILA DELINARA UNTOROI Updated". Setelah itu, sistem akan memproses permintaan pembaruan yang dikirimkan dan mengembalikan data yang telah diperbarui kepada pengguna dengan pesan "Data Updated Successfully". Berdasarkan hasil pengujian, sistem dapat menjalankan fungsi *UPDATE* dengan kecepatan 49 ms.

Pengujian terakhir bertujuan untuk menguji fungsionalitas sistem dalam menghapus data di dalam basis data. Pengujian ini ditandai dengan ID 2D01 pada Tabel 4.26. Dalam pengujian ini, peneliti menggunakan operasi *DELETE* untuk melakukan penghapusan data. Peneliti menggunakan salah satu *endpoint*, yaitu *localhost:4000/dataPenduduk/51*, di mana angka 51 di akhir menunjukkan ID data yang akan dihapus. Setelah menerima permintaan penghapusan, sistem akan menghapus data sesuai dengan permintaan tersebut dan mengirimkan respons kepada pengguna dengan pesan "Data has been deleted". Hasil dari pengujian menunjukkan bahwa sistem dapat menjalankan fungsi *DELETE* dengan kecepatan 19 ms

Create, Read, Update, dan Delete (CRUD) adalah salah satu fungsi utama dalam pengelolaan basis data. Dengan diterapkannya fungsi CRUD secara efektif dan menghasilkan waktu respons yang cepat dalam sistem yang dikembangkan, kebutuhan untuk mengelola data dapat terpenuhi dengan baik.

Hal ini menunjukkan bahwa sistem mampu melaksanakan berbagai operasi dasar yang diperlukan untuk memanipulasi dan mengelola data secara efisien.

### 3.3 Pengujian Multi Platform

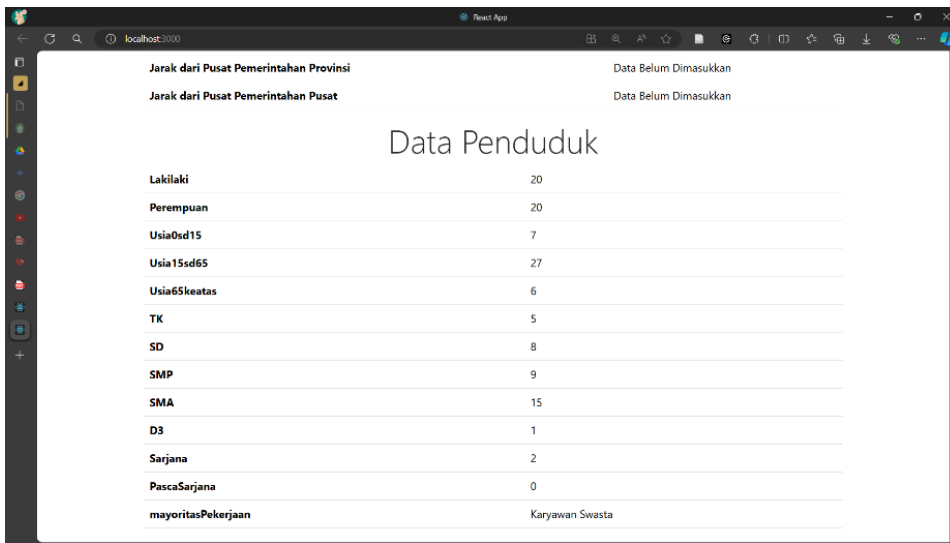
Pengujian multiplatform dilakukan untuk memastikan bahwa sistem dapat berfungsi dengan baik dan terintegrasi secara lancar pada berbagai platform. Tujuan dari pengujian ini adalah untuk mengukur kemampuan sistem dalam mengelola permintaan dan respons dari berbagai platform, serta memastikan bahwa tidak ada gangguan dalam proses integrasi data antar platform. Dalam hal ini, peneliti memutuskan untuk menggunakan dua platform yang paling umum, yaitu situs web dan aplikasi Android, sebagai alat uji. Peneliti merancang situs web dan aplikasi Android yang keduanya menggunakan REST API sebagai penghubung utama untuk berkomunikasi dengan server. Proses fetching data dilakukan melalui *endpoint* yang telah disediakan pada server, yang bertugas mengelola permintaan data dan memberikan respons yang sesuai. Setelah data diambil dari server, baik situs web maupun aplikasi Android akan menampilkan informasi tersebut kepada pengguna dalam format yang sesuai dengan masing-masing platform. Deskripsi pengujian Multi Platform menggunakan metode *black box* tertera pada Tabel 9

Tabel 9 Pengujian Multi Platform

| Id   | Fungsi                                | Deskripsi Pengujian  | Perkiraan Hasil  | Hasil Pengujian   | Kesimpulan |
|------|---------------------------------------|--|--|---|------------|
| 3A01 | Aksesibilitas dengan Platform Website | Pengguna melakukan <i>fetching</i> data ke platform melalui <i>endpoint</i> yang telah disediakan. Proses <i>fetching</i> data ini dilakukan dalam pengembangan kode situs web.                | Situs web akan menampilkan data yang diperoleh dari respons <i>server</i> REST API.      | Situs web menampilkan data yang diperoleh dari respons <i>server</i> REST API | Berhasil   |
| 3B01 | Aksesibilitas dengan platform android | Pengguna melakukan <i>fetching</i> data ke platform Android melalui <i>endpoint</i> yang telah disediakan. Proses <i>fetching</i> data ini dilakukan dalam pengembangan kode aplikasi Android. | Aplikasi android akan menampilkan hasil data <i>fetching</i> dari <i>server</i> REST API | Aplikasi menampilkan hasil data <i>fetching</i> dari <i>server</i> REST API   | Berhasil   |

Pengujian pengambilan data dari basis data terpadu monografi dengan ID 3A01 pada Tabel 9 dilakukan melalui situs web yang telah dikembangkan. Peneliti menggunakan alat paket *React JS* untuk menulis kode situs web tersebut. Dalam proses ini, peneliti menerapkan operasi *GET* sesuai dengan protokol *HTTP* untuk memperoleh data dari basis data. *Endpoint* yang digunakan untuk

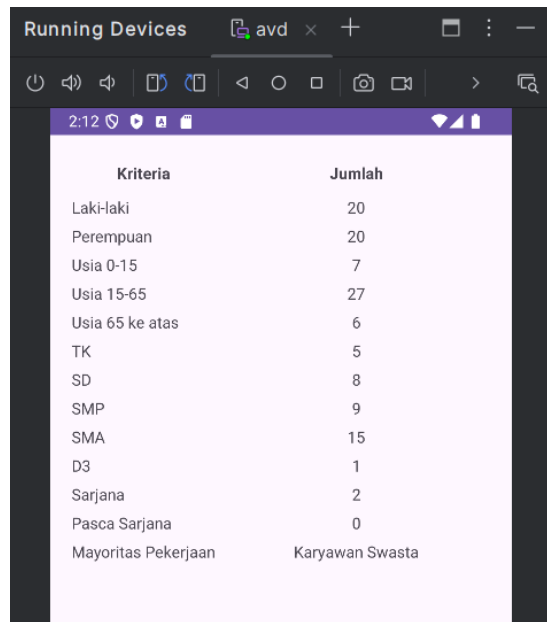
*fetching* adalah *localhost:4000/view/Penduduk*, yang menyajikan rekap data penduduk yang tersimpan di dalam basis data. Setelah data berhasil diambil, *server* mengirimkan respons dalam format *JSON* ke situs web. Respons ini kemudian diproses dan ditampilkan dalam bentuk tabel yang mencantumkan nilai-nilai data, sesuai dengan tampilan yang diperlihatkan dalam Gambar 4.



Gambar 4 Tampilan Fetching Data Website

Pengujian 3B01 pada Tabel 9 yaitu aksesibilitas server RestAPI terhadap aplikasi mobile menggunakan platform android sebagai alat pengujian. Peneliti mengembangkan aplikasi android mobile sederhana menggunakan android studio untuk melakukan fetching data dari server RestAPI untuk di tampilkan kepada pengguna. Pengujian menggunakan operasi GET pada endpoint

localhost:4000/view/Penduduk untuk mengakses rekap data penduduk pada basis data. Setelah mengirim permintaan kepada server, server akan mengirim respon rekap data penduduk kepada aplikasi android dengan format JSON. Aplikasi akan memproses data tersebut dan menampilkannya dalam bentuk tabel kepada pengguna seperti pada Gambar 5.



Gambar 5 Tampilan Fetching Data Android Studio

Pengujian multiplatform telah menunjukkan bahwa basis data dapat terhubung dengan berbagai platform yang berbeda. Dalam konteks ini, basis data mampu berkomunikasi secara efektif dengan platform situs web serta aplikasi mobile berbasis Android. Hasil ini menegaskan fleksibilitas sistem dalam menjalin koneksi lintas platform, sehingga

memungkinkan integrasi yang lebih luas dalam pengelolaan data.

#### 4 DISKUSI

Penelitian sebelumnya yang berjudul "Penerapan JSON Web Token sebagai Strategi Pengamanan Data pada Aplikasi MultiMasjid" oleh Astowo

menunjukkan bahwa penerapan JSON Web Token (JWT) dapat memastikan keamanan dalam pengelolaan data. Dalam penelitian tersebut, metode autentikasi menggunakan JWT diterapkan sebagai langkah pengamanan data. Pada penelitian ini, objek penelitian diubah dengan memanfaatkan data monografi desa, sehingga pengelolaan data desa dilakukan sesuai dengan hak akses yang telah ditentukan. Dengan memanfaatkan teknologi yang diadaptasi dari penelitian yang dilakukan oleh Astowo, penelitian ini berhasil menghasilkan sistem basis data terintegrasi untuk monografi desa yang aman, menggunakan teknologi REST API dan JSON Web Token (JWT). Kecepatan yang diperoleh selama pengujian menunjukkan bahwa aksesibilitas dan pengelolaan data desa dapat berlangsung secara optimal. Hal ini juga mendukung pemerintahan menjadi good governance yang menerapkan optimilisasi dan transparansi data dalam penyelenggaraan pemerintahan [17].

Penelitian mengenai pengembangan basis data pemerintahan dengan teknologi REST API telah dilakukan oleh Febriansyah dengan judul Perancangan Sistem Informasi E-Arsip Berbasis Website Pada Desa Wateswinangun. Namun dalam penelitian tersebut tidak menerapkan sistem keamanan dalam menjaga data desa. Data desa yang merupakan aset pembangunan sudah seharusnya dilindungi. Dalam penelitian ini kemudian diimplementasikan sistem keamanan JSON Web Token untuk melindungi pertukaran

## 5 KESIMPULAN

Data merupakan aset penting bagi pembangunan dan pemberdayaan masyarakat desa. Dengan menjaga pengelolaan dan aksesibilitas data, pemerintah dapat mencapai good governance. Hasil penelitian ini berhasil mengembangkan sistem basis data yang dapat terintegrasi di tingkat pemerintahan desa secara aman dengan menggunakan teknologi REST API dan JSON Web Token. Dengan sistem ini, pengelolaan data di tingkat pemerintahan desa dapat dilakukan secara efektif. Informasi yang diterima oleh masyarakat pun terhindar dari ambiguitas dan menjadi lebih akurat. Penerapan sistem ini dalam penyelenggaraan pemerintahan

data melalui REST API. Dengan ini data desa dapat terjaga aksesibilitasnya dari tangan yang tidak bertanggung jawab. Penelitian ini juga mengembangkan basis data menjadi cakupan yang lebih besar yaitu monografi. Monografi sendiri dapat menjadi sumber data yang akurat untuk menggambarkan seluruh keadaan desa, dan dapat menjadi dasar dari pembangunan pemberdayaan masyarakat [17] [18].

Kesiapan pemerintah desa dalam memanfaatkan teknologi ini perlu diperhatikan. Tingkat kesulitan dalam implementasi sistem oleh pemerintah secara langsung memberikan kesempatan bagi pihak swasta untuk berkontribusi dalam penyelenggaraan pemerintahan tingkat desa. Kolaborasi antara pihak pemerintah desa dan sektor swasta dapat menjadi alternatif untuk mengoptimalkan penggunaan basis data terintegrasi di tingkat desa [3]. Dengan demikian, proses transformasi menuju e-governance dapat lebih mudah tercapai. Penelitian ini merupakan langkah awal dalam membangun pengelolaan basis data desa sebagai sumber informasi. Sistem yang dihasilkan dapat menjadi referensi bagi pihak-pihak yang membutuhkan, sehingga akurasi data desa dapat terjaga dengan baik. Dalam sisi pengelolaan data desa, sistem ini memungkinkan mengelola data secara efisien dan efektif. Sistem ini dapat menyesuaikan data desa di masa depan dengan meningkatkan skala menjadi lebih besar, karena sistem ini dirancang untuk bersifat skalabel.

desa dapat meningkatkan efisiensi pelayanan publik.

Penggunaan REST API dalam penelitian ini berfungsi sebagai sistem komunikasi untuk mengelola data dalam basis data. Dengan penerapan JSON Web Token, hak akses dalam REST API dibatasi hanya kepada pemilik token. Setiap pemilik token dapat mengelola data sesuai dengan token yang dimiliki. Dalam pengembangan selanjutnya, disarankan untuk menambahkan peran dalam JSON Web Token, sehingga fungsi Create, Update, dan Delete dapat dilakukan oleh pemegang hak akses tertentu. Dengan demikian, tidak semua pemilik token akan memiliki kemampuan untuk menambah, mengubah, atau menghapus data desa.

## DAFTAR PUSTAKA

- [1] R. Septima dan I. Zulfa, "Pelatihan Ilmu Teknologi Komputer Dalam Meningkatkan Kemampuan Administrasi Perangkat Desa Tan Saril," *JPKMI (Jurnal Pengabdian Kepada Masyarakat Indonesia)*, vol. 1, no. 1, Art. no. 1, Feb 2020, doi: 10.36596/jpkmi.v1i1.10.
- [2] N. S. Warman, S. Syamsir, M. Maldini, O. Nurhasanah, N. R. Oktariandani, dan I. H. Syafikruzi, "Implementasi Inovasi Kebijakan
- Dalam Penyelenggaraan Sistem Pemerintahan Berbasis Elektronik (SPBE) di Kota Pekanbaru," *PROSIDING SEMINAR NASIONAL PENDIDIKAN, BAHASA, SASTRA, SENI, DAN BUDAYA*, vol. 1, no. 2, Art. no. 2, Nov 2022, doi: 10.55606/mateandrau.v1i2.161.
- [3] S. Susniwati dan M. Zamili, "Acceleration of One Indonesian Data through Collaborative Governance in Indonesia," *Publik (Jurnal Ilmu Administrasi)*, vol. 11, no. 2, hlm. 166–

- 177, Des 2022, doi: 10.31314/pjia.11.2.166-177.2022.
- [4] M. Qi dan J. Wang, "Using the Internet of Things E-Government Platform to Optimize the Administrative Management Mode," *Wireless Communications and Mobile Computing*, vol. 2021, no. 1, hlm. 2224957, 2021, doi: 10.1155/2021/2224957.
- [5] S. Groppe dan J. Groppe, "Hybrid Multi-model Multi-platform (HM3P) Databases," dipresentasikan pada International Conference on Data Science, Technology and Applications, SCITEPRESS, Jul 2020, hlm. 177–184. doi: 10.5220/0009802401770184.
- [6] R. A. Budiantoro, M. R. W. Aji, J. Prayogi, dan A. Susanti, "SI Desa (Sistem Informasi Desa): Inisiasi Sistem Informasi dan Pengelolaan Data Ekonomi Desa yang Terintegrasi oleh Pemerintah Provinsi Jawa Timur," *Jurnal Bina Praja*, vol. 15, no. 2, Art. no. 2, Agu 2023, doi: 10.21787/jbp.15.2023.233-247.
- [7] Z. Zaidir dan I. Listiawan, "Sistem Terintegrasi untuk Perencanaan dan Pengendalian Pembangunan Wilayah Berbasis Kampung," *JISKA (Jurnal Informatika Sunan Kalijaga)*, vol. 5, no. 1, Art. no. 1, Mei 2020, doi: 10.14421/jiska.2020.51-07.
- [8] Kementerian Komunikasi dan Informasi Republik Indonesia, "Panduan Penyelenggaraan Situs Web Pemerintah Daerah."
- [9] "Permendagri No. 13 Tahun 2012 Tentang Monografi Desa Dan Kelurahan," Database Peraturan Perundang-undangan Indonesia - [PERATURAN.GO.ID]. Diakses: 3 Juni 2024. [Daring]. Tersedia pada: <https://peraturan.go.id/id/permendagri-no-13-tahun-2012>
- [10] I. Ahmad, E. Suwarni, R. I. Borman, Asmawati, F. Rossi and Y. Jusman, "Implementation of RESTful API Web Services Architecture in Takeaway Application Development," 2021 1st International Conference on Electronic and Electrical Engineering and Intelligent System (ICE3IS), Yogyakarta, Indonesia, 2021, pp. 132-137, doi: 10.1109/ICE3IS54102.2021.9649679.
- [11] R. Ramakrishnan, "Enhancing Data Management with MongoDB and its Rest API," *IJRASET*, vol. 11, no. 7, hlm. 189–193, Jul 2023, doi: 10.22214/ijraset.2023.54593.
- [12] G. Vega-Gorgojo, "CRAFTS: Configurable REST APIs for Triple Stores," *IEEE Access*, vol. 10, hlm. 32426–32441, 2022, doi: 10.1109/ACCESS.2022.3160610.
- [13] A. Y. Nashikhuddin, J. Karaman, dan Y. Litanianda, "IMPLEMENTASI API RESTFUL DENGAN JSON WEB TOKEN (JWT) PADA APLIKASI E-COMMERCE THRIFTY SHOP UNTUK OTENTIKASI DAN OTORISASI PENGGUNA," *METHOMIKA: Jurnal Manajemen Informatika & Komputerisasi Akuntansi*, vol. 7, no. 2, Art. no. 2, Okt 2023, doi: 10.46880/jmika.Vol7No2.pp239-246.
- [14] U. B. Astowo dan A. Sujarwo, "Penerapan JSON Web Token sebagai Strategi Pengamanan Data pada Aplikasi MultiMasjid," *Innovative: Journal Of Social Science Research*, vol. 3, no. 6, Art. no. 6, Des 2023, doi: 10.31004/innovative.v3i6.6908.
- [15] A. Setiawan dan A. I. Purnamasari, "Implementasi JSON Web Token Berbasis Algoritma SHA-512 untuk Otentikasi Aplikasi BatikKita," *Jurnal RESTI (Rekayasa Sistem dan Teknologi Informasi)*, vol. 4, no. 6, Art. no. 6, Des 2020, doi: 10.29207/resti.v4i6.2533.
- [16] F. D. K. Febriansyah, L. R. A, A. Alfianto, M. I. Lathoif, dan N. Adilah, "PERANCANGAN SISTEM INFORMASI E-ARSIP BERBASIS WEBSITE PADA DESA WATESWINANGUN," *ABDI MASSA: Jurnal Pengabdian Nasional (e-ISSN: 2797-0493)*, vol. 2, no. 06, Art. no. 06, Nov 2022.
- [17] M. E. Atmojo, E. Zuhriyati, dan N. A. Hanif, "Peningkatan Kualitas Pemerintah Desa Melalui Penyusunan Profil Desa Trimulyo," *Society: Jurnal Pengabdian Masyarakat*, vol. 1, no. 3, Art. no. 3, Apr 2022, doi: 10.55824/jpm.v1i3.88.
- [18] D. E. Rahmawati dan H. A. N. Rasyid, "Pendampingan Penyusunan Buku Profil Desa," *JCOMENT (Journal of Community Empowerment)*, vol. 3, hlm. 1–11, Des 2022.