

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



Disusun Oleh:

N a m a : Akhmad Ramadhan
NIM : 20524049

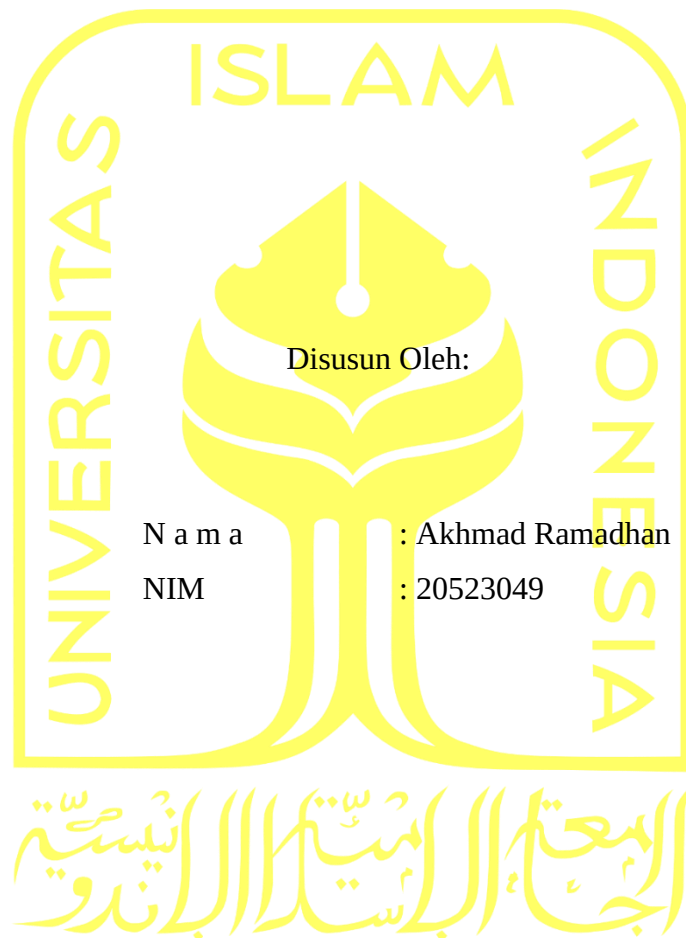
**PROGRAM STUDI INFORMATIKA – PROGRAM SARJANA
FAKULTAS TEKNOLOGI INDUSTRI
UNIVERSITAS ISLAM INDONESIA**

2024

HALAMAN PENGESAHAN DOSEN PEMBIMBING

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

TUGAS AKHIR



Yogyakarta, 28 Oktober 2024

Pembimbing,


(Sri Mulyati, S.Kom.,M.Kom)

HALAMAN PENGESAHAN DOSEN PENGUJI

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

TUGAS AKHIR

Telah dipertahankan di depan sidang penguji sebagai salah satu syarat untuk memperoleh gelar Sarjana Komputer dari Program Studi Informatika – Program Sarjana di Fakultas Teknologi Industri Universitas Islam Indonesia

Yogyakarta, 4 November 2024

Tim Penguji

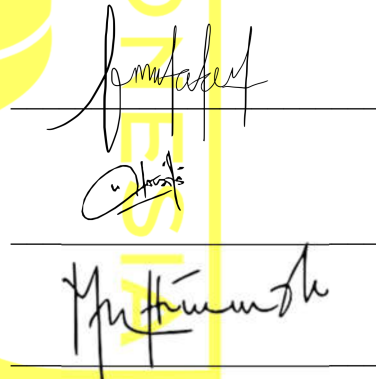
Sri Mulyati, S.Kom., M.Kom

Anggota 1

Dr. Novi Setiani, S.T., M.T.

Anggota 2

Izzati Muhimmah, S.T., M.Sc., Ph.D.

Three handwritten signatures of the examiners are placed over horizontal lines. The first signature is for Sri Mulyati, the second for Dr. Novi Setiani, and the third for Izzati Muhimmah.

Mengetahui,

Ketua Program Studi Informatika – Program Sarjana

Fakultas Teknologi Industri

Universitas Islam Indonesia

The official blue circular stamp of Universitas Islam Indonesia, Yogyakarta, Faculty of Industrial Technology is shown. Overlaid on the stamp is a blue ink signature of the Dean, Dhomas Hatta Fudholi.

(Dhomas Hatta Fudholi, S.T, M.Eng., Ph.D.)

HALAMAN PERNYATAAN KEASLIAN TUGAS AKHIR

Yang bertanda tangan di bawah ini:

Nama : Akhmad Ramadhan

NIM : 20523049

Tugas akhir dengan judul:

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

Menyatakan bahwa seluruh komponen dan isi dalam tugas akhir ini adalah hasil karya saya sendiri. Apabila di kemudian hari terbukti ada beberapa bagian dari karya ini adalah bukan hasil karya sendiri, tugas akhir yang diajukan sebagai hasil karya sendiri ini siap ditarik kembali dan siap menanggung risiko dan konsekuensi apapun.

Demikian surat pernyataan ini dibuat, semoga dapat dipergunakan sebagaimana mestinya.

Yogyakarta, 28 Oktober 2024

A pink rectangular stamp with a QR code on the left and a signature on the right. The text on the stamp includes 'METERAN ELEKTRONIK' and '1000'. The signature is written in black ink over the stamp.

(Akhmad Ramadhan)

HALAMAN PERSEMBAHAN

Bismillahirrahmanirahim, alhamdulillah puji syukur kepada Allah subhanahu wa ta'ala atas segala pemberian kesehatan, kemudahan, dan kelancaran dalam mengerjakan tugas akhir. Shalawat serta salam kepada Nabi Muhammad sallallahu ala muhammad sallallahu alaihi wasallam. Laporan ini saya persembahkan untuk keluarga yang selalu mendoakan, memberikan semangat, dan mendukung sepanjang proses penyusunan. Terima kasih atas dukungan luar biasa yang sangat berarti bagi saya.

HALAMAN MOTO

"Sesungguhnya bersama kesulitan ada kemudahan."

(QS. Al-Insyirah ayat 5)

*"Ketahuilah bahwa kemenangan bersama kesabaran, jalan keluar bersama kesulitan,
dan sesudah kesulitan ada kemudahan."*

(Rasulullah SAW)

"Belajarlah, kejar tujuanmu. Bapak mendukungmu disini"

(Alm.Kadarno)

KATA PENGANTAR

Assalamu'alaikum Wr. Wb.

Alhamdulillahirabbilalamin, puja dan syukur penulis panjatkan pada kehadiran Allah S.W.T. yang telah melimpahkan berkah, rahmat, dan karunia-Nya sehingga penulis dapat menyelesaikan tugas akhir dengan judul “IMPLEMENTASI JSON WEB TOKEN DALAM PENGEMBANGAN BASIS DATA MONOGRAFI DESA BERBASIS REST API”.

Skripsi ini ditulis sebagai salah satu syarat untuk menyelesaikan program pendidikan Strata Satu Informatika, Fakultas Teknologi Industri di Universitas Islam Indonesia. Penulis menyadari bahwa skripsi ini dapat diselesaikan berkat dukungan, bantuan dan doa dari berbagai pihak. Oleh karena itu, penulis ingin berterima kasih sebesar-besarnya kepada semua pihak yang telah berkontribusi dalam penyelesaian skripsi ini. Secara khusus penulis mengucapkan terimakasih kepada :

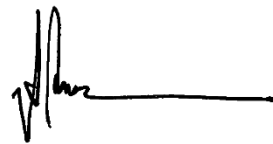
1. Kedua orang tua penulis yang selalu memberikan kasih sayang, dukungan serta doa sehingga penulis dapat menyelesaikan tugas akhir ini.
2. Bapak Dhomas Hatta Fudholi, S.T., M.Eng., Ph.D. selaku Dosen Pembimbing Akademik (DPA) yang tidak henti-hentinya merespon kebingungan saya saat menempuh masa perkuliahan
3. Ibu Sri Mulyati, S.Kom., M.Kom. selaku dosen pembimbing tugas akhir yang telah membimbing dan mengarahkan penulis dalam menyusun tugas akhir ini.
4. Seluruh Dosen Program Studi Informatika Universitas Islam Indonesia atas ilmu pengetahuan yang telah diberikan sehingga dapat menjadi bekal bagi penulis di kemudian hari.
5. Seluruh staf karyawan Program Studi Informatika Universitas Islam Indonesia yang telah membantu penulis dalam berbagai kepentingan perkuliahan.
6. Teman seperjuangan saya Martina Trikus Rahayu, Auladi Arya Farosi, dan Luthfie Faizal Maulana yang telah menjadi motivasi saya untuk terus maju serta tidak mudah menyerah dalam segala hal.
7. Saudari Shafa Candra Calista yang tidak henti hentinya memberi support secara moral dan emosional dari awal penelitian hingga selesai.
8. Serta masih banyak lagi pihak lainnya yang tidak bisa disebutkan namanya.

Saya menyadari bahwa tugas akhir ini masih jauh dari sempurna. Oleh karena itu, saya sangat mengharapkan kritik, saran, dan masukan yang membangun dari pembaca agar penulisan ilmiah ini dapat lebih baik lagi di masa yang akan datang.

Akhir kata, saya berharap penulisan skripsi ini dapat memberikan manfaat dan kontribusi yang positif bagi perkembangan ilmu pengetahuan di bidang analisis data. Semoga tugas akhir ini dapat membuka wawasan dan memberikan inspirasi kepada pembaca yang mengkaji topik serupa di masa mendatang. Akhirnya, saya mengakhiri kata pengantar ini dengan memohon maaf jika terdapat kekurangan dan kesalahan di dalamnya. Semoga penulisan ini dapat menjadi pijakan untuk penelitian lebih lanjut di masa yang akan datang.

Wassalamu 'alaikum Wr. Wb

Yogyakarta, 28 Oktober 2024

A handwritten signature in black ink, consisting of stylized initials followed by a horizontal line.

(Akhmad Ramadhan)

SARI

Kualitas data merupakan aspek yang sangat penting bagi perkembangan suatu desa. Namun, keakuratan data desa sering kali menghadapi berbagai kendala. Hal ini disebabkan oleh banyaknya lembaga independen yang melakukan pengumpulan data dengan waktu dan metode yang berbeda-beda. Di sisi lain pemerintah desa telah menyediakan data mengenai desa, namun akses terhadap data tersebut terhambat karena sistem penyimpanannya masih dilakukan secara lokal. Berdasarkan permasalahan tersebut, peneliti mengembangkan basis data monografi desa yang terpadu dan dapat terintegrasi dengan berbagai platform. Dengan penerapan sistem ini, pemerintah dapat menyediakan layanan yang lebih responsif, efisien, dan berkualitas tinggi.

Sistem basis data ini memanfaatkan REST API sebagai metode pertukaran data sehingga dapat diakses oleh berbagai platform. Dengan penggunaan REST API, pengelolaan data menjadi lebih efisien, cepat, dan fleksibel, memungkinkan pengelolaan serta akses informasi secara menyeluruh dan terpadu. Untuk menjaga keamanan basis data dari akses pihak yang tidak bertanggung jawab, sistem ini menerapkan JSON Web Token (JWT) sebagai mekanisme keamanan. JWT berfungsi sebagai alat autentifikasi dan otorisasi, memastikan bahwa hanya pengguna dengan hak akses yang sah dapat mengakses dan mengelola data.

Dalam pengembangan ini, peneliti menerapkan pendekatan metode *prototype* sebagai metode pengembangan sistem. Metode ini melibatkan beberapa tahap, yaitu komunikasi, analisis, desain, implementasi, dan pengujian. Pada tahap implementasi desain, peneliti memanfaatkan framework NodeJS dan ExpressJS di sisi server API. Sementara itu, di sisi backend, peneliti menggunakan ORM Sequelize sebagai alat pengelolaan data dan MySQL sebagai server basis data. Hasil dari penelitian dan pengujian menunjukkan bahwa sistem basis data yang dikembangkan mampu terintegrasi dengan berbagai platform secara aman. Dengan adanya sistem ini, pemerintah desa dapat mengelola data desa secara efektif dan efisien. Selain itu, konsistensi serta aksesibilitas data desa juga dapat terjaga dengan baik.

Kata kunci: Basis data, JSON Web Token, Monografi Desa, REST API

GLOSARIUM

API	Antarmuka yang memungkinkan dua aplikasi atau sistem untuk berkomunikasi dan bertukar data.
ORM	Teknik yang menghubungkan basis data relasional dengan kode pemrograman.
Backend	Bagian dari sistem atau aplikasi yang bekerja di sisi server, menangani logika, pengolahan data, dan komunikasi dengan basis data untuk mendukung fungsi yang terlihat di antarmuka.
Server	Sistem yang menyediakan layanan, sumber daya, atau data kepada komputer lain (klien) melalui jaringan, seperti internet atau intranet.
Mobile	perangkat atau teknologi yang dapat digunakan secara portabel, seperti smartphone dan tablet.
REST	Salah satu arsitektur pada API.
HTTP	Protokol komunikasi yang digunakan untuk mengirim dan menerima data antara klien (seperti browser) dan server di web.
Endpoint	Titik akhir dalam sebuah API yang digunakan sebagai alamat untuk menerima atau mengirim data antara klien dan server.

DAFTAR ISI

HALAMAN JUDUL	i
HALAMAN PENGESAHAN DOSEN PEMBIMBING.....	ii
HALAMAN PENGESAHAN DOSEN PENGUJI.....	iii
HALAMAN PERNYATAAN KEASLIAN TUGAS AKHIR.....	iv
HALAMAN PERSEMBAHAN	v
HALAMAN MOTO	vi
KATA PENGANTAR	vii
SARI	ix
GLOSARIUM.....	x
DAFTAR ISI.....	xi
DAFTAR TABEL.....	xii
DAFTAR GAMBAR.....	xiv
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	4
1.3 Batasan Masalah	4
1.4 Tujuan Penelitian	4
1.5 Manfaat Penelitian	5
1.6 Sistematika Penulisan	5
BAB II LANDASAN TEORI.....	7
2.1 Penelitian Terdahulu	7
2.2 Basis Data	10
2.3 Rest API.....	15
2.4 JavaScript.....	16
2.5 Node JS	18
2.6 Black Box Testing.....	20
BAB III Metodologi.....	21
3.1 Metode penelitian.....	21
3.2 Metode pengembangan sistem	22
3.3 Komunikasi dengan Klien.....	23
3.4 Analisis Kebutuhan	24
3.5 Desain.....	26
BAB IV HASIL DAN PEMBAHASAN	49
4.1 Implementasi	49
4.2 Pengujian.....	66
BAB V PENUTUP	96
5.1 Kesimpulan	96
5.2 Saran.....	97
DAFTAR PUSTAKA	98
LAMPIRAN.....	109

DAFTAR TABEL

Tabel 2.1 Penelitian Terdahulu	7
Tabel 3.1 Masalah Pemerintahan Desa	23
Tabel 3.2 Masalah Pengembang IT	23
Tabel 3.3 Kebutuhan Pengguna	24
Tabel 3.4 Kebutuhan Fungsional	25
Tabel 3.5 Struktur Tabel Iterasi I	27
Tabel 3.6 Desain Endpoint Iterasi I	34
Tabel 3.7 Penambahan Desain Struktur Tabel Iterasi II	42
Tabel 3.8 Pengembangan Desain Endpoint Iterasi II	45
Tabel 4.1 Package proyek	50
Tabel 4.2 Relasi antar tabel	53
Tabel 4.3 Seeder Jenis Kelamin	55
Tabel 4.4 Seeder Pendidikan Terakhir	55
Tabel 4.5 Seeder Agama	56
Tabel 4.6 Seeder SHDK	56
Tabel 4.7 Seeder Status	57
Tabel 4.8 Seeder Pekerjaan	57
Tabel 4.9 Seeder Kondisi sarpras	58
Tabel 4.10 Seeder Jenis Sarpras	58
Tabel 4.11 Seeder Jabatan Lembaga Desa	58
Tabel 4.12 Seeder Jabatan Aparatur Desa	59
Tabel 4.13 Seeder Kategori Kewenangan	59
Tabel 4.14 Seeder Jenis Penghasilan Aparatur	60
Tabel 4.15 Seeder Kategori Peristiwa Bencana	60
Tabel 4.16 Seeder Kategori Peristiwa Keamanan	60
Tabel 4.17 Seeder Kategori Pendapatan Desa	61
Tabel 4.18 Seeder Sumber Pendapatan Desa	61
Tabel 4.19 Seeder Kategori Belanja Desa	62
Tabel 4.20 Seeder Jenis Belanja Desa	62
Tabel 4.21 Seeder Kategori Pembiayaan Desa	63
Tabel 4.22 Seeder Jenis Pembiayaan Desa	63

Tabel 4.23 Seeder Kategori Geografi Desa	63
Tabel 4.24 Seeder Kategori Sumber Daya Alam Desa.....	64
Tabel 4.25 Tabel Pengujian Autentifikasi	66
Tabel 4.26 Tabel Pengujian CRUD	73
Tabel 4.27 Tabel Pengujian MultiPlatform	78
Tabel 4.28 Pengujian Syntax	82
Tabel 4.29 Penguian Boundery Value Analisis	87
Tabel 4.30 Pengujian Equivalent Partitioning	93

DAFTAR GAMBAR

Gambar 2.1 Tahapan Normalisasi.....	12
Gambar 2.2 Simbol Entity dalam Crow's Foot Notation	13
Gambar 2.3 Simbol Atribut dalam Crow's Foot Notation.....	13
Gambar 2.4 Simbol Relasi One to One dalam Crow's Foot Notation.....	14
Gambar 2.5 Simbol Relasi One to Many dalam Crow's Foot Notation	14
Gambar 2.6 Simbol Relasi Many to Many dalam Crow's Foot Notation	14
Gambar 2.7 Struktur JSON Web Token	17
Gambar 2.8 Konsep Struktur JWT.....	18
Gambar 3.1 Metodologi Penelitian	21
Gambar 3.2 Metode Prototype	22
Gambar 3.3 Entity Relationship Diagram Iterasi I	32
Gambar 3.4 Arsitektur Sistem REST API	32
Gambar 3.5 Arsitektur Autentifikasi dan Otorisasi JSON Web Token Iterasi I.....	34
Gambar 3.6 Entity Relationship Diagram Iterasi II	43
Gambar 3.7 Arsitektur Autentifikasi dan Otorisasi JSON Web Token Iterasi II	44
Gambar 4.1 Inisiasi Proyek NodeJs	49
Gambar 4.2 inisialisasi ORM sequelize-cli	49
Gambar 4.3 Konfigurasi ORM ke dalam basis data	50
Gambar 4.4 Generate Model.....	52
Gambar 4.5 Konfigurasi Migrasi	54
Gambar 4.6 Generate Seeder	55
Gambar 4.7 Migrasi Seeder	65
Gambar 4.8 Pengujian Register User 1A01	69
Gambar 4.9 Pengujian Register User 1A02.....	70
Gambar 4.10 Pengujian Login User 1B01	70
Gambar 4.11 Pengujian Login User 1B02.....	71
Gambar 4.12 Pengujian JSON Web Token 1C01.....	71
Gambar 4.13 Pengujian JSON Web Token 1C02.....	72
Gambar 4.14 Pengujian JSON Web Token 1C03.....	72
Gambar 4.15 Pengujian Create Data.....	75
Gambar 4.16 Pengujian Read Data.....	76

Gambar 4.17 Pengujian Update Data.....	77
Gambar 4.18 Pengujian Delete Data.....	77
Gambar 4.19 Hasil Pengujian Platform Situs Web.....	80
Gambar 4.20 Hasil Pengujian Platform Aplikasi Android	81
Gambar 4.21 Pengujian Format JSON Object	84
Gambar 4.22 Pengujian Format JSON Array	85
Gambar 4.23 Pengujian Format XML	86
Gambar 4.24 Pengujian Nilai 15 Digit Integer	88
Gambar 4.25 Pengujian Nilai 16 Digit Unik Terkecil	89
Gambar 4.26 Pengujian Nilai 16 Digit Unik Terbesar	89
Gambar 4.27 Debugging Nilai 16 Digit Unik Terbesar.....	90
Gambar 4.28 Pengujian Nilai 16 Digit Tidak Unik	90
Gambar 4.29 Pengujian Nilai 17 Digit Terkecil	91
Gambar 4.30 Pengujian Nilai Format Tidak Integer	92
Gambar 4.31 Pengujian Partisi Valid.....	94
Gambar 4.32 Pengujian Partisi Tidak Valid	94

BAB I

PENDAHULUAN

1.1 Latar Belakang

Perkembangan teknologi merupakan suatu hal yang tidak bisa dihindari, termasuk dalam lingkup pemerintahan desa (Septima & Zulfa, 2020). Banyak lembaga pemerintahan telah memanfaatkan perkembangan teknologi untuk mengoptimalkan pelayanan, memperkuat kemitraan, dan memaksimalkan efisiensi manajemen. (Warman dkk., 2022). Situasi ini justru menyebabkan munculnya banyak perbedaan data di berbagai platform lembaga pemerintahan di Indonesia (Susniwati & Zamili, 2022). Menurut Ristianto dalam jurnal yang ditulis oleh (Susniwati & Zamili, 2022) perbedaan ini disebabkan oleh perbedaan metode dan waktu pengumpulan data yang diterapkan oleh masing-masing lembaga. Perbedaan data yang terjadi sering kali menyebabkan informasi yang disampaikan kepada masyarakat menjadi tidak jelas. Hal ini dapat berdampak buruk pada perkembangan dan kesejahteraan desa. Berbagai lembaga yang mengumpulkan data dengan cara berbeda menunjukkan adanya pengulangan yang tidak efisien. Proses pengumpulan data yang tidak terkoordinasi ini dapat menyebabkan pekerjaan yang tumpang tindih dan mengurangi efektivitas pengelolaan data.

Pemerintahan desa saat ini telah melakukan pengarsipan terkait kebutuhan sumber data. Namun, ada beberapa lembaga yang belum melakukan upaya terbaik mereka untuk mengelola sumber data, terutama yang disimpan secara elektronik (Febriansyah dkk., 2022). Pengelolaan data yang ada cenderung terpisah dan kurang terorganisir. Selain itu, karena data disimpan secara lokal, hak akses terhadap sumber data tersebut sulit didapatkan. Salah satu penyebab rendahnya partisipasi masyarakat dalam pemerintahan dan pembangunan desa atau kelurahan adalah kesulitan mendapatkan akses ini (Atmojo dkk., 2022). Atmojo juga menyatakan bahwa optimalisasi akses terhadap sumber data yang dimiliki oleh pemerintahan desa dapat menciptakan transparansi dalam penyelenggaraan pemerintahan, yang merupakan salah satu syarat penting untuk mencapai *good governance* (Atmojo dkk., 2022).

Dalam konteks ini, pengembangan basis data yang dapat terintegrasi menjadi sangat penting untuk memastikan konsistensi data dan mencegah terjadinya perbedaan informasi di berbagai platform. Integrasi basis data memungkinkan pemerintah desa menyediakan layanan informasi 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 (Qi & Wang,

2021). 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 (Groppe & Groppe, 2024).

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, dan Berita (Kementerian Komunikasi dan Informasi Republik Indonesia, 2003) . 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 personil, data kewenangan, data keuangan, dan data kelembagaan (*Permendagri No. 13 Tahun 2012 Tentang Monografi Desa Dan Kelurahan*, t.t.). Monografi desa berperan sebagai sumber data untuk menggambarkan keadaan, potensi, dan perkembangan desa secara akurat, komprehensif, dan menyeluruh (Atmojo dkk., 2022). Ketersediaan data monografi desa yang lengkap sesuai dengan Permendagri No. 12 Tahun 2007 memungkinkan data tersebut menjadi dasar dalam pelaksanaan program pembangunan dan pemberdayaan masyarakat desa (Rahmawati & Rasyid, 2022). Dengan mempertimbangkan kelengkapan gambaran mengenai desa, mulai dari sumber daya alam hingga sumber daya manusia, yang tersedia dalam monografi desa, serta kesesuaian dengan kriteria informasi yang ditetapkan oleh Kementerian Komunikasi dan Informatika, dapat disimpulkan bahwa monografi desa memiliki potensi sebagai sumber data yang valid mengenai desa. Oleh karena itu, pengembangan basis data monografi yang terintegrasi sangat diperlukan. Melalui integrasi ini, pengelolaan dan akses terhadap data setiap desa akan menjadi lebih mudah dan konsisten bagi berbagai pihak yang membutuhkannya.

Pengembangan sistem basis data monografi terintegrasi dapat memanfaatkan *API* (*Application Programming Interface*) sebagai komponen pendukung utama dalam proses pengembangan. Dalam fungsinya *API* menyederhanakan proses pengembangan, sehingga memungkinkan pengembang bekerja lebih cepat (Thung dkk., 2023). Dalam penelitiannya, Mazorova menyimpulkan bahwa pemilihan *API* yang sesuai dalam pengembangan basis data terintegrasi dapat secara signifikan meningkatkan kinerja sistem, baik dari segi efektivitas maupun efisiensi. (Mazurova dkk., 2023). Hal tersebut dapat dilihat melalui perbedaan signifikan dalam waktu respons dan tingkat penggunaan CPU selama berlangsungnya proses

pengiriman dan penerimaan data dalam pengelolaan data (M. K. Yusof dkk., 2022). Penggunaan API yang sesuai dalam pengembangan basis data terintegrasi memastikan kelancaran penyaluran data, yang menjadi faktor penting dalam menjaga relevansi dan akurasi informasi monografi desa.

REST API merupakan salasatu contoh *Aplication Programing 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 (Ahmad dkk., 2022). 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 (Ramakrishnan, 2023). 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 (Vega-Gorgojo, 2022). 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 autentifikasi dan otorisasi antara *server* dan pengguna dengan menggunakan JSON Web Token, sehingga keamanan data dapat terjaga secara efektif dan aman. (Nashikhuddin dkk., 2023).

Beberapa peneliti sebelumnya telah melakukan pengembangan basis data menggunakan metode REST API dan autentifikasi JWT. Antara lain adalah penelitian yang berjudul Penerapan JSON Web Token sebagai Strategi Pengamanan Data pada Aplikasi MultiMasjid yang dilakukan oleh (Astowo & Sujarwo, 2023). Penelitian tersebut menunjukkan bahwa penggunaan REST API sebagai alat pertukaran data, serta penerapan JWT sebagai mekanisme keamanan, menghasilkan 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 mengenai pengembangan basis data layanan pemerintahan desa dengan metode REST API juga telah dilakukan oleh (Febriansyah dkk., 2022) dengan judul Perancangan Sistem Informasi E-Arsip Berbasis Website Pada Desa Wateswinangun. Dalam penelitian tersebut, Febriansyah dan rekan-rekan menggunakan basis data yang mengimplementasikan REST API pada aplikasi web pemerintahan desa. Hasil penelitian

menunjukkan bahwa penerapan REST API dalam basis data situs web pemerintah desa mampu 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 autentifikasi 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 autentifikasi dan otorisasi. Dengan demikian, penyebaran informasi dapat dilakukan melalui berbagai platform, sehingga meningkatkan efektivitas penyebaran informasi monografi desa. 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.

1.2 Rumusan Masalah

Berdasarkan latar belakang masalah yang diatas, penulis merumuskan masalah yaitu bagaimana merancang basis data untuk monografi desa yang dapat terintegrasi dengan berbagai platform secara aman melalui autentifikasi dan otorisasi.

1.3 Batasan Masalah

Berdasarkan latar belakang yang telah dijelaskan sebelumnya dan agar penelitian lebih terfokus, terukur, dan efisien. Peneliti menetapkan batasan masalah pada penelitian ini adalah sebagai berikut:

- a. Penelitian ini hanya berfokus dalam pengembangan basis data monografi.
- b. API yang digunakan dalam pengembangan basis data adalah REST API.
- c. Pengamanan basis data menggunakan token autentifikasi JSON Web Token.
- d. Target platform yang mempunyai aksesibilitas adalah android pada mobile dan situs web

1.4 Tujuan Penelitian

Berdasarkan rumusan yang telah dirumuskan, penelitian ini memiliki tujuan antara lain:

- a. Meningkatkan integritas data desa dan meminimalkan redudansi data, sehingga data desa menjadi lebih akurat dan konsisten dalam setiap platform publikasi.
- b. Meningkatkan efektifitas portabilitas data dan pengelolaan data dalam lingkup pemerintahan desa.

1.5 Manfaat Penelitian

Dalam pelaksanaan penelitian ini penulis berharap agar penelitian ini kedepannya akan memberikan manfaat bagi penulis, pemerintahan desa, maupun masyarakat terkait. Adapula manfaatnya antara lain :

- a. Bagi penulis
 1. Penelitian ini diharapkan menjadi tugas akhir yang memuaskan sehingga mampu meluluskan penulis dengan nilai yang memuaskan.
 2. Penelitian ini diharapkan mejadi salasatu media belajar oleh penulis untuk melangkah ke jenjang selanjutnya.
- b. Bagi pemerintah
 1. Memberikan referensi kepada pemerintah dalam hal penyimpanan basis data di lingkungan desa.
 2. Membantu meningkatkan kualitas penyebaran informasi kepada masyarakat di lingkungan desa.
- c. Bagi masyarakat
 1. Mendapatkan akses informasi yang baik dan berkualitas secara struktur dan penyimpanan.

1.6 Sistematika Penulisan

Penyusunan karya tulis ilmiah ini, penulis membuat sistematika penulisan sehingga pembahasan dapat dengan mudah dimengerti oleh pembaca. Adapun sistematika penulisan yang disusun oleh penulis sebagai berikut:

BAB I PENDAHULUAN

Dalam Bab Pendahuluan penulis memaparkan Latar Belakang Masalah, Rumusan Masalah, Batasan Masalah, Tujuan Penelitian, Manfaat Penelitian, dan Sistematika Penulisan.

BAB II TINJAUAN PUSTAKA

Pada bab ini penulis akan memaparkan Kajian Pustaka dari penelitian yang berhubungan sebelumnya. Selain itu penulis juga memaparkan Landasan Teori yang dipakai dalam penelitian ini. Landasan Teori ini mencakupi Basis Data, ERD, Normalisasi, Rest API, Javascript, JSON Web Token, Node JS, dan Black Box Testing

BAB III METODOLOGI PENELITIAN

Pada Bab Metodologi Penelitian, penulis akan menjelaskan secara rinci metodologi yang digunakan dalam penelitian ini. Penjelasan mencakup metodologi pengembangan, mulai

dari proses pengumpulan data hingga perencanaan pemecahan masalah. Dalam bab ini, penulis juga memaparkan analisis kebutuhan fungsional sistem yang didasarkan pada hasil analisis kebutuhan pengguna yang telah ditetapkan sebelumnya. Selanjutnya, penulis merancang desain sistem berdasarkan analisis kebutuhan fungsional tersebut, guna memastikan bahwa sistem yang dirancang dapat memenuhi kebutuhan pengguna secara optimal.

BAB IV HASIL DAN PEMBAHASAN

Pada bab ini, penulis akan menjelaskan langkah-langkah implementasi dari proses perencanaan yang telah dilakukan. Selain itu, penulis juga akan menyajikan hasil yang diperoleh dari proses implementasi tersebut. Penulis akan menguji hasil-hasil tersebut untuk memastikan apakah mereka telah mampu menjawab rumusan masalah yang telah ditetapkan.

BAB V PENUTUP

Dalam Bab ini penulis menuliskan Kesimpulan dan Saran dari penelitian yang telah dilakukan.

BAB II

LANDASAN TEORI

2.1 Penelitian Terdahulu

Penelitian terdahulu merupakan salah satu sumber utama landasan teori dalam penelitian ini. Pengembangan basis data terintegrasi yang memanfaatkan REST API dan *JSON Web Token* sebagai alat telah dilakukan sebelumnya, dan hasilnya memberikan pemahaman yang mendalam bagi penelitian ini. Selain itu, penelitian terdahulu berfungsi sebagai dasar yang kuat untuk merumuskan argumen dalam penelitian serta membantu peneliti mengidentifikasi peluang untuk mengisi kekurangan pengetahuan yang masih ada. Dengan memanfaatkan referensi dari penelitian sebelumnya, peneliti dapat memperluas wawasan dan memastikan relevansi serta keakuratan pendekatan yang digunakan dalam penelitian ini. Pada Tabel 2.1 adalah sumber penelitian terdahulu yang relevan terhadap topik penelitian ini.

Tabel 2.1 Penelitian Terdahulu

No	Judul	Penulis	Persamaan	Perbedaan
1	Penerapan JSON Web Token sebagai Strategi Pengamanan Data pada Aplikasi MultiMasjid	(Astowo & Sujarwo, 2023)	Menggunakan REST API untuk komunikasi antara <i>server</i> dan klien serta JSON Web Token	Mempunyai fokus penelitian, efektifitas keamanan menggunakan JWT, membangun basis data untuk aplikasi masjid
2	Implementasi Mongo Db, Express Js, React Js Dan Node Js (Mern) Pada Pengembangan Aplikasi Formulir, Kuis, Dan Survei Online	(NASUTION, 2022)	Menggunakan metode REST API untuk berkomunikasi antara server dan klien, Pengembangan REST API menggunakan ExpressJS, menggunakan JWT sebagai alat	Penelitian mengimplementasikan REST API dalam pengembangan Aplikasi formulir, kuis, dan survei online, platform basis data yang digunakan adalah MONGO DB

			autentifikasi dan otorisasi	
3	REST API Menggunakan NodeJS pada Aplikasi Transaksi Jasa Elektronik Berbasis Android	(Kurniawan dkk., 2020)	Menggunakan REST API untuk komunikasi antara server dan klien, menggunakan Sequelize sebagai alat pengembangan basis data, menggunakan JSON Web Token sebagai alat autentifikasi dan otorisasi	Penelitian mengimplementasikan REST API dalam pengembangan Aplikasi Transaksi Jasa Elektronik berbasis android
4	Perancangan Sistem Informasi E-Arsip Berbasis Website Pada Desa Wateswinangun	(Febriansyah dkk., 2022)	Merancang dan mengembangkan sistem dalam lingkup pemerintahan desa, dan menggunakan REST API sebagai alat komunikasi antara server dan klient	Peneliti tidak menggunakan JWT dalam autentifikasi dan otorisasi REST API
5	Sistem Informasi Pemerintahan Desa Pada Kantor Desa Tenggir Panji Situbondo Menggunakan Php Dan Mysql	(Waqiah, 2018)	Merancang dan mengembangkan sistem dalam lingkup pemerintahan desa, dan menggunakan MySQL sebagai server basis data	Peneliti tidak menggunakan REST API, dan JWT dalam pengembangan aplikasi

Penelitian pertama dilakukan oleh (Astowo & Sujarwo, 2023) dengan judul Penerapan JSON Web Token sebagai Strategi Pengamanan Data pada Aplikasi MultiMasjid membahas tentang sistem keamanan menggunakan JSON Web Token (JWT) yang diterapkan dalam metode REST API pada Aplikasi MultiMasjid. Penelitian ini bertujuan untuk mengevaluasi efektivitas JWT dalam mengamankan data dan memastikan akses yang tepat bagi pengguna. Hasil yang diperoleh menunjukkan bahwa implementasi JWT memberikan performa yang optimal, terutama dalam hal kecepatan dan respon *server*. Dengan menggunakan JWT, sistem tidak hanya mampu meningkatkan kecepatan pengolahan data, tetapi juga menjamin keamanan akses data, sehingga pengguna dapat mengakses informasi dengan aman sesuai dengan hak akses yang telah ditentukan.

Penelitian kedua dilakukan oleh (NASUTION, 2022) yang berjudul Implementasi Mongo Db, Express Js, React Js Dan Node Js (MERN) Pada Pengembangan Aplikasi Formulir, Kuis, Dan Survei Online. Dalam penelitian ini, Nasution mengembangkan aplikasi menggunakan metode REST API, di mana ExpressJS digunakan sebagai kerangka kerja *backend* dan JWT sebagai mekanisme keamanan data. Hasil penelitian menunjukkan bahwa penggunaan ExpressJS dan NodeJS dapat meningkatkan kinerja *backend* serta mempercepat pemuatan data secara keseluruhan. Selain itu, penerapan *stack* MERN mempermudah pengembang dan meningkatkan efisiensi waktu dalam proses pengkodean.

Penelitian ketiga dilakukan oleh (Kurniawan dkk., 2020) dengan judul REST API Menggunakan NodeJS pada Aplikasi Transaksi Jasa Elektronik Berbasis Android. Penelitian ini membangun sebuah aplikasi transaksi jasa elektronik berbasis android dengan menggunakan teknologi REST API dan NodeJS pada backendnya. Untuk mengamankan pertukaran data penelitian ini menggunakan JSON Web Token sebagai alat autentifikasi. Sedangkan untuk melakukan pengelolaan data Kurniawan menggunakan ORM Sequelize. Hasil dari penelitian ini menunjukkan bahwa aplikasi dapat digunakan oleh pengguna mulai dengan memesan jasa hingga melakukan pembayaran. Kemudian pembayaran akan di konfirmasi oleh admin dan detail diteruskan kepada pengguna melalui email.

Penelitian keempat dilakukan oleh (Febriansyah dkk., 2022) yang berjudul Perancangan Sistem Informasi E-Arsip Berbasis Website Pada Desa Wateswinangun. Dalam penelitian ini, Febriansyah mengembangkan aplikasi berbasis web untuk pengarsipan data di desa. Data yang diarsipkan meliputi arsip surat, arsip informasi desa, arsip inventaris desa, arsip administrasi desa, dan arsip administrasi penduduk. Aplikasi ini menggunakan REST API sebagai sarana

komunikasi antara server dan klien. Hasil penelitian menunjukkan bahwa sistem pengarsipan data desa ini memudahkan perangkat desa dalam mengelola data, serta membuat akses informasi menjadi lebih efektif.

Penelitian kelima dilakukan oleh (Waqiah, 2018) dengan judul Sistem Informasi Pemerintahan Desa pada Kantor Desa Tenggir Panji Situbondo Menggunakan PHP dan MySQL. Dalam penelitian ini, Waqiah mengembangkan sistem informasi untuk pemerintahan desa menggunakan bahasa pemrograman PHP dan MySQL sebagai basis data. Sistem ini menyimpan berbagai data, seperti data penduduk, data pemerintahan desa, data dusun, data rukun tetangga, dan data rukun warga. Hasil penelitian menunjukkan bahwa sistem yang dibangun dengan PHP dan MySQL mampu menyajikan informasi secara lebih cepat dan akurat, serta mempermudah pengelolaan data oleh pemerintahan desa.

Berdasarkan hasil pembelajaran dari penelitian-penelitian sebelumnya, peneliti mengajukan topik baru mengenai implementasi JSON Web Token dalam pengembangan basis data terintegrasi untuk monografi desa dengan menggunakan metode REST API. Pembaruan yang dilakukan dalam penelitian ini bertujuan untuk meningkatkan aspek keamanan pada REST API melalui penerapan JSON Web Token. Dalam pengembangan sistem, peneliti menggunakan NodeJS dan ExpressJS sebagai kerangka kerja server API, serta ORM Sequelize untuk pengelolaan data di sisi server. Pada akhirnya, pembaruan ini diharapkan dapat mendukung tercapainya tata kelola pemerintahan desa yang lebih baik, dengan pengelolaan data yang aman, efisien, dan transparan.

2.2 Basis Data

Menurut (Fathansyah, 2012), basis data dapat dipahami dari berbagai sudut pandang, antara lain:

- a. Sekumpulan data yang saling terkait dan diatur sedemikian rupa sehingga dapat dimanfaatkan kembali dengan cepat dan mudah.
- b. Kumpulan data yang saling terkait dan disimpan bersama-sama dengan cara yang menghindari redundansi yang tidak perlu, sehingga dapat memenuhi berbagai kebutuhan.
- c. Sekelompok file atau tabel yang saling terkait dan disimpan dalam media penyimpanan elektronik.

Seiring perkembangan zaman basis data yang semula adalah *paper based database* berkembang menjadi *computer based database*, yang akhirnya hingga kini berkembang menjadi *web based database* (WBDB). WBDB merupakan sekumpulan data rangkaian halaman web yang terorganisir dengan baik, menyediakan akses yang mudah dan sistematis

bagi pengguna (Nicholson, S., 2002). Sedangkan *Database Management System* (DBMS) adalah kumpulan program untuk mendefinisikan, mengatur, dan memproses basis data (Siswanti, Sri, 2021) .

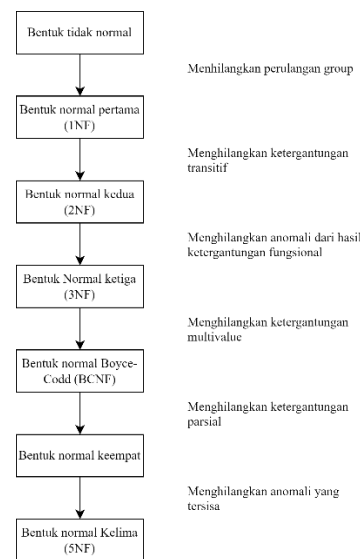
Menurut Nurmawati sebuah basis data akan sangat berpengaruh terhadap sebuah aplikasi, makin rumit aplikasi dibangun maka aplikasi tersebut makin memerlukan pemodelan basis data yang kompleks (Nurnawati & Ermawati, 2018). Dalam situs web, basis data digunakan untuk menyimpan data maupun informasi yang akan ditampilkan untuk pengguna website (Darnis & Azdy, 2018). Oleh karena itu pengembangan basis data dianjurkan untuk seoptimal mungkin. Basis data yang optimal dapat meringankan beban kerja hardware dan mampu memberikan hasil yang maksimal. Basis data yang optimal dapat terbentuk ketika data base di design dengan benar, selain itu dapat juga dilakukan normalisasi database dengan tujuan basis data yang optimal (Dwita Mentari, 2014).

2.2.1 Normalisasi

Pada perancangan basis data terdapat tahapan normalisasi yang digunakan dalam mengurangi ambiguitas, anomali, dan inkonsistensi data. Tahap normalisasi dapat menunjang kinerja basis data dan mengurangi kesalahan jika mendapat perintah melalui protokol khususnya *UPDATE*, *INSERT*, dan *DELETE* (Puspitasari dkk., 2016). Normalisasi adalah cara untuk menyusun atribut-atribut dari sebuah tabel dalam basis data relasional agar tidak ada redundansi, sehingga strukturnya menjadi lebih efisien (Fathansyah, 2012). Menurut Puspitasari proses normalisasi melibatkan beberapa langkah. Pertama, data diorganisir dalam bentuk tabel dan kemudian dianalisis berdasarkan persyaratan tertentu ke beberapa tingkat. Selanjutnya, jika tabel yang dianalisis belum memenuhi persyaratan tersebut, maka tabel tersebut perlu dipecah menjadi beberapa tabel yang lebih sederhana hingga memenuhi bentuk yang optimal. Normalisasi melibatkan enam tahapan seperti pada Gambar 2.6, di mana setiap tahapan memiliki sejumlah aturan yang harus dipatuhi (Elmasri & Navathe, 2011). Berikut aturan yang normalisasi yang dikemukakan (Puspitasari dkk., 2016), sebuah tabel akan dikatakan efisien apabila:

- a. Apabila terjadi dekomposisi (penguraian) tabel, dekomposisi tersebut harus dipastikan aman (Lossless-Join Decomposition). Ini berarti, setelah tabel tersebut diuraikan menjadi tabel-tabel baru, tabel-tabel baru tersebut dapat digabungkan kembali untuk menghasilkan tabel semula dengan persis sama.
- b. Pentingnya mempertahankan ketergantungan fungsional saat terjadi perubahan data (Dependency Preservation).

- c. Memastikan bahwa struktur basis data tidak melanggar Bentuk Normal Boyce-Codd (BCNF).



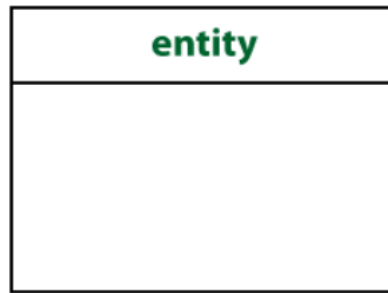
Gambar 2.1 Tahapan Normalisasi

Meskipun pada Gambar 2.1 memperlihatkan terdapat enam tahap dalam normalisasi namun pada praktik didalam industri hanya menggunakan sampai tahap BCNF atau paling banyak 4NF. Hal itu sama seperti yang dikatakan oleh Elmasri dalam bukunya, dikarenakan bentuk 4NF dan 5NF ini sulit dipahami dan jarang terjadi kasus yang menjadi kendala permasalahanya (Elmasri & Navathe, 2011).

2.2.2 ERD

Menurut (Brady & Loonam, 2010) *Entity Relationship Diagram* (ERD) adalah sebuah teknik yang digunakan untuk memodelkan kebutuhan data dalam suatu perusahaan atau organisasi. Menurut Pulungan ERD merupakan diagram yang berbentuk notasi grafis yang berada dalam pembuatan basisdata yang menghubungkan antara data satu dengan data lainnya. Selain itu Pulungan menjelaskan, di dalam ERD terdapat tiga elemen dasar yaitu entitas, atribut, dan relasi (Pulungan dkk., 2023).

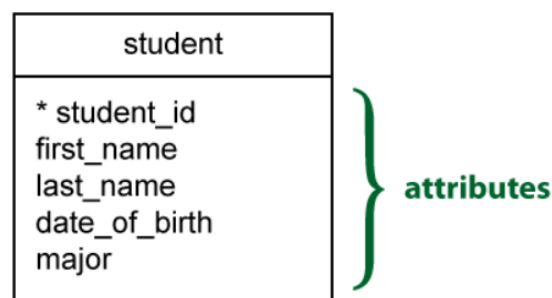
a. Entitas



Gambar 2.2 Simbol Entity dalam Crow's Foot Notation
(*Crow's Foot Notation*, 2016)

Menurut (Fathansyah, 2012) Entitas adalah suatu penjelasan yang dapat digambarkan oleh suatu data, yang dapat diartikan sebagai kata yang secara individu yang akan mewakili sesuatu yang nyata yang mewakili sesuatu yang lainnya. Dalam suatu entitas harus berisi satu atribut yang unik dan digunakan untuk mengidentifikasi entitas tersebut (Pulungan dkk., 2023). Atribut ini unik ini disebut *primary key*. *Primary key* ini berfungsi untuk memastikan bahwa setiap entitas dalam sebuah basis data dapat dikenali secara unik, sehingga tidak terjadi duplikasi data dan setiap entitas dapat diakses atau diubah dengan benar. Entitas ditunjukkan pada Gambar 2.2

b. Atribut



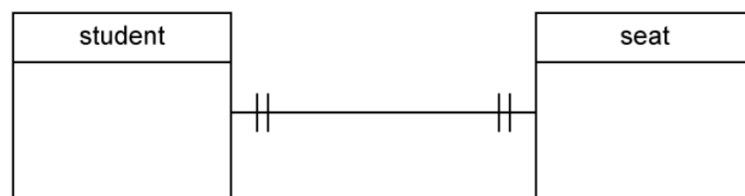
Gambar 2.3 Simbol Atribut dalam Crow's Foot Notation
(*Crow's Foot Notation*, 2016)

Atribut adalah informasi yang dimiliki oleh entitas (Pulungan dkk., 2023). Setiap entitas terdiri dari sejumlah atribut yang mendefinisikan karakteristik atau detail spesifik dari entitas tersebut. Misalnya pada Gambar 2.3 terapat entitas student, entitas tersebut dapat memiliki atribut *student_id*, *first_name*, *last_name*, *date_of_birth*, dan *major* . Atribut-

atribut ini membantu dalam mengidentifikasi, mengklasifikasikan, dan mengelola data terkait entitas tersebut.

c. Relasi

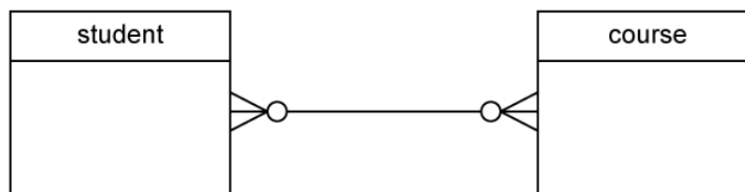
Menurut (Storey, 1991) relasi adalah asosiasi dari setiap entitas. Terdapat tiga jenis relasi yang sering digunakana yaitu *one-to-one* (1:1), *one-to-many* (1:M), dan *many to many* (N:M) (Harrington, 2016). Relasi *one-to-one* seperi pada Gambar 2.4 menghubungkan satu himpunan dalam entitas dengan tepat satu himpunan dalam entitas lainnya. Relasi *one-to-many* seperti pada Gambar 2.5 menghubungkan satu himpunan dalam entitas dengan banyak himpunan dalam entitas lainnya. Sedangkan relasi *many-to-many* seperti pada Gambar 2.6 menghubungkan banyak himpunan dalam entitas dengan banyak himpunan dalam entitas lainnya, memungkinkan hubungan yang lebih kompleks dan dinamis dalam basis data.



Gambar 2.4 Simbol Relasi One to One dalam Crow's Foot Notation



Gambar 2.5 Simbol Relasi One to Many dalam Crow's Foot Notation



Gambar 2.6 Simbol Relasi Many to Many dalam Crow's Foot Notation

(*Crow's Foot Notation*, 2016)

2.3 Rest API

REST API (Representational State Transfer Application Programming Interface) adalah sebuah pendekatan arsitektur yang digunakan dalam pengembangan aplikasi web untuk berkomunikasi antara *client* dan *server*. REST API berkomunikasi melalui protokol HTTP yang menjadikan API ini tidak terikat dengan protokol lain. Hal tersebut selaras dengan tiga prinsip utama REST API (Prayogi dkk., 2020)

Menurut (Rodriguez dkk., 2016) REST API mampu mengelola beragam jenis sumber daya, seperti dokumen untuk sebuah entitas tunggal, koleksi untuk kumpulan sumber daya, dan pengontrol untuk tindakan yang tidak dapat langsung dipetakan ke metode HTTP standar. Meskipun pemodelan sumber daya untuk REST API pada dasarnya mirip dengan kelas pemodelan dalam pemrograman berorientasi objek (OO) atau entitas dalam pemodelan data, terdapat praktik penamaan yang direkomendasikan yang khas untuk REST API. Misalnya, menggunakan kata benda tunggal untuk merepresentasikan dokumen, kata benda jamak untuk koleksi, dan kata kerja hanya untuk pengontrol (Masse, 2011). Selain itu, tidak disarankan untuk menggunakan nama operasi CRUD di dalam URL

Dalam konteks pengembangan database, REST API memiliki beberapa fungsi penting menurut (Li et al., 2016; Li & Chou, 2011; Sohan et al 2017):

- a. Memungkinkan aplikasi untuk mengakses dan mengelola data dalam database Dengan menggunakan API endpoint yang sesuai, pengembang dapat membaca, menulis, mengupdate, dan menghapus data dalam database.
- b. Memungkinkan aplikasi yang berbeda, baik itu di platform yang sama atau berbeda, untuk berkomunikasi satu sama lain. Sebuah aplikasi dapat mengirim permintaan HTTP ke REST API untuk meminta atau mengirim data ke aplikasi lain, termasuk database.
- c. Mempermudah integrasi dengan berbagai aplikasi dan platform. Dalam konteks database, ini berarti bahwa aplikasi dapat terhubung dengan berbagai jenis database, termasuk basis data SQL (seperti MySQL, PostgreSQL) dan NoSQL (seperti MongoDB, Couchbase).
- d. REST API memungkinkan pengembang untuk mengimplementasikan lapisan keamanan, seperti otentikasi dan otorisasi, untuk melindungi data dalam database. Penggunaan token API, HTTPS, dan teknik keamanan lainnya dapat memastikan bahwa hanya pengguna yang sah yang memiliki akses ke data sensitif.
- e. REST API memungkinkan pengembangan aplikasi yang dapat diskalakan dengan mudah. Dengan menggunakan API, aplikasi dapat memisahkan antara antarmuka pengguna dan

logika bisnis, memungkinkan pengembangan tim bekerja secara independen pada berbagai bagian dari aplikasi.

2.4 JavaScript

JavaScript adalah bahasa yang berbentuk kumpulan kode yang pada fungsinya dapat berjalan pada suatu dokumen HTML (Brew, 2003). JavaScript adalah bahasa yang sangat kuat, dan dapat digunakan untuk menambahkan fitur-fitur canggih guna meningkatkan pengalaman pengguna dalam pengalaman menggunakan website (Suehring & Valade, 2013). Melalui JavaScript, pengembang dapat menciptakan interaksi yang lebih dinamis dan responsif, sehingga membuat website menjadi lebih menarik dan fungsional.

Sintaks JavaScript berasal dari bahasa Java, mendukung fungsi kelas satu seperti Skema, dan menggunakan pewarisan berbasis prototipe (Flanagan, 2012). JavaScript adalah bahasa yang bersifat "case sensitive", yang artinya dapat membedakan penamaan variabel dan fungsi berdasarkan penggunaan huruf besar dan kecil (Brew, 2003). Sebagai contoh, variabel atau fungsi dengan nama "CONTOH" berbeda dari variabel atau fungsi dengan nama "contoh". Seperti halnya dalam bahasa Java dan C, setiap instruksi di JavaScript diakhiri dengan tanda titik koma (;).

2.4.1 JSON

JSON (Java Script Object Nation) adalah format data yang lebih ringan yang didasarkan pada tipe bahasa pemrograman JavaScript (Pezoa dkk., 2016). Format ini terdiri dari pasangan kunci dan nilai, di mana nilai-nilai tersebut dapat dikonversi kembali menjadi format JSON (Bourhis dkk., 2017). Dalam pengelolaan data dalam skala besar, JSON menawarkan fleksibilitas dan mampu menangani throughput tinggi serta latensi rendah tanpa mengorbankan skalabilitas (K. Yusof & Man, 2017). JSON mendukung berbagai platform pemrograman, yang menjadikannya sangat efisien (Pezoa dkk., 2016)

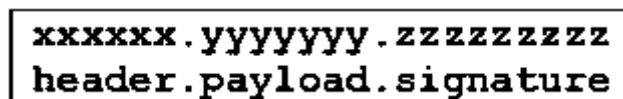
Menurut Isnanda, JSON mempunyai dua struktur universal dalam format yang disediakan (Isnanda, t.t.). Hal itu di dasarkan kepada dukungan JSON terhadap semua bahasa pemrograman moderen dalam bentuk yang sama maupun berlainan. Dua struktur universal menurut (Isnanda, t.t.) antara lain:

- a. Kumpulan pasangan nama/nilai. Dalam beberapa bahasa pemrograman, ini dikenal sebagai objek, rekaman, struktur, kamus, tabel hash, daftar berkunci, atau associative array.
- b. Deretan nilai yang teratur. Dalam banyak bahasa pemrograman, konsep ini dikenal sebagai larik (array), vektor (vector), daftar (list), atau urutan (sequence).

2.4.2 JSON Web Token

JSON Web Token (JWT) merupakan sebuah token dengan format data string JSON yang sangat ringkas dan mempunyai informasi independen (Astowo & Sujarwo, 2023). Meskipun terbentuk dari format data JSON string, JWT tidak bergantung pada bahasa pemrograman tertentu (Painem & Soetanto, 2020). Informasi dalam JWT digunakan untuk autentifikasi dan pertukaran informasi antar sistem (Kelvin, 2019). Saat melakukan proses autentifikasi arsitektur JWT mempunyai algoritma yang mendukung enkripsi token yaitu HMAC SHA-512 (Setiawan & Purnamasari, 2020). Purnamasari juga mengatakan, algoritma HMAC SHA-512 merupakan algoritma enkripsi yang efisien dan unggul. Algoritma dapat ini mendukung kinerja REST API dengan cara meningkatkan kinerja kecepatan dan ukuran data pada keamanan REST API (Astowo & Sujarwo, 2023)

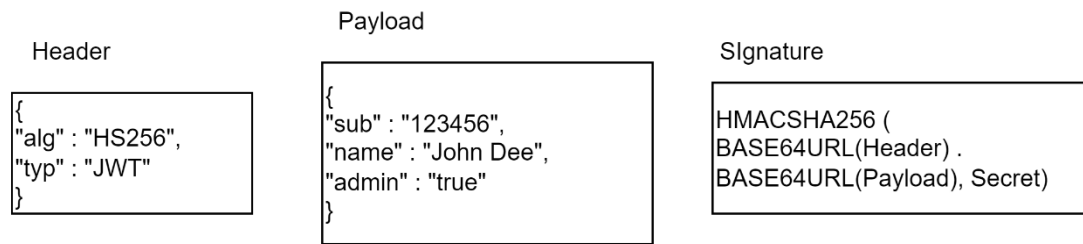
Menurut Kelvin, JSON Web Token terdiri dari tiga bagian yang dipisahkan oleh titik. Seperti yang di Gambar 2.7 JWT mempunyai struktur *header*, *payload*, dan *signature* (Kelvin, 2019)



Gambar 2.7 Struktur JSON Web Token

(Kelvin, 2019)

Header terdiri dari dua bagian, yaitu tipe token (JWT) dan algoritma hashing yang digunakan seperti HMAC SHA, RSA, atau yang lainnya (Painem & Soetanto, 2020). Payload berisi klaim yang menyatakan suatu entitas dan metadata lainnya (Kelvin, 2019). Signature berisi hasil hashing dari *header*, *payload*, dan kunci sandi (Kelvin, 2019). Pada Gambar 2.8 adalah contoh *Header*, *Payload*, dan *Signature* JSON Web Token. Cara Kerja JSON Web Token sama seperti kata sandi, ketika user berhasil login maka *server* akan memberikan token yang akan disimpan dalam *localstorage* atau di *cookies* browser (Gunawan & Rahmatulloh, 2019).



Gambar 2.8 Konsep Struktur JWT

(Astowo & Sujarwo, 2023)

2.5 Node JS

Dalam penelitian yang dilakukan oleh (Haryadi dkk., 2023) berjudul *Perbandingan REST API NodeJS dan PHP pada Aplikasi Pemilihan Umum*, ia membahas perbedaan antara REST API berbasis NodeJS dan PHP. Penelitian ini menguji aspek kecepatan respon, skalabilitas, dan kemudahan pengembangan dengan menggunakan alat Postman. Hasil penelitian menunjukkan bahwa NodeJS memiliki ekosistem dan modul yang lebih beragam, fleksibel, serta menawarkan kecepatan respon yang lebih stabil dan konsisten dibandingkan PHP, karena NodeJS memungkinkan arsitektur yang lebih bebas.

Node JS, yang kadang disebut sebagai Node, adalah lingkungan runtime JavaScript yang dirancang dengan karakteristik ringan dan mampu menangani permintaan berbasis peristiwa serta non-blokir pada sisi server JavaScript (Shah & Soomro, 2017). Bahasa pemrograman JavaScript biasa dikenal sebagai bahasa pemrograman yang berjalan disisi client, sementara NodeJS ada sebagai pelengkap JavaScript disisi server (Kurniawan dkk., 2020). NodeJS memiliki beberapa framework seperti ExpressJS, NestJS, KoaJS, dan lain sebagainya (Muhammad & Paputungan, 2024). Berdasarkan survei dari situs stackoverflow tahun 2023 ExpressJS dan NestJS merupakan framework NodeJS yang paling banyak digunakan pengembang.

Penggunaan framework ExpressJS dalam pengembangan REST API memberikan sejumlah manfaat. ExpressJS merupakan framework NodeJS yang minimalis dan ringan (*Express - Node.js Web Application Framework*, t.t.). Aplikasi yang dikembangkan menggunakan ExpressJS mampu menangani permintaan HTTP hingga 249% lebih efisien dibandingkan dengan framework lainnya (Wicha & Pańczyk, 2023). Selain itu, penggunaan ExpressJS dalam pengembangan REST API cukup umum diterapkan (Azkarin dkk., 2023).

Penggunaan ExpressJS dalam pengembangan REST API kerap diterapkan (Mardan, 2014), sehingga dukungan komunitas dapat membantu dalam pengembangan sistem. Dengan berbagai keunggulan tersebut, dapat disimpulkan bahwa pengembangan REST API dengan ExpressJS memberikan efisiensi dalam respon server, mendukung skalabilitas API, serta memudahkan pengembangan sistem berkat dukungan komunitas yang luas.

2.5.1 Sequelize

Object-Relational Mapping (ORM) adalah sebuah mekanisme yang digunakan untuk menghubungkan kelas dalam bahasa pemrograman berorientasi objek (OO) dengan tabel pada sistem manajemen basis data relasional (RDBMS) (Lorenz dkk., 2024). Penggunaan ORM mempermudah pengembang dalam bekerja dengan basis data relasional, mengurangi kebutuhan untuk menulis query SQL secara manual, serta memungkinkan akses data yang lebih efisien (Juneau & Juneau, 2013). Dalam penelitian Armas, disebutkan bahwa penggunaan ORM dapat mengurangi waktu respons dan jumlah baris kode dalam pengembangan basis data, sekaligus mengoptimalkan pengelolaan basis data dengan peningkatan efisiensi hingga sepuluh kali lebih cepat (Armas dkk., 2017).

Sequelize adalah alat Object-Relational Mapping (ORM) berbasis *promise* yang digunakan dalam Node.js (*Sequelize v6 | Sequelize*, 2024). Di situs web resminya, Sequelize menyatakan bahwa alat ini mendukung berbagai jenis basis data, termasuk Postgres, MySQL, MariaDB, SQLite, Microsoft SQL Server, Oracle Database, Amazon Redshift, dan Snowflake's Data Cloud. Sequelize memiliki dukungan yang baik untuk transaksi, manajemen relasi antar tabel, serta fitur *eager loading* dan *lazy loading* (pemanggilan data secara cepat atau bertahap sesuai kebutuhan), replikasi baca, dan banyak fitur lainnya (*Sequelize v6 | Sequelize*, 2024).

Dalam pengembangan basis data sequelize menyediakan antarmuka berorientasi objek yang memetakan panggilan metode di JavaScript ke dalam query basis data (Turcotte dkk., 2023). Dengan demikian, pengembang tidak perlu menulis query secara langsung dalam SQL, melainkan menggunakan metode JavaScript yang secara otomatis diterjemahkan menjadi query SQL. Sequelize mendukung berbagai operasi basis data, seperti pengelompokan, *join* antar tabel, dan fungsi agregasi seperti COUNT dalam query SQL, sehingga memudahkan pengembang melakukan operasi yang kompleks dengan cara yang lebih sederhana dan terstruktur (Turcotte dkk., 2023).

Penggunaan ORM Sequelize dalam pengembangan REST API

2.6 Black Box Testing

Black Box Testing (BBT) merupakan metode pengujian di mana fungsionalitas dari sebuah perangkat yang diuji (Device Under Test/DUT) diperiksa tanpa pengetahuan tentang struktur internal perangkat tersebut, seperti perangkat keras, perangkat lunak, dan hubungan di antara keduanya (Kaprocki dkk., 2015). Pengujian dengan kotak hitam hanya dapat menguji sistem secara fungsional, pengujian black box tidak dapat menguji model dan kode program untuk mengetahui fungsi input dan output sistem (Nugraha dkk., 2022).

Menurut (Jaya, 2018) metode black box testing mempunyai keuntungan diantaranya:

- a. Tidak Memerlukan Pengetahuan Pemrograman. Penguji tidak perlu memiliki pengetahuan tentang bahasa pemrograman tertentu. Hal ini memungkinkan pengujian dilakukan oleh individu yang tidak memiliki latar belakang teknis mendalam.
- b. Sudut Pandang Pengguna. Pengujian dilakukan dari sudut pandang pengguna, yang membantu mengungkap ambiguitas atau inkonsistensi dalam spesifikasi persyaratan. Dengan melihat aplikasi seperti yang akan dilihat oleh pengguna akhir, penguji dapat mengidentifikasi masalah yang mungkin tidak terlihat oleh pengembang.

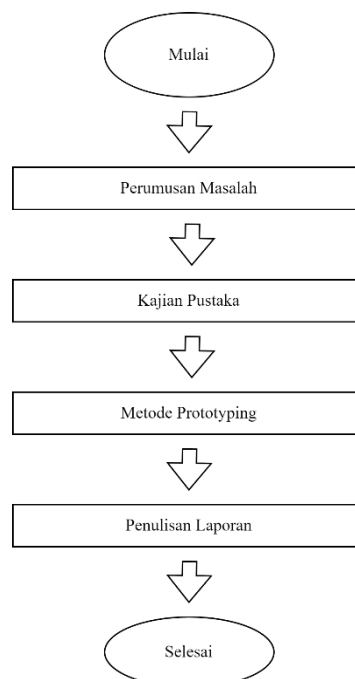
Programmer dan tester saling bergantung satu sama lain dalam konteks kerjasama untuk memastikan kualitas aplikasi. Meskipun metode Black Box Testing memungkinkan tester untuk menguji aplikasi tanpa memerlukan pengetahuan mendalam tentang kode yang ditulis oleh programmer, komunikasi dan kolaborasi antara kedua pihak tetap penting.

BAB III

Metodologi

3.1 Metode penelitian

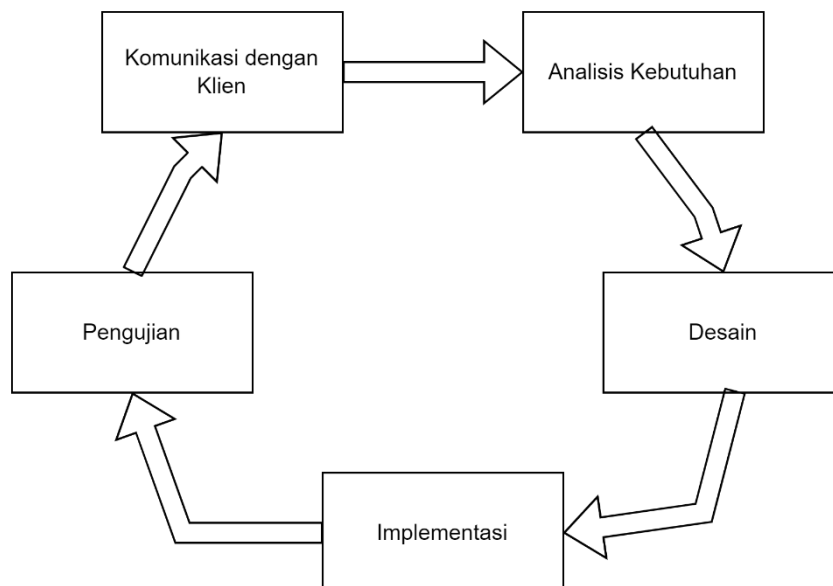
Dalam pelaksanaan penelitian ini, peneliti menggunakan metode yang tertera pada Gambar 3.1. Metode penelitian dibagi menjadi empat bagian, perumusan masalah, kajian literatur, metode *prototyping*, dan penulisan laporan. Tahap perumusan masalah dilakukan untuk merumuskan dan membatasi penelitian yang akan dilakukan, sehingga penelitian akan menjadi terarah dan tidak keluar dari batasan masalah yang ditetapkan (Ardiansyah dkk., 2021). Tahap kajian pustaka dilakukan dengan menggali informasi terkait segala kebutuhan pengembangan sistem yang akan dilakukan. Dalam tahap ini informasi yang didapat dapat berupa teori mengenai pengembangan sistem maupun batasan atau aturan yang telah dibuat pemerintah mengenai objek yang akan dikembangkan. Untuk pengembangan sistem yang digunakan adalah pendekatan metode *prototyping*, metode ini diawali dengan komunikasi dengan klien, menganalisis kebutuhan, merancang sistem, melakukan implementasi, dan berakhir dengan menguji sistem yang sudah dibuat. Penulisan laporan berguna sebagai dokumentasi seluruh tahapan penelitian, dan sebagai informasi bagaimana cara membuat basis data terpadu monografi desa yang dapat terintegrasi di berbagai platform



Gambar 3.1 Metodologi Penelitian

3.2 Metode pengembangan sistem

Metode pengembangan sistem yang digunakan adalah metode prototype. Menurut Pressman di dalam jurnal yang ditulis oleh Wicasana dan Djutalov, metode prototipe adalah metode pengembangan perangkat lunak yang memungkinkan adanya komunikasi antara pengembang dan pengguna sistem secara terus menerus. Dengan demikian, metode ini dapat meminimalisir ketidaksesuaian antara pengembang dan pengguna (Wicasana & Djutalov, 2023). Penggunaan metode ini memberikan keuntungan bagi pengembang dan pengguna, karena memungkinkan komunikasi berkelanjutan antara keduanya selama proses pengembangan (Ichwani dkk., 2021), sehingga pengembang mendapatkan respon pengguna untuk memperbaiki sistem yang dikembangkan.



Gambar 3.2 Metode Prototype

Tahap-tahap yang digunakan dalam pengembangan yang digunakan dalam metode prototype ditunjukkan pada Gambar 3.2. Proses pengembangan perangkat lunak dimulai dengan komunikasi dengan pengguna, yang bertujuan untuk mengidentifikasi berbagai kebutuhan pengguna. Setelah kebutuhan tersebut diidentifikasi, tahap kedua adalah menganalisis kebutuhan tersebut secara mendalam untuk memahami spesifikasi yang diinginkan oleh pengguna. Tahap ketiga adalah perancangan desain, yang disusun berdasarkan analisis kebutuhan pengguna yang telah dilakukan sebelumnya. Desain ini kemudian dievaluasi oleh pengguna untuk mendapatkan masukan yang relevan, sehingga sistem yang dikembangkan nantinya sesuai dengan kebutuhan pengguna. Setelah itu, desain tersebut diimplementasikan. Setelah implementasi selesai, langkah berikutnya adalah melakukan pengujian sistem untuk

memastikan bahwa sistem berfungsi dengan baik dan memenuhi seluruh persyaratan yang telah ditetapkan.

3.3 Komunikasi dengan Klien

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.

Dari hasil diskusi yang dilakukan dengan stakeholder terkait yaitu pemerintahan desa dan pengembang IT, dirumuskan masalah apa saja yang berkaitan dengan pengembangan basis data monografi desa. Pada Tabel 3.1 tertera masalah yang dialami oleh pemerintahan desa sedangkan pada Tabel 3.2 tertera masalah yang dialami oleh pengembang IT.

Tabel 3.1 Masalah Pemerintahan Desa

No	Masalah
1	Informasi mengenai monografi desa cenderung tidak konsisten.
2	Pemerintahan desa tidak mempunyai dokumentasi digital mengenai monografi desa.
3	Desa menerima permintaan data monografi yang terus berulang untuk kebutuhan pihak yang sedang bekerja sama.
4	Pencatatan data desa dilakukan setiap periode waktu tertentu sehingga memerlukan waktu yang lama dalam melakukan resume data

Tabel 3.2 Masalah Pengembang IT

Mo	Masalah
1	Basis data yang tidak fleksibel tidak dapat memenuhi kebutuhan pengembangan sistem yang akan datang.
2	Data desa dapat berubah setiap waktu.

3	Keamanan yang tepat untuk melindungi data desa.
4	Standar pengelolaan data

3.4 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. Sementara itu, analisis kebutuhan fungsional sistem berfokus pada identifikasi kebutuhan fungsional dari sistem yang akan dikembangkan. Melalui pendekatan ini, peneliti berharap dapat merancang solusi yang tepat guna dan sesuai dengan kebutuhan yang diidentifikasi.

3.4.1 Analisis Kebutuhan Pengguna

Tahap ini merupakan pengembangan lebih lanjut dari masalah yang telah dirumuskan sebelumnya. Pada tahap ini, dilakukan analisis kebutuhan yang didasarkan pada tabel masalah yang telah dirumuskan sebelumnya. Hasil analisis kebutuhan pengguna dapat dilihat pada Tabel 3.3

Tabel 3.3 Kebutuhan Pengguna

No	Pengguna	Kebutuhan
1.	Pemerintahan Desa	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. 4. Pemerintah mendapatkan resume data secara otomatis
2.	Pengembang IT	1. Pengembang memerlukan sebuah basis data relational yang scalable .

		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.
--	--	--

3.4.2 Analisis Kebutuhan Fungsional

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 ditunjukkan pada Tabel 3.4

Tabel 3.4 Kebutuhan Fungsional

No	Pengguna	Kebutuhan
1.	Pemerintahan Desa	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.
2.	Pengembang IT	1. Sistem menggunakan basis data yang relasional

		2. Sistem menyediakan layanan register dan login pengguna yang berguna untuk mendapatkan token autentifikasi. 3. Sistem menggunakan format JSON dalam penyajian dan penginputan data. 4. Sistem menyediakan JSON Web Token sebagai alat autentifikasi dan otorisasi dalam proses pengolahan data. 5. Sistem menyediakan akses untuk memasukkan data kedalam basis data. 6. Sistem menyediakan akses untuk mengubah data yang berada dalam basis data. 7. Sistem menyediakan akses untuk menghapus data yang berada dalam basis data. 8. Sistem menyediakan akses untuk melihat seluruh data dalam tabel maupun detail data dalam basis data.
--	--	--

3.5 Desain

Dalam tahap ini, peneliti mengembangkan kebutuhan yang telah dianalisis pada Subbab 3.4 menjadi sebuah desain sistem yang akan dibangun. Tahap ini dibagi menjadi beberapa bagian utama. Pertama, peneliti akan membuat desain struktur data yang mencakup definisi dan organisasi data yang akan digunakan dalam sistem, memastikan bahwa data tersebut terstruktur secara efisien dan mudah diakses. Kedua, desain Entity-Relationship Diagram (ERD) akan dikembangkan untuk memvisualisasikan hubungan antar entitas dalam basis data, membantu dalam pemodelan data yang lebih jelas dan terorganisir. Terakhir, peneliti akan merancang arsitektur sistem REST API yang menggunakan JSON Web Tokens (JWT) untuk otentikasi dan otorisasi, memastikan bahwa komunikasi antar komponen sistem aman dan terjamin.

Dalam keseluruhan proses desain dalam penelitian ini dilakukan dua iterasi, di mana setiap iterasi akan memberikan kesempatan untuk melakukan penyesuaian dan penyempurnaan,

sehingga hasil akhir dapat benar-benar memenuhi ekspektasi dan spesifikasi pengguna. Setiap iterasi dalam proses pengembangan ini akan melalui tahap evaluasi secara menyeluruh untuk memastikan bahwa hasil yang dicapai sudah sesuai dengan kebutuhan dan tujuan yang ditetapkan. Evaluasi dilakukan pada akhir setiap iterasi untuk mengidentifikasi kekurangan, mendapatkan umpan balik, dan menentukan perbaikan yang diperlukan sebelum melanjutkan ke tahap berikutnya.

3.5.1 Iterasi I

3.5.1.1 Desain Struktur Tabel

Desain struktur tabel berfungsi menggambarkan seluruh atribut yang akan disimpan didalam tabel-tabel basis data. Desain struktur tabel ini berdasarkan aturan pengisian data monografi yang ditetapkan menteri dalam negeri. Menurut Peraturan Menteri Dalam Negeri Republik Indonesia No. 13 Tahun 2012 tentang Monografi Desa dan Kelurahan, monografi mencakup berbagai jenis data yang jelaskan di Tabel 3.5, yaitu: data umum, data personil, data kewenangan, data keuangan, dan data kelembagaan (*Permendagri No. 13 Tahun 2012 Tentang Monografi Desa Dan Kelurahan*, t.t.). Data umum terbagi menjadi data sosial (data kependudukan) pada nomor dua data ketentraman dan ketertiban pada nomor enam, data bencana pada nomor lima, dan data Geografi desa pada nomor tiga. Data personil melingkupi data penyelenggara pemerintahan desa seperti pada nomor tujuh. Data kewenangan berisi data kewenangan yang dilaksanakan pemerintahan desa seperti pada nomor delapan. Data keuangan desa melingkupi Data pendapatan pada nomor sembilan, data pembiayaan desa pada nomor sepuluh, dan data belanja desa pada nomor sebelas. Data kelembagaan pemerintahan desa melingkupi data RT,RW,Dusun pada Nomor empat, dan data kelembagaan desa pada nomor 12. Nomor satu berguna sebagai penyimpanan data user dan dasar autentifikasi dan otorisasi penggunaan basis data monografi.

Tabel 3.5 Struktur Tabel Iterasi I

No	Nama Tabel	Atribut	Tipe
1	User	id	integer
		username	string
		password	string
		createdAt	date
		updatedAt	date
2	Data penduduk	id	integer

		nomorIndukPenduduk	bigint
		nama	string
		tempatLahir	string
		tanggalLahir	date
		namaIbu	string
		jenisKelamin	string
		pendidikanTerakhir	string
		agama	string
		pekerjaan	string
		status	string
		shdk	string
		alamat	string
		createdAt	date
		updateAt	date
3	Data Geografi Desa	namaDesa	string
		kodeWilayah	integer
		kodePos	integer
		kecamatan	string
		kabupaten	string
		provinsi	string
		koordinat	integer
		batasUtara	string
		batasSelatan	string
		batasBarat	string
		batasTimur	string
		luasWilayah	integer
		jarakDariPemerintahanPusat	integer
		jarakDariPusatPemerintahanProvinsi	integer
		jarakDariPusatPemerintahanKota	integer
		jarakDariPusatPemerintahanKecamatan	integer
		createdAt	date
		updatedAt	date

4	Data Lembaga Pemerintahan	id	integer
		ketuaRT	string
		nomorRT	integer
		nomorRW	integer
		ketuaRW	string
		dusun	string
		kepalaDusun	string
		jumlahPenduduk	integer
		createdAt	date
		updatedAt	date
5	Data Peristiwa Bencana	id	integer
		namaPeristiwa	string
		tanggalPeristiwa	date
		kategoriPeristiwa	string
		createdAt	date
		updatedAt	date
6	Data Peristiwa Keamanan	id	integer
		namaPeristiwa	string
		tanggalPeristiwa	date
		kategoriPeristiwa	string
		createdAt	date
		updatedAt	date
7	Data Aparatur Desa	id	integer
		namaAparatur	string
		jabatanAparatur	string
		tanggalMulaiMenjabat	date
		penghasilanAparatur	integer
		sumberPenghasilanAparatur	string
		tunjanganAparatur	integer
		sumberTunjanganAparatur	string
		createdAt	date
		updatedAt	date

8	Data Kewenangan Desa	id	integer
		namaKewenangan	string
		tanggalPenetapan	date
		kategoriKewenangan	string
		bidangKewenangan	string
		landasanHukum	file
		createdAt	date
		updateAt	date
9	Data Pendapatan Desa	id	integer
		kategoriPendapatan	string
		sumberPendapatan	string
		jumlah	string
		tanggalTransaksi	date
		buktiTransaksi	file
		keterangan	string
		createdAt	date
		updateAt	date
10	Data Pembiayaan Desa	id	integer
		kategoriPembiayaan	string
		jenisPembiayaan	string
		jumlahPembiayaan	integer
		tanggal Transaksi	date
		buktiTransaksi	file
		keterangan	string
		createdAt	date
		updateAt	date
11	Data Belanja Desa	id	integer
		kategoriBelanja	string
		jenisBelanja	string
		jumlahBelanja	integer
		tanggalBelanja	date
		buktiBelanja	file

		keterangan	string
		createdAt	date
		updateAt	date
12	Data Kelembagaan Desa	id	integer
		namaLembaga	string
		ketuaLembaga	string
		kegiatanLembaga	string
		jumlahPengurus	integer
		jumlahAnggota	integer
		createdAt	date
		updateAt	date

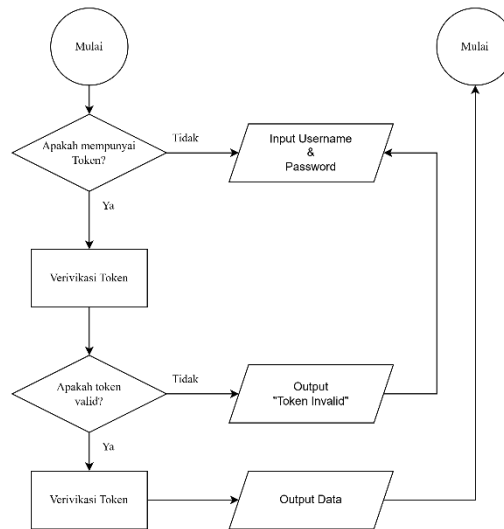
3.5.1.2 Desain ERD

Pada tahap ini, dilakukan desain diagram entitas-hubungan (ERD) berdasarkan struktur tabel yang telah dinormalisasi hingga bentuk normal ketiga (BCNF). Proses normalisasi ini memastikan bahwa tabel-tabel yang dibuat memiliki struktur yang efisien dan bebas dari anomali. Gambar 3.3 memberikan ilustrasi mengenai hasil akhir dari proses ini, menampilkan bagaimana entitas-entitas dan hubungan-hubungan antara mereka diorganisir secara logis dan sistematis. Dengan adanya desain ERD yang telah dioptimalkan melalui normalisasi, sistem basis data diharapkan dapat berfungsi dengan lebih baik dalam menyimpan dan mengelola data, sekaligus meminimalkan redudansi dan inkonsistensi data.

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 REST API melalui protokol HTTP.

Alur sistem REST API, seperti yang ditunjukkan pada Gambar 3.4, 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 autentifikasi 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 3.5 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 dengan dilakukan pemeriksaan tanda tangan (*signature*). Jika token tidak valid, *server* akan merespons dengan "Token Invalid". Apabila token valid *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 3.5 Arsitektur Autentifikasi dan Otorisasi JSON Web Token Iterasi I

3.5.1.4 Desain Endpoint API

Tahapan perancangan endpoint adalah langkah-langkah yang dirancang sebagai panduan bagi peneliti dalam mengembangkan endpoint yang akan digunakan oleh pengguna. Proses ini melibatkan pengelompokan kebutuhan sistem ke dalam beberapa kategori, seperti yang dijelaskan pada Tabel 3.6 Desain Endpoint Iterasi I. Pada kolom endpoint, dicantumkan komponen URL API yang akan digunakan untuk akses data atau layanan tertentu. Kolom kebutuhan menjelaskan aktivitas spesifik yang akan dilakukan oleh setiap endpoint, sementara kolom deskripsi memberikan penjelasan mendetail tentang aktivitas yang terdaftar dalam kolom kebutuhan.

Tabel 3.6 Desain Endpoint Iterasi I

Kebutuhan	Endpoint	Metode	Deskripsi
Autentifikasi	/user/sign-up	POST	Proses Mencatat User baru
	/user/login	POST	Proses Mendapat token
Pengelolaan Data Penduduk	/dataPenduduk	POST	Proses mencatat data penduduk baru
		GET	Proses mengambil seluruh data penduduk
	/dataPenduduk/:id	GET	Proses mengambil detail dari satu data penduduk

		PATCH	Proses memperbaiki data penduduk
		DELETE	Proses menghapus data penduduk
Pengelolaan Data Dusun	/dusun	POST	Proses mencatat data dusun baru
		GET	Proses mengambil seluruh data dusun
	/dusun/:id	GET	Proses mengambil detail data dusun
		PATCH	Proses memperbaiki data dusun
		DELETE	Proses menghapus data dusun
Pengelolaan Data Rukun Warga	/rukunWarga	POST	Proses mencatat data RW baru
		GET	Proses mengambil seluruh data RW
	/rukunWarga/:id	GET	Proses mengambil detail data RW
		PATCH	Proses memperbaiki data RW
		DELETE	Proses menghapus data RW
Pengelolaan Data Rukun Tetangga	/rukunTetangga	POST	Proses mencatat data RT baru
		GET	Proses mengambil seluruh data RT
	/rukunWarga/:id	GET	Proses mengambil detail data RT
		PATCH	Proses memperbaiki data RT

		DELETE	Proses menghapus data RT
Pengelolaan Data Lembaga Desa	/lembagaDesa	POST	Proses mencatat data lembaga desa baru
		GET	Proses mengambil seluruh data lembaga desa
	/lembagaDesa/:id	GET	Proses mengambil detail data lembaga desa
		PATCH	Proses memperbarui data lembaga desa
		DELETE	Proses menghapus data lembaga desa
Pengelolaan Data Kepengurusan Lembaga Desa	/kepengurusanLembagaDesa	POST	Proses mencatat data kepengurusan lembaga desa baru
		GET	Proses mengambil seluruh data kepengurusan lembaga desa
	/kepengurusanLembagaDesa/:id	GET	Proses mengambil detail data kepengurusan lembaga desa
		PATCH	Proses memperbarui data kepengurusan lembaga desa
		DELETE	Proses menghapus data kepengurusan lembaga desa
Pengelolaan Data Kegiatan Lembaga Desa	/kegiatanLembagaDesa	POST	Proses mencatat data kegiatan lembaga desa baru

		GET	Proses mengambil seluruh data kegiatan lembaga desa
	/kegiatanLembagaDesa/:id	GET	Proses mengambil detail data kegiatan lembaga desa
		PATCH	Proses memperberui data kegiatan lembaga desa
		DELETE	Proses menghapus data kegiatan lembaga desa
Pengelolaan Data Kewenangan Desa	/kewenanganDesa	POST	Proses mencatat data kewenangan desa baru
		GET	Proses mengambil seluruh data kewenangan desa
	/kewenanganDesa/:id	GET	Proses mengambil detail data kewenangan desa
		PATCH	Proses memperbarui data kewenangan desa
		DELETE	Proses menghapus data kewenangan desa
Pengelolaan Data Bidang Kewenangan Desa	/bidangKewenangan	POST	Proses mencatat data bidang kewenangan baru
		GET	Proses mengambil seluruh data bidang kewenangan
	/bidangKewenangan/:id	GET	Proses mengambil detail data bidang kewenangan
		PATCH	Proses memperbarui data bidang kewenangan

		DELETE	Proses menghapus data bidang kewenangan
Pengelolaan Data Sumber Penghasilan Aparatur Desa	/sumberPenghasilanAparatur	POST	Proses mencatat data sumber penghasilan aparatur baru
		GET	Proses mengambil seluruh data sumber penghasilan aparatur
	/sumberPenghasilanAparatur/:id	PATCH	Proses memperbarui data sumber penghasilan aparatur
		DELETE	Proses menghapus data sumber penghasilan aparatur
Pengelolaan Data Kategori Jabatan Aparatur Desa	/kategoriJabatanAparaturDesa	POST	Proses mencatat data kategori jabatan aparatur desa baru
		GET	Proses mengambil seluruh data kategori jabatan aparatur desa
	/kategoriJabatanAparaturDesa/:id	PATCH	Proses memperbarui data kategori jabatan aparatur desa
		DELETE	Proses menghapus data kategori jabatan aparatur desa
Pengelolaan Data Aparatur Desa	/aparaturDesa	POST	Proses mencatat data aparatur desa baru
		GET	Proses mengambil seluruh data aparatur desa

	/aparaturDesa/:id	GET	Proses mengambil detail data aparatur desa
		PATCH	Proses memperbarui data aparatur desa
		DELETE	Proses menghapus data aparatur desa
Pengelolaan Data Penghasilan Aparatur Desa	/penghasilanAparatur	POST	Proses mencatat data penghasilan aparatur desa baru
		GET	Proses mengambil seluruh data penghasilan aparatur desa
	/penghasilanAparatur/:id	GET	Proses mengambil detail data penghasilan aparatur desa
		PATCH	Proses memperbarui data penghasilan aparatur desa
		DELETE	Proses menghapus data penghasilan aparatur desa
Pengelolaan Data Peristiwa Keamanan Desa	/peristiwaKeamanan	POST	Proses mencatat data peristiwa keamanan desa baru
		GET	Proses mengambil seluruh data peristiwa keamanan desa
	/peristiwaKeamanan/:id	GET	Proses mengambil detail data peristiwa keamanan desa

		PATCH	Proses memperbarui data peristiwa keamanan desa
		DELETE	Proses menghapus data peristiwa keamanan desa
Pengelolaan Data Peristiwa Bencana Desa	/peristiwaBencana	POST	Proses mencatat data peristiwa bencana desa baru
		GET	Proses mengambil seluruh data peristiwa bencana desa
	/ peristiwaBencana /:id	GET	Proses mengambil detail data peristiwa bencana desa
		PATCH	Proses memperbarui data peristiwa bencana desa
		DELETE	Proses menghapus data peristiwa bencana desa
Pengelolaan Data Pendapatan Desa	/pendapatanDesa	POST	Proses mencatat data pendapatan desa baru
		GET	Proses mengambil seluruh data pendapatan desa
	/ pendapatanDesa /:id	GET	Proses mengambil detail data pendapatan desa
		PATCH	Proses memperbarui data pendapatan desa
		DELETE	Proses menghapus data pendapatan desa

Pengelolaan Data Belanja Desa	/belanjaDesa	POST	Proses mencatat data belanja desa baru
		GET	Proses mengambil seluruh data belanja desa
	/belanjaDesa /:id	GET	Proses mengambil detail data belanja desa
		PATCH	Proses memperbarui data belanja desa
		DELETE	Proses menghapus data belanja desa
Pengelolaan Data Pembiayaan Desa	/pembiayaanDesa	POST	Proses mencatat data pembiayaan desa baru
		GET	Proses mengambil seluruh data pembiayaan desa
	/pembiayaanDesa /:id	GET	Proses mengambil detail data pembiayaan desa
		PATCH	Proses memperbarui data pembiayaan desa
		DELETE	Proses menghapus data pembiayaan desa
Pengelolaan Data Geografi Desa	/geografiDesa	POST	Proses mencatat data geografi desa baru
		GET	Proses mengambil seluruh data geografi desa
	/geografiDesa /:id	GET	Proses mengambil detail data geografi desa
		PATCH	Proses memperbarui data geografi desa

		DELETE	Proses menghapus data geografi desa
--	--	--------	-------------------------------------

3.5.1.5 Evaluasi Desain

Evaluasi desain dalam iterasi pertama dilakukan pada tanggal 3 Juni 2024 dengan pihak pemerintahan desa dan pihak pengembang IT di Balai Desa Tunjungan. Evaluasi diuraikan sebagai berikut:

a. Pemerintahan Desa

1. Menambah struktur Sarana dan Prasarana sebagai alat monitoring sarana prasarana dalam lingkup desa.
2. Menambah struktur Sumber Daya Alam Desa sebagai sumber data informasi mengenai potensi sumber daya alam desa.

b. Pengembang IT

1. Menambah tenggat waktu untuk JSON Web Token (JWT) bertujuan untuk memaksimalkan perlindungan keamanan data desa.
2. Membuat akses untuk melihat nilai atribut pada tabel tabel pendukung.

3.5.2 Iterasi II

3.5.2.1 Desain Struktur Tabel

Pada evaluasi desain dalam iterasi pertama, pemerintah desa menambahkan dua struktur tabel baru. Pertama, tabel Sarana dan Prasarana, yang merupakan bagian dari data umum dan termasuk dalam kriteria monografi desa. Atribut serta jenis nilai dalam tabel Sarana dan Prasarana dapat dilihat pada Tabel 3.7 nomor satu. Kedua, tabel Sumber Daya Alam Desa, yang menurut pemerintah desa akan memberikan dasar yang kuat bagi pembangunan dan pemberdayaan potensi desa. Atribut dan jenis nilai data pada tabel Sumber Daya Alam dapat dilihat pada Tabel 3.7 nomor dua.

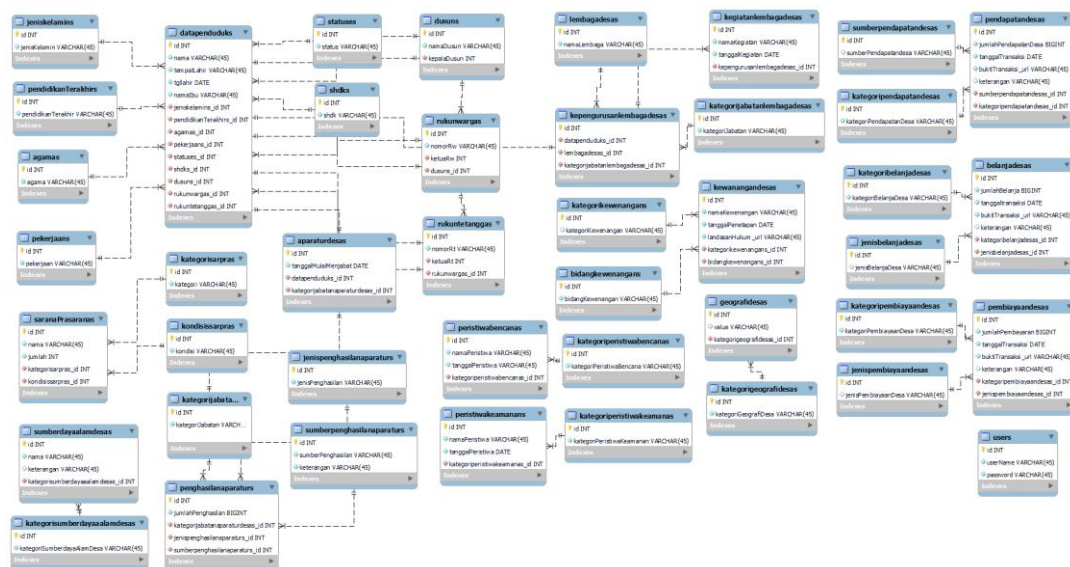
Tabel 3.7 Penambahan Desain Struktur Tabel Iterasi II

No	Nama Tabel	Atribut	Tipe
1	Data Sarana Prasarana	id	integer
		namaSarpras	string
		jumlah	integer
		kategoriSarpras	string
		kondisiSarpras	string

		createdAt	date
		updateAt	date
2	Data Sumber Daya Alam Desa	id	integer
		nama	string
		kategotiSDA	string
		keterangan	string
		createdAt	date
		updateAt	date

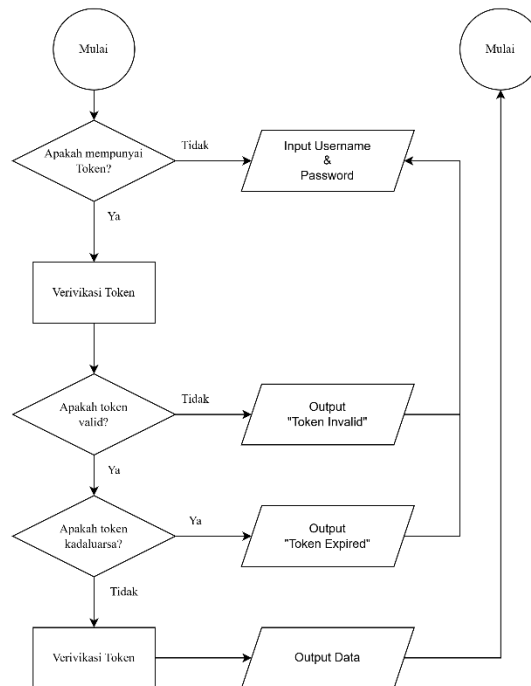
3.5.2.2 Desain ERD

Penambahan dalam desain struktur tabel pada iterasi II berpengaruh pada tahap normalisasi dan Entity Relationship Diagram (ERD). Jumlah tabel dalam ERD meningkat seiring dengan penambahan data, menghasilkan total 43 tabel, di mana 14 di antaranya merupakan tabel utama. Tabel-tabel tambahan mencakup Sarana Prasarana, Kategori Sarpras, Kondisi Sarpras, Sumber Daya Alam Desa, dan Kategori SDA. Hasil perubahan pada tahap normalisasi dan ERD dapat dilihat pada Gambar 3.6.



Gambar 3.6 Entity Relationship Diagram Iterasi II

3.5.2.3 Desain Arsitektur Sistem



Gambar 3.7 Arsitektur Autentifikasi dan Otorisasi JSON Web Token Iterasi II

Gambar 3.7 menunjukkan hasil pengembangan arsitektur JSON Web Token (JWT) pada iterasi pertama. Pengembangan arsitektur ini meliputi penambahan satu tahap verifikasi token kadaluarsa. Setelah sistem memverifikasi validitas token, tahap berikutnya adalah verifikasi tenggat waktu token. Pada sistem ini, token yang dihasilkan memiliki masa aktif selama 60 detik. Apabila token digunakan setelah masa aktifnya berakhir, sistem akan merespons dengan pesan “Token Expired.” Namun, untuk token yang masih berada dalam tenggat waktu, sistem akan memverifikasi token tersebut dan melanjutkan permintaan (request) ke basis data.

3.5.2.4 Desain Endpoint API

Dalam perancangan endpoint, telah dilakukan penyesuaian yang sejalan dengan perkembangan desain struktur tabel. Pada iterasi kedua ini, peneliti menambahkan endpoint yang berfungsi untuk melihat nilai atribut pada tabel-tabel pendukung, antara lain tabel agama, pendidikan terakhir, jenis kelamin, status hubungan dalam keluarga (SHDK), status, jenis sarana dan prasarana, kondisi sarana dan prasarana, kategori jabatan lembaga desa, kategori kewenangan, jenis penghasilan aparatur, kategori peristiwa keamanan, kategori peristiwa bencana, sumber pendapatan desa, kategori pendapatan desa, jenis belanja desa, kategori belanja desa, jenis pembiayaan desa, kategori pembiayaan desa, kategori geografi desa, dan kategori sumber daya alam desa. Selain itu, peneliti juga menambahkan endpoint berdasarkan

fungsi kebutuhan pada tabel struktur baru, yaitu tabel sarana dan prasarana serta sumber daya alam desa. Pembaruan endpoint dapat dilihat pada Tabel 3.8.

Tabel 3.8 Pengembangan Desain Endpoint Iterasi II

Kebutuhan	Endpoint	Metode	Deskripsi
Pengelolaan Data Penduduk	/dataPenduduk/v/Agama	GET	Proses mengambil detail tabel agama
	/dataPenduduk/v/PendidikanTerakhir	GET	Proses mengambil detail tabel pendidikan terakhir
	/dataPenduduk/v/JenisKelamin	GET	Proses mengambil detail tabel jenis kelamin
	/dataPenduduk/v/Shdk	GET	Proses mengambil detail tabel shdk
	/dataPenduduk/v/Status	GET	Proses mengambil detail tabel status
Pengelolaan Data Sarana Dan Prasarana	/saranaPrasarana	POST	Proses mencatat data sarana dan prasarana baru
		GET	Proses mengambil seluruh data sarana dan prasarana
	/saranaPrasarana/:id	GET	Proses mengambil detail data sarana dan prasarana
		PATCH	Proses memperbarui data sarana prasarana
		DELETE	Proses menghapus data sarana dan prasarana
	/saranaPrasarana/v/JenisSarpras	GET	Proses mengambil data tabel jenisSarpras

	/saranaPrasarana/v/ KondisiSarpras	GET	Proses mengambil data tabel kondisiSarpras
Pengelolaan Data Kepengurusan Lembaga Desa	/kepengurusanLembagaDesa /v/KategoriJabatanLembaga Desa	GET	Proses mengambil data tabel kategori jabatan lembaga desa
		GET	Proses mengambil seluruh data kepengurusan lembaga desa
Pengelolaan Data Kewenangan Desa	/kewenanganDesa/v/ KategoriKewenangan	GET	Proses mengambil data tabel kategori kewenangan
Pengelolaan Data Penghasilan Aparatur Desa	/penghasilanAparatur/v/ JenisPenghasilanAparatur	GET	Proses mengambil data tabel jenis penghasilan aparatur
Pengelolaan Data Peristiwa Keamanan Desa	/ peristiwaKeamanan/v/ KategoriPeristiwaKeamanan	GET	Proses mengambil data tabel kategori peristiwa keamanan desa
Pengelolaan Data Peristiwa Bencana Desa	/ peristiwaBencana /v/ KategoriPeristiwaBencana	GET	Proses mengambil data tabel kategori peristiwa bencana desa
Pengelolaan Data Pendapatan	/pendapatanDesa /v/SumberPendapatanDesa	GET	Proses mengambil data tabel sumber pendapatan desa
	pendapatanDesa /v/KategoriPendapatanDesa	GET	Proses mengambil data tabel kategori pendapatan desa
Pengelolaan Data Belanja Desa	/belanjaDesa /v/JenisBelanjaDesa	GET	Proses mengambil data tabel jenis pembiayaan desa

	belanjaDesa /v/KategoriBelanjaDesa	GET	Proses mengambil data tabel kategori belanja desa
Pengelolaan Data Pembiayaan Desa	/pembiayaanDesa /v/JenisPembiayaanDesa	GET	Proses mengambil data tabel jenis pembiayaan desa
	pembiayaanDesa /v/KategoriPembiayaanDesa	GET	Proses mengambil data tabel kategori pembiayaan desa
Pengelolaan Data Geografi Desa	/geografiDesa /v/KategoriGeografiDesa	GET	Proses mengambil data tabel kategori geografi desa
Pengelolaan Data Sumber Daya Alam Desa	/ sumberDayaAlamDesa	POST	Proses mencatat data sumber daya alam desa baru
		GET	Proses mengambil seluruh data sumber daya alam desa
	/sumberDayaAlamDesa /:id	GET	Proses mengambil detail data sumber daya alam desa
		PATCH	Proses memperbarui data sumber daya alam desa
		DELETE	Proses menghapus data sumber daya alam desa
	/sumberDayaAlamDesa / KategoriSumberDayaAlamDesa	GET	Proses mengambil data tabel kategori sumber daya alam desa

3.5.2.5 Evaluasi Desain

Evaluasi desain pada iterasi kedua dilaksanakan pada tanggal 17 Agustus 2024 bersama pihak pemerintahan desa dan pihak pengembang IT di Balai Desa Tunjungan. Dalam evaluasi

desain ini, hanya pihak pengembang IT yang memberikan saran agar terdapat satu token yang dihasilkan dengan tenggat waktu (lifetime) tertentu, yang akan disimpan oleh pemerintahan desa terkait sebagai cadangan. Mengingat evaluasi desain pada iterasi kedua ini bersifat minor, tahap pengembangan dapat dilanjutkan ke tahap implementasi desain.

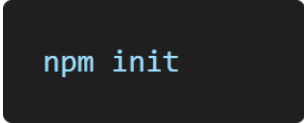
BAB IV

HASIL DAN PEMBAHASAN

4.1 Implementasi

4.1.1 Konfigurasi

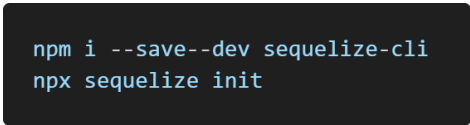
Tahap implementasi atau pengkodean dimulai dengan menginisiasi proyek Node.js menggunakan Node Package Manager (npm) seperti yang ditunjukkan pada kode di Gambar 4.1. Setelah inisiasi npm dilakukan, akan muncul file package.json yang berfungsi untuk mengelola informasi terkait proyek tersebut.



```
npm init
```

Gambar 4.1 Inisiasi Proyek NodeJs

Dalam pembangunan basis data, peneliti menerapkan teknik Object-Relational Mapping (ORM). ORM adalah teknik yang memetakan objek dalam bahasa pemrograman konvensional ke dalam struktur tabel basis data relasional (Syahputri & Nasution, 2023). Syahputri dan Nasution juga menyatakan bahwa penggunaan ORM dalam pembangunan basis data memungkinkan manipulasi data dengan lebih mudah, fleksibel, dan dapat diintegrasikan dengan bahasa pemrograman lainnya. Peneliti menggunakan sequelize-cli sebagai platform ORM dalam pembangunan basis data. Penggunaan sequelize-cli diawali dengan eksekusi kode seperti yang terlihat pada Gambar 4.2 dalam terminal, kemudian dilakukan konfigurasi ke dalam basis data seperti yang terlihat pada Gambar 4.3.



```
npm i --save-dev sequelize-cli  
npx sequelize init
```

Gambar 4.2 inialisasi ORM sequelize-cli

```

{
  "development": {
    "username": "root", //username basis data
    "password": null, //password basis data
    "database": "tugasakhir_monografidesa", // nama basis data
    "host": "127.0.0.1", //jaringan yang digunakan
    "dialect": "mysql", //basis data yang digunakan
    "timezone": "+07:00"
  },
  "test": {
    "username": "root",
    "password": null,
    "database": "database_test",
    "host": "127.0.0.1",
    "dialect": "mysql",
    "timezone": "+07:00"
  },
  "production": {
    "username": "root",
    "password": null,
    "database": "database_production",
    "host": "127.0.0.1",
    "dialect": "mysql",
    "timezone": "+07:00"
  }
}

```

Gambar 4.3 Konfigurasi ORM ke dalam basis data

Tahapan konfigurasi dilanjutkan dengan menginstal package yang diperlukan dalam pembangunan proyek seperti pada Tabel 4.1

Tabel 4.1 Package proyek

No	Kode	Deskripsi
1	Express	
	\$ npm i --save express	Framework ini merupakan kerangka kerja aplikasi web Node.js yang efisien dan fleksibel (NASUTION, 2022). Dalam proyek ini, peneliti memanfaatkan pustaka Express untuk mengirim permintaan (<i>request</i>) ke basis data.
2	Body Parser	
	\$ npm i --save body-parser	Proyek ini menggunakan properti <code>req`</code> dan <code>res`</code> dari <code>body-parser`</code> . Properti tersebut berfungsi untuk memproses data JSON. Hasilnya adalah konversi data JSON menjadi objek JavaScript.
3	Nodemon	

	<code>\$ npm i --save nodemon</code>	Dalam proyek ini, nodemon digunakan untuk otomatisasi <i>server</i> . Ini memungkinkan <i>server</i> secara otomatis memuat ulang ketika ada perubahan pada kode. Dengan demikian, peneliti tidak perlu secara manual merestart <i>server</i> setelah setiap perubahan.
4	Mysql2	
	<code>\$ npm i --save mysql2</code>	MySQL2 adalah klien MySQL untuk Node.js yang mengutamakan performa. Klien ini mendukung berbagai fitur seperti pernyataan terprepared, pengkodean non-UTF8, protokol log biner, kompresi, dan SSL, serta banyak fitur lainnya.
6	Fastest Validator	
	<code>\$ npm i --save fastest-validator</code>	Proyek ini melakukan validasi terhadap tipe atribut yang diinputkan oleh pengguna. Dalam proses tersebut, digunakan package `fastest-validator` untuk memeriksa tipe atribut. Dengan demikian, validasi tipe objek dapat dilakukan dengan efisien.
7	Json Web Token	
	<code>\$ npm i --save jsonwebtoken</code>	Package `jsonwebtoken` digunakan untuk menghasilkan dan memvalidasi token. Token ini berfungsi dalam proses autentifikasi dan otorisasi akses ke basis data. Dengan menggunakan package ini, sistem dapat mengelola hak akses dengan lebih aman.
8	Bycryptjs	
	<code>\$ npm i --save bcryptjs</code>	Dalam proyek ini, paket `bcryptjs` digunakan untuk melakukan proses <i>hashing</i> pada kata sandi pengguna yang kemudian disimpan dalam basis data. Fungsi <i>hashing</i> ini memastikan keamanan

		kata sandi dengan mengubahnya menjadi format yang tidak dapat dibaca secara langsung.
9	Multer	
	<code>\$ npm i --save multer</code>	Proyek ini menyediakan fitur untuk mengunggah dokumen yang diperlukan guna mendukung kebutuhan basis data. Untuk menangani data bertipe `multipart/form-data`, digunakan paket `Multer`, yang memungkinkan dokumen diunggah ke <i>server</i> dengan efisien.
10	Path	
	<code>\$ npm i --save path</code>	Dalam proses penyimpanan dokumen, dilakukan penamaan ulang dokumen sesuai dengan format yang diunggah ke <i>server</i> . Penamaan ulang ini dilakukan dengan memanfaatkan paket `path`, yang membantu dalam mengelola nama dan struktur direktori dokumen. Dengan cara ini, dokumen disimpan dengan nama yang terorganisir dan sesuai dengan format yang ditentukan.

4.1.2 Model

Setelah menyelesaikan tahap konfigurasi, langkah selanjutnya adalah mengimplementasikan dan menghasilkan model dari tabel-tabel yang telah dirancang. Tahap ini memanfaatkan konfigurasi yang telah disediakan oleh ORM sequelize-cli. Konfigurasi tersebut harus mengikuti format yang ditunjukkan pada Gambar 4.4. Dengan demikian, setiap model akan sesuai dengan struktur tabel yang telah direncanakan.

```
npx sequelize-cli model:generate
--name TableName --attributes attributeName:type
```

Gambar 4.4 Generate Model

Perintah konfigurasi pada Gambar 4.4 akan menghasilkan dua berkas, yaitu berkas model tabel yang disimpan di dalam folder `models` dan berkas migrasi yang ditempatkan di dalam folder `migrations`. Pengkodean relasi dilakukan dalam berkas model tabel. Relasi antar tabel

dapat dilihat pada Tabel 4.2, di mana kolom Relasi berisi jenis relasi yang digunakan oleh kolom tabel terhadap kolom target tabel. Dan kolom peran menjelaskan peranan target tabel pada tabel

Tabel 4.2 Relasi antar tabel

Tabel	Relasi	Target Tabel	Peranan
AparaturDesa	1:1	Data_Penduduk	DataAparatur
	M:1	KategoriJabatanAparaturDesa	KategoriJabatan
BelanjaDesa	M:1	JenisBelanjaDesa	JenisBelanja
	M:1	KategoriBelanjaDesa	KategoriBelanja
Data_Penduduk	M:1	JenisKelamin	JenisKelamin
	M:1	Agama	Agama
	M:1	Pekerjaan	Pekerjaan
	M:1	PendidikanTerakhir	PendidikanTerakhir
	M:1	Shdk	Shdk
	M:1	Status	Status
	M:1	Dusun	AlamatDusun
	M:1	RukunWarga	AlamatRukunWarga
	M:1	RukunTetangga	AlamatRukunTetangga
Dusun	1:M	Data_Penduduk	WargaDusun
	1:1	Data_Penduduk	KepalaDusun
GeografiDesa	1:1	KategoriGeografiDesa	Kategori
KepengurusanLembagaDesa	M:1	LembagaDesa	KepengurusanLmbg
	M:1	KategoriJabatanLembagaDesa	KategoriJabatan
	M:1	Data_Penduduk	DetailAnggota
KewenanganDesa	M:1	KategoriKewenangan	KategoriKewenangan
	M:1	BidangKewenangan	BidangKewenangan
LembagaDesa	M:1	KepengurusanLembagaDesa	AnggotaLembaga
	M:1	KegiatanLembagaDesa	KegiatanLembaga
PembiayaanDesa	M:1	JenisPembiayaanDesa	JenisPembiayaan
	M:1	KategoriPembiayaanDesa	KategoriPembiayaan
PendapatanDesa	M:1	SumberPendapatanDesa	SumberPendapatan
	M:1	KategoriPendapatanDesa	KategoriPendapatan

PenghasilanAparatur	M:1	KategoriJabatanAparaturDesa	KategoriJabatan
	M:1	JenisPenghasilanAparatur	JenisPenghasilan
	M:1	SumberPenghasilanAparatur	SumberPenghasilan
PeristiwaBencana	M:1	KategoriPeristiwaBencana	KategoriPeristiwa
PeristiwaKeamanan	M:1	KategoriPeristiwaKeamanan	KategoriPeristiwa
RukunTetangga	1:1	Data_Penduduk	KepalaRt
	1:M	Data_Penduduk	Warga
	M:1	RukunWarga	AlamatRw
	M:1	Dusun	AlamatDusun
RukunWarga	1:1	Data_Penduduk	KepalaRw
	1:M	Data_Penduduk	Warga
	M:1	Dusun	AlamatDusun
SaranaPrasarana	M:1	KategoriSarpras	KategoriSarpras
	M:1	KondisiSarpras	KondisiSarpras
SumberDayaAlamDesa	M:1	KategoriSumberDayaAlamDesa	KategoriSDA

```
npx sequelize-cli db:migrate
```

Gambar 4.5 Konfigurasi Migrasi

Pada tahap ini, basis data masih kosong dan belum memiliki tabel maupun atribut. Untuk menginisiasi model yang telah dibuat ke dalam basis data, perlu dilakukan perintah migrasi yang disediakan oleh Sequelize, seperti yang terlihat pada Gambar 4.5 di terminal. Perintah ini akan menjalankan beberapa langkah (*Migrations* | *Sequelize*, 2024):

- Akan memastikan adanya tabel bernama `SequelizeMeta` di dalam basis data. Tabel ini digunakan untuk mencatat migrasi mana saja yang telah dijalankan pada basis data saat ini.
- Sistem kemudian akan memeriksa file migrasi yang belum dijalankan, hal ini dapat dilakukan dengan memeriksa tabel `SequelizeMeta`. Dalam hal ini, sistem akan menjalankan migrasi `XXXXXXXXXXXXXXXXX-create-namaTabel.js`, yang telah dibuat pada langkah sebelumnya.
- Sistem akan membuat tabel bernama `NamaTabel` dengan semua kolom sesuai yang telah ditentukan dalam file migrasi tersebut.

4.1.3 Seeding

Seed adalah segala sesuatu yang harus dimasukkan ke dalam sistem untuk memastikan aplikasi dapat berjalan dengan baik (Jayadi & Nugraha, 2019). Sementara itu, seeding adalah aktivitas memasukkan seed ke dalam sistem. Dalam konteks ini, seed yang dimasukkan adalah data yang diperlukan untuk melengkapi nilai atribut dalam basis data.

```
npx sequelize-cli seed:generate --name tableName
```

Gambar 4.6 Generate Seeder

Perintah pada Gambar 4.6 akan menghasilkan file seeder yang disimpan dalam folder `seeders`. Pengkodean seeder dilakukan dalam file tersebut dengan format JSON untuk atribut dan nilai atribut. Pengisian data default seeder dilakukan pada tabel yang nilainya telah ditetapkan. Diantaranya:

a. Jenis Kelamin

Tabel jenisKelamin berisi atribut id dan jenisKelamin. Nilai atribut jenisKelamin pada Tabel 4.3 ditetapkan berdasarkan (*UU No. 24 Tahun 2013, t.t.*) pasal 64.

Tabel 4.3 Seeder Jenis Kelamin

Id	jenisKelamin
1	Laki-laki
2	Perempuan

b. Pendidikan Terakhir

Tabel Pendidikan terakhir berisi atribut id dan pendidikanTerakhir. Penetapan nilai pendidikanTerakhir diambil dari tiga jenjang yang ditetapkan oleh pemerintahan indonesia. Pendidikan dasar yang mencakupi taman kanak-kanak sampai dengan SMP atau sederajat, pendidikan menengah yaitu SMA atau sederajat, dan pendidikan tinggi yang mencakupi Akademi atau D1-D3 sampai dengan Pasca Sarjana (*Permendikbudriset No. 8 Tahun 2024, t.t.; UU No. 12 Tahun 2012, t.t.*). id dan pendidikanTerakhir dapat dilihat dalam Tabel 4.4.

Tabel 4.4 Seeder Pendidikan Terakhir

Id	pendidikanTerakhir
1	Taman Kanak-kanak
2	Sekolah Dasar
3	SMP atau Sederajat

4	SMA atau Sederajat
5	Akademi atau D1-D3
6	Sarjana
7	Pasca Sarjana

c. Agama

Tabel agama pada basis data berisi dua atribut, yaitu atribut id dan agama seperti pada Tabel 4.5. Nilai atribut agama ditetapkan berdasarkan (*Penetapan Presiden Nomor 1 Tahun 1965 - Pusat Data Hukumonline*, t.t.) yang memuat enam agama yang diakui oleh negara.

Tabel 4.5 Seeder Agama

Id	Agama
1	Islam
2	Kristen
3	Katolik
4	Hindu
5	Budha
6	Khonghucu

d. Status Hubungan Dalam Keluarga

Dalam tabel status hubungan dalam keluarga (SHDK) terdapat atribut id dan shdk yang bernilai kategori shdk yang ditetapkan. Penetapan nilai pada atribut shdk berlandaskan (*PERMENDAGRI No. 95 Tahun 2019*, t.t.) yang mengatur sistem informasi administrasi kependudukan. Nilai id dan shdk seperti yang terdapat pada Tabel 4.6.

Tabel 4.6 Seeder SHDK

Id	Shdk
1	Kepala Keluarga
2	Suami
3	Istri
4	Anak
5	Menantu
6	Cucu
7	Orangtua

8	Mertua
9	Famili Lain
10	Pembantu
11	Lainya

e. Status

Pada Tabel 4.7 terdapat atribut id dan status. Nilai dalam atribut status di tetapkan berdasarkan (*PERMENDAGRI No. 95 Tahun 2019, t.t.*) yang mengatur sistem informasi administrasi kependudukan.

Tabel 4.7 Seeder Status

Id	Status
1	Belum Kawin
2	Kawin Belum Tercatat
3	Kawin Tercatat
4	Cerai Hidup
5	Cerai Mati

f. Pekerjaan

Pada tabel pekerjaan terdapat atribut id dan pekerjaan seperti yang terlihat pada Tabel 4.8. Nilai pada atribut pekerjaan ditetapkan berdasarkan pekerjaan rata-rata masyarakat indonesia diatara 89 kategori pekerjaan yang ditetapkan oleh menteri dalam negeri pada (*PERMENDAGRI No. 19 Tahun 2010, t.t.*). Dalam pengembanganya pengguna dapat menambahkan kategori pekerjaan sesuai yang ditetapkan.

Tabel 4.8 Seeder Pekerjaan

Id	Pekerjaan
1	PNS/ASN
2	Karyawan Swasta
3	Petani
4	Nelayan
5	Buruh
6	Ibu Rumah Tangga
7	Pensiunan

8	Pelajar/Mahasiswa
9	Tenaga Kerja Informal

g. Kondisi Sarpras

Data seed yang digunakan pada tabel kondisiSarpras dicantumkan pada Tabel 4.9. Penetapan nilai kondisi berdasarkan kondisi penggunaan sarana prasarana yang ada dalam lingkungan desa

Tabel 4.9 Seeder Kondisi sarpras

Id	Kondisi
1	Aktif
2	Non-Aktif

h. Jenis Sarpras

Dalam tabel jenis sarpras terdapat atribut id dan kategori, kategori dalam hal ini adalah kategori sarana dan prasarana. Penetapan nilai kategori berdasarkan (*PERMENDAGRI No. 1 Tahun 2017*, t.t.) seperti pada Tabel 4.10.

Tabel 4.10 Seeder Jenis Sarpras

Id	Kategori
1	Umum
2	Pendidikan
3	Kesehatan
4	Ibadah
5	Lainya

i. Kategori Jabatan Lembaga Desa

Nilai kategori jabatan lembaga desa ditetapkan berdasarkan (*PERMENDAGRI No. 18 Tahun 2018*, t.t.) seperti pada Tabel 4.11.

Tabel 4.11 Seeder Jabatan Lembaga Desa

Id	kategoriJabatanLembagaDesa
1	Ketua
2	Sekretaris
3	Bendahara

4	Anggota
5	Lainya

j. Kategori Jabatan Aparatur Desa

Terdapat atribut id dan kategoriJabatanAparaturDesa pada tabel kategori jabatan aparatur desa seperti pada Tabel 4.12. Penetapan nilai kategoriJabatanAparaturDesa berdasarkan permendagri yang menyatakan bahwa pelaksana pemerintahan desa yaitu kepala desa, sekretaris desa, dan ketua BPD atau yang disebut dengan nama lain dibantu perangkat desa sebagai penyelenggara pemerintah (*PERMENDAGRI No. 73 Tahun 2020, t.t.*). Selanjutnya pengguna dapat menambahkan jabatan yang sesuai dengan kebutuhan pada endpoint yang disediakan.

Tabel 4.12 Seeder Jabatan Aparatur Desa

Id	kategoriJabatanAparaturDesa
1	KepalaDesa
2	Sekretaris Desa
3	Ketua BPD

k. Kategori Kewenangan

Kategori kewenangan berisi atribut id dan kategoriKewenangan. Nilai atribut kategoriKewenangan pada Tabel 4.13 ditetapkan berdasarkan (*UNDANG-UNDANG REPUBLIK INDONESIA NOMOR 6 TAHUN 2014 TENTANG DESA – Laboratorium Politik Dan Tata Pemerintahan, t.t.*)

Tabel 4.13 Seeder Kategori Kewenangan

Id	kategoriKewenangan
1	Peraturan Desa
2	Tugas atau program dari pemerintah pusat
3	Tugas atau program dari pemerintah provinsi
4	Tugas atau program dari pemerintah kabupaten/kota

l. Jenis Penghasilan Aparatur

Berdasarkan Peraturan Pemerintah No.43 Tahun 2014 Pasal 81 jenis penghasilan aparatur desa berupa penghasilan tetap dan tunjangan (*PP No. 43 Tahun 2014*, t.t.). Dasar tersebut digunakan untuk nilai atribut jenisPenghasilanAparatur pada Tabel 4.14.

Tabel 4.14 Seeder Jenis Penghasilan Aparatur

Id	jenisPenghasilanAparatur
1	Penghasilan Tetap
2	Tunjangan

m. Kategori Peristiwa Bencana

Tabel kategori peristiwa bencana berisi id dan kategori peristiwa bencana. Penetapan nilai kategori peristiwa bencana berdasarkan Undang-undang yang membagi tiga kategori seperti pada Tabel 4.15. (*UU No. 24 Tahun 2007*, t.t.).

Tabel 4.15 Seeder Kategori Peristiwa Bencana

Id	kategoriPeristiwaBencana
1	Bencana Alam
2	Bencana Non-Alam
3	Bencana Sosial

n. Kategori Peristiwa Keamanan

Tabel 4.16 menunjukkan nilai dari atribut id dan kategori peristiwa keamanan yang menjadi data seed. Penetapan ini berdasarkan pengembangan dari kategori kejahatan konvensional, transnasional, berimplikasi konjungsi, dan kejahatan yang menyebabkan kerugian negara yang di ungkapkan (Syahbana & Tejawati, 2018).

Tabel 4.16 Seeder Kategori Peristiwa Keamanan

Id	kategoriPeristiwaKeamanan
1	Operasi Keamanan
2	Pencurian
3	Pemeriksaan
4	Kenakalan Remaja
5	Pembunuhan
6	Perampokan
7	Penipuan

8	Tindakan Kekerasan
9	Penyalahgunaan Zat dan Obat Terlarang
10	Tindak Kriminalitas Lainnya
11.	Lainya

o. Kategori Pendapatan Desa

Tabel kategoriPendapatanDesa memuat atribut id dan kategoriPendapatanDesa Nilai atribut pada kolom kategoriPendapatanDesa Tabel 4.17 ditetapkan berdasarkan (PERMENDAGRI No. 20 Tahun 2018, t.t.).

Tabel 4.17 Seeder Kategori Pendapatan Desa

Id	kategoriPendapatanDesa
1	Pendapatan Asli Desa
2	Transfer
3	Pendapatan Lain

p. Sumber Pendapatan Desa

Tabel sumberPendapatanDesa memuat atribut id dan sumberPendapatanDesa. Berdasarkan (PERMENDAGRI No. 20 Tahun 2018, t.t.) sumber pendapatan desa dibagi sesuai dengan kategori yang tercantum dalam kolom sumberPendapatanDesa pada Tabel 4.18

Tabel 4.18 Seeder Sumber Pendapatan Desa

Id	sumberPendapatanDesa
1	Hasil Usaha
2	Hasil Aset
3	Swadaya, Partisipasi dan Gotong Royong
4	Pendapatan Asli Desa Lain
5	Dana Desa
6	Bagian dari hasil pajak daerah dan retribusi daerah kabupaten/kota
7	Bantuan Keuangan dari APBD Provinsi
8	Bantuan Keuangan dari APBD Kabupaten/Kota
9	Penerimaan dari Hasil Kerjasama Desa

10	Penerimaan dari bantuan perusahaan yang berlokasi di Desa
11	Penerimaan dari hibah dan sumbangan dari pihak ketiga
12	Koreksi kesalahan belanja tahun anggaran sebelumnya yang mengakibatkan penerimaan di kas Desa pada tahun anggaran berjalan
13	Bunga Bank
14	Pendapatan lain Desa yang sah

q. Kategori Belanja Desa

Nilai kategori belanja desa ditetapkan berdasarkan (*PERMENDAGRI No. 20 Tahun 2018, t.t.*). Rincian mengenai kategoriBelanjaDesa dapat dilihat dalam Tabel 4.19, yang memuat id dan kategoriBelanjaDesa.

Tabel 4.19 Seeder Kategori Belanja Desa

Id	kategoriBelanjaDesa
1	Penyelenggaraan Pemerintahan Desa
2	Pelaksanaan Pembangunan Desa
3	Pembinaan Kemasyarakatan Desa
4	Pemberdayaan Masyarakat Desa
5	Penanggulangan Bencana, Keadaan Darurat, Keadaan Mendesak Desa

r. Jenis Belanja Desa

Tabel 4.20 menunjukkan isi dari tabel jenisBelanjaDesa dalam basis data yaitu atribut id dan jenisBelanjaDesa. Nilai atribut jenisBelanjaDesa ditetapkan berdasarkan (*PERMENDAGRI No. 20 Tahun 2018, t.t.*).

Tabel 4.20 Seeder Jenis Belanja Desa

Id	jenisBelanjaDesa
1	Belanja Pegawai
2	Belanja Barang/Jasa

3	Belanja Modal
4	Belanja Tak Terduga

s. Kategori Pembiayaan Desa

Berdasarkan (*PERMENDAGRI No. 20 Tahun 2018*, t.t.), kategori pembiayaan desa terbagi seperti pada Tabel 4.21. Data seeder yang akan dimasukkan ke dalam tabel kategoriPembiayaanDesa dapat dilihat pada Tabel 4.19.

Tabel 4.21 Seeder Kategori Pembiayaan Desa

Id	kategoriPembiayaanDesa
1	Penerimaan Pembiayaan
2	Pengeluaran Pembiayaan

t. Jenis Pembiayaan Desa

Tabel 4.22 mencantumkan data yang menjadi data seed pada tabel jenisPembiayaanDesa. Nilai atribut jenisPembiayaanDesa ditetapkan dengan (*PERMENDAGRI No. 20 Tahun 2018*, t.t.) sebagai dasar nilai.

Tabel 4.22 Seeder Jenis Pembiayaan Desa

Id	jenisPembiayaanDesa
1	SilPA Tahun Sebelumnya
2	Pencairan Dana Cadangan
3	Hasil penjualan kekayaan Desa yang dipisahkan kecuali tanah dan bangunan
4	Pembentukan Dana Cadangan
5	Penyertaan Modal

u. Kategori Geografi Desa

Tabel kategoriGeografi desa berisi atribut isi dan kategoriGeografiDesa seperti pada Tabel 4.23. Nilai atribut kategoriGeografiDesa mengacu pada informasi mengenai geografi desa.

Tabel 4.23 Seeder Kategori Geografi Desa

Id	kategoriGeografiDesa
1	Nama Desa

2	Kode Wilayah
3	Kode Pos
4	Kecamatan
5	Kabupaten/Kota
6	Provinsi
7	Koordinat
8	Batas Utara
9	Batas Selatan
10	Batas Barat
11	Batas Timur
12	Luas Wilayah
13	Jarak dari Pusat Pemerintahan Kecamatan
14	Jarak dari Pusat Pemerintahan Kabupaten/Kota
15	Jarak dari Pusat Pemerintahan Provinsi
16	Jarak dari Pusat Pemerintahan Pusat

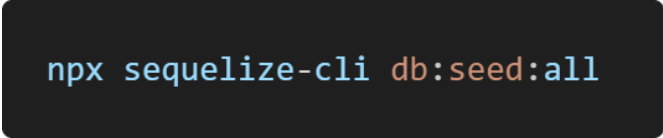
v. Kategori Sumber Daya Alam Desa

Dalam tabel kategoriSumberDayaAlamDesa nilai atribut ditetapkan berdasarkan (*Permendagri No. 13 Tahun 2012 Tentang Monografi Desa Dan Kelurahan, t.t.*) yang mengatur tentang pencatatan monografi. Nilai-nilai kategori terdapat pada Tabel 4.24

Tabel 4.24 Seeder Kategori Sumber Daya Alam Desa

Id	Kategorisumberdayaalamdesa
1	Pertanian
2	Perkebunan
3	Peternakan
4	Perikanan
5	Bahan Galian
6	Sumberdaya Air
7	Kualitas Lingkungan
8	Ruang Publik
9	Wisata

Pengisian data pada file seeders tidak akan mempengaruhi basis data yang digunakan. Untuk memindahkan seluruh data seed ke dalam basis data, diperlukan perintah migrasi seperti yang ditunjukkan pada Gambar 4.7. Berbeda dengan migrasi model basis data, data seed tidak menyimpan catatan apapun di dalam basis data. (*Migrations | Sequelize, 2024*).



```
npx sequelize-cli db:seed:all
```

Gambar 4.7 Migrasi Seeder

4.1.4 Modul

Tahapan dilanjutkan membagi pengkodean menjadi beberapa bagian modul. Modul terbagi sesuai dengan peranan fungsi masing-masing. Pembagian Modul terbagi menjadi modul pengguna, modul Autentifikasi, Controller Module, Route Module, FilesUpload Module.

a. Modul Pengguna

Modul Pengguna memiliki fitur-fitur berikut: registrasi pengguna baru, validasi keunikan nama pengguna, hashing kata sandi pengguna, login pengguna, dan pembuatan token.

b. Modul Autentifikasi

Modul ini memiliki fitur untuk autentifikasi dan otorisasi menggunakan JSON Web Token yang diterima saat registrasi dan login pengguna.

c. Controller Module

Modul controller memiliki peran dalam mengendalikan fungsi utama dalam manajemen basis data. Fitur post berfungsi untuk menambahkan data ke dalam basis data. Fitur Get bertugas untuk menunjukkan semua nilai yang ada dalam tabel tertentu di basis data. Fitur Get Detail dimanfaatkan untuk menampilkan informasi rinci dari data tertentu dalam tabel. Fitur delete dipergunakan untuk menghilangkan data spesifik dalam tabel. Sementara itu, fitur update digunakan untuk memperbaharui data yang ada dalam tabel basis data.

d. Route Module

Dalam route modul, pengelolaan endpoint disesuaikan untuk mendukung fungsi Get, Patch, Post, dan Delete. Selain itu, modul ini mengintegrasikan endpoint dengan modul terkait, serta middleware yang terlibat, untuk membentuk sebuah sistem yang terintegrasi secara keseluruhan.

e. FilesUpload Module

Modul ini mempunyai fitur untuk mengunggah berkas ke dalam folder *server*, menyaring berkas yang diunggah berdasarkan kriteria yang telah ditentukan, serta memberikan nama pada berkas sesuai dengan konfigurasi yang ditetapkan.

4.2 Pengujian

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

4.2.1 Pengujian Autentifikasi

Dalam pengujian autentifikasi, peneliti menguji fungsionalitas *register* pengguna, *login* pengguna, dan autentifikasi *JSON Web Token* dari sistem yang telah dikembangkan. Pengujian ini dilakukan menggunakan *software Postman* sebagai alat pengujian. Daftar pengujian yang menggunakan metode *blackbox* ditampilkan pada Tabel 4.25.

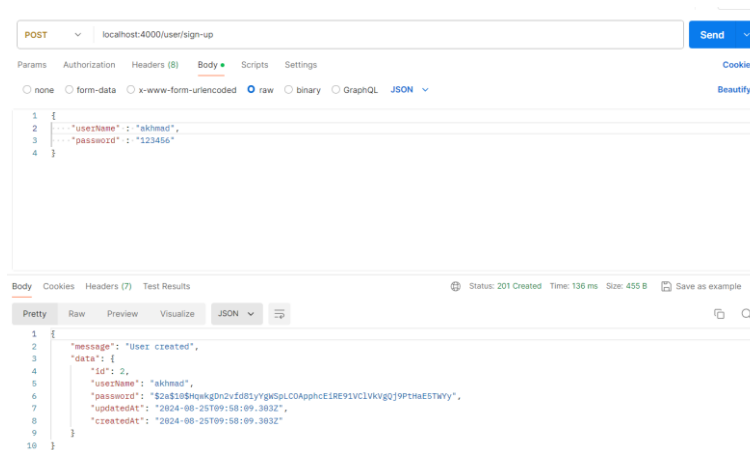
Tabel 4.25 Tabel Pengujian Autentifikasi

Id	Fungsi	Deskripsi Pengujian	Perkiraan Hasil	Hasil Pengujian	Kecepatan Respon	Kesimpulan
1A01	Register	Pengguna menggunakan endpoint <code>../user/signup</code> dan mengisi <code>userName</code> dan <code>password</code> 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-</code>	Sistem akan merespon dengan message	Sistem merespon dengan message	47 ms	Berhasil

		up dan mengisi userName yang telah digunakan dan password dengan format JSON di body	“User already exist”	“User already exist”		
1B0 1	Login	Pengguna menggunakan endpoint ../user/login dan mengisi userName dan password dengan format JSON di body yang telah didaftarkan	Sistem akan merespon dengan message “authentication succes” dan JSON Web Token yang telah digenerate	Sistem merespon dengan message “authentication succes” dan JSON Web Token yang telah digenerate	106 ms	Berhasil
1B0 2		Pengguna menggunakan endpoint ../user/login dan mengisi userName dan password	Sistem akan merespon dengan message “UserName or password incorrect”	Sistem merespon dengan message “UserName or password incorrect”	15 ms	Berhasil

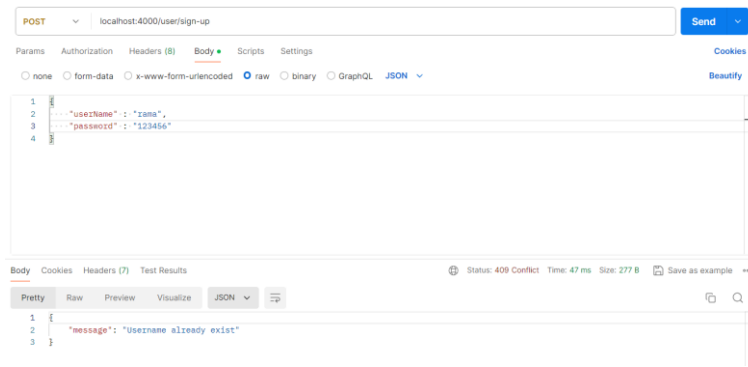
		yang salah dengan format JSON di body				
1C0 1	JSON Web Token	Pengguna melakukan request data di endpoint yang sesuai, dan menyertakan JSON Web Token di header	Sistem akan merespon sesuai dengan request pengguna	Sistem merespon sesuai dengan request pengguna	15 ms	Berhasil
1C0 2		Pengguna melakukan request data di endpoint yang sesuai, dan menyertakan JSON Web Token salah di header atau tidak menyertakan JSON Web Token	Sistem akan merespon dengan message “Invalid Token”	Sistem merespon dengan message “Invalid Token”	8 ms	Berhasil
1C0 3		Pengguna melakukan request data di endpoint	Sistem akan merespon dengan message	Sistem merespon dengan message	8 ms	Berhasil

		yang sesuai, dan menyertakan JSON Web Token yang telah kadaluarsa di header	“jwt expired”	“jwt expired”		
--	--	--	------------------	------------------	--	--



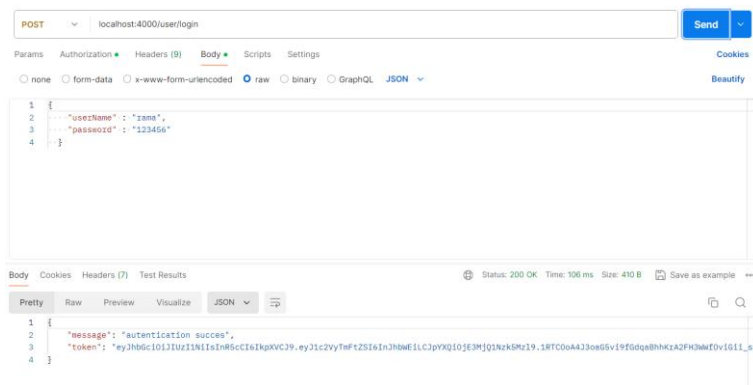
Gambar 4.8 Pengujian Register User 1A01

Pengujian fungsionalitas pertama 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 memberikan respons berupa pesan "user created" yang disertai dengan data pengguna, seperti nama pengguna, kata sandi yang telah di-hash, serta informasi waktu pembaruan (updatedAt) dan waktu pembuatan (createdAt) dalam waktu 136 ms, sebagaimana ditunjukkan pada Gambar 4.8



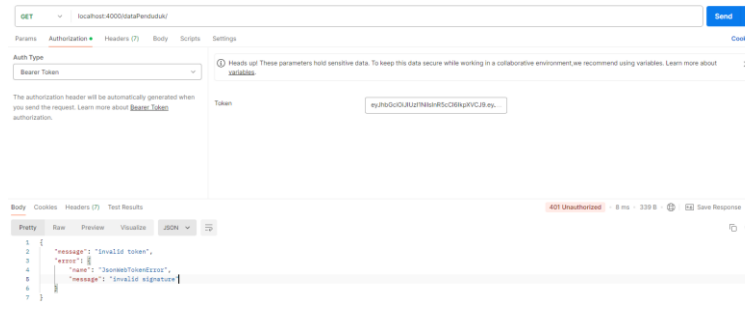
Gambar 4.9 Pengujian Register User 1A02

Pengujian dengan ID 1A02 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" dalam 47 ms. Hasil pengujian seperti pada Gambar 4.9.



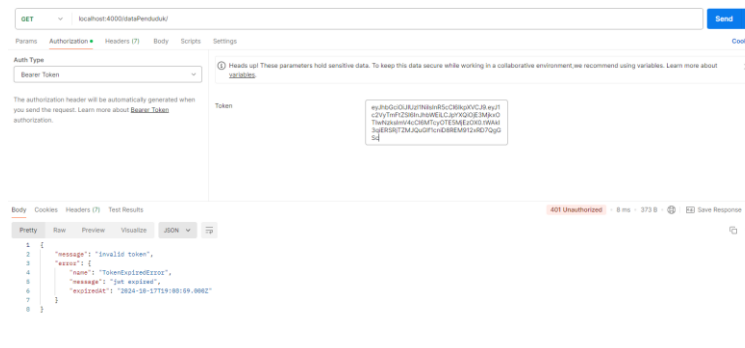
Gambar 4.10 Pengujian Login User 1B01

Pengujian 1B01 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 seperti pada Gambar 4.10. Berdasarkan pengujian kecepatan respon sistem 106 ms.



Gambar 4.13 Pengujian JSON Web Token 1C02

Pengujian dengan ID 1C02 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. Pengujian ini ditunjukkan melalui Gambar 4.13 dengan kecepatan respon 8 ms.



Gambar 4.14 Pengujian JSON Web Token 1C03

Pengujian terakhir dalam bagian Autentifikasi adalah pengujian dengan ID 1C03. 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". Hasil pengujian terdapat pada Gambar 1.14 dengan kecepatan respon 8 ms.

Hasil pengujian autentikasi menunjukkan bahwa sistem berhasil memenuhi seluruh kebutuhan fungsional yang telah dirumuskan oleh peneliti sebelumnya. Dalam proses autentikasi, sistem ini mampu melakukan pendaftaran pengguna, menyediakan fitur login, serta menghasilkan token yang digunakan untuk mengatur hak akses dalam pengelolaan basis data. Berdasarkan pengujian yang dilakukan, kecepatan respons sistem tercatat memiliki rata-rata 48 ms, dengan waktu respons tercepat sebesar 8 ms dan waktu respons paling lambat mencapai 136 ms. Hal ini menunjukkan bahwa sistem memiliki performa autentikasi yang cukup cepat dan stabil dalam mendukung keamanan pengelolaan basis data.

4.2.2 Pengujian CRUD

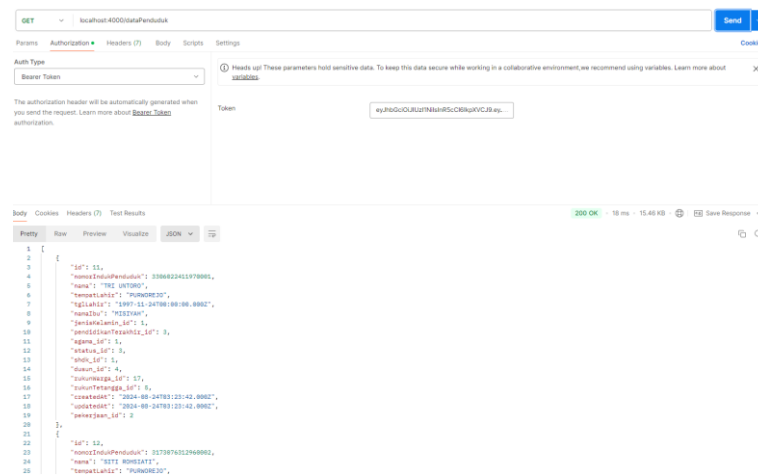
Pada tahap pengujian CRUD, peneliti menguji fungsi utama dalam pengelolaan basis data, yaitu Create, Read, Update, dan Delete. Fungsi *Create* digunakan untuk mencatat data baru ke dalam basis data dan dilakukan menggunakan operasi *POST* sesuai aturan HTTP. Fungsi *Read* berperan untuk membaca data 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 4.26.

Tabel 4.26 Tabel Pengujian CRUD

Id	Fungsi	Deskripsi Pengujian	Perkiraan Hasil	Hasil Pengujian	Kecepatan Respon	Kesimpulan
2A01	Create	Pengguna melakukan request operasi post sesuai dengan endpoint yang akan diakses, menyertakan data sesuai dengan field	Sistem akan mencatat data dan merespon dengan message “Data Created Succesfully”	Sistem mencatat data dan merespon dengan message “Data Created Succesfully”	115 ms	Berhasil

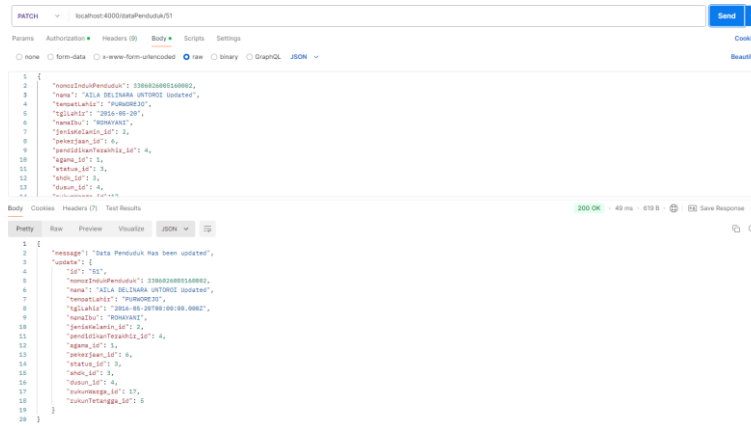
		yang disediakan JSON Web Token yang telah di dapat				
2B01	Read	Pengguna melakukan request operasi GET sesuai dengan endpoint yang akan diakses, dan menyertakan JSON Web Token yang telah di dapat	Sistem akan merespon dengan mengambil data yang diminta oleh pengguna	Sistem merespon dengan mengambil data yang diminta oleh pengguna	18 ms	Berhasil
2C01	Update	Pengguna melakukan request operasi PATCH sesuai dengan endpoint dan id data yang akan diubah, menyertakan data sesuai	Sistem akan merespon dengan message “Data Updated Succesfully ”	Sistem merespon dengan message “Data Updated Succesfully ”	49 ms	Berhasil

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. Pengujian dapat dilihat pada Gambar 4.15 dengan kecepatan respon 115 ms.



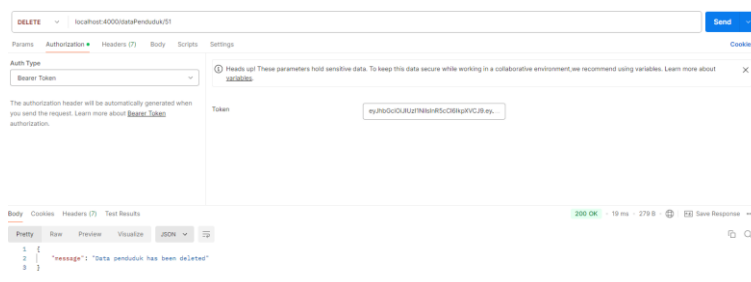
Gambar 4.16 Pengujian Read Data

Pengujian dengan ID 2B01 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 pengujian ini dapat dilihat pada Gambar 4.16 dengan kecepatan respon 18 ms.



Gambar 4.17 Pengujian Update Data

Pengujian dengan ID 2C01 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" dalam 49 ms sebagaimana ditampilkan pada Gambar 4.17.



Gambar 4.18 Pengujian Delete Data

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," dalam 19 ms sebagaimana terlihat pada Gambar 4.18.

Berdasarkan hasil pengujian terhadap empat fungsi utama dalam pengelolaan basis data, yaitu Create, Read, Update, dan Delete (CRUD), sistem berhasil menjalankan seluruh perintah sesuai dengan endpoint dan protokol yang telah ditentukan dengan baik. Kecepatan operasional dan respon sistem tercatat dengan rata-rata waktu 50 ms, dengan waktu tercepat 18 ms dan waktu terlama 115 ms. Hasil ini menunjukkan bahwa sistem mampu melaksanakan operasi CRUD dengan efisien dan efektif, serta dapat menangani berbagai macam permintaan dengan waktu respons yang relatif singkat.

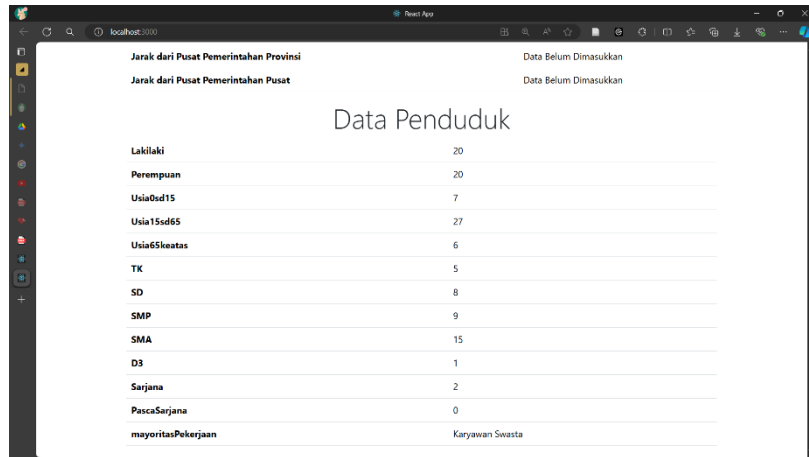
4.2.3 Pengujian Multi Platform

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. Daftar pengujian terdapat pada Tabel 4.27.

Tabel 4.27 Tabel Pengujian MultiPlatform

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.	Situs web akan menampilkan data yang diperoleh dari respons <i>server REST API</i> .	Situs web menampilkan data yang diperoleh dari respons <i>server REST API</i>	Berhasil

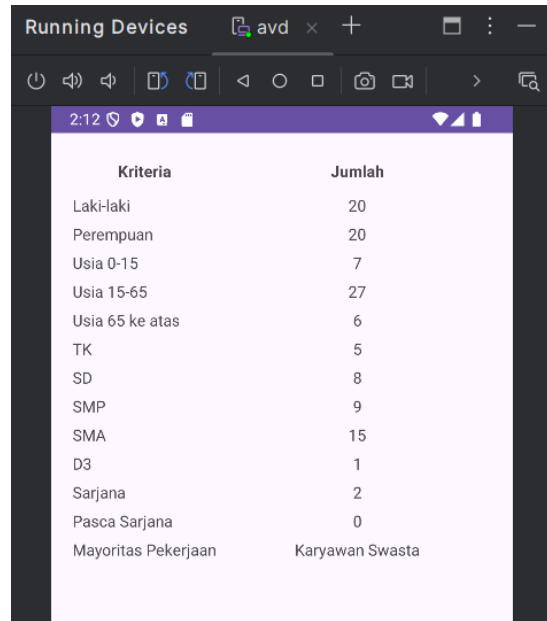
		Proses <i>fetching</i> data ini dilakukan dalam pengembangan kode situs web.			
3B01	Aksesibilitas dengan platform android	Pengguna melakukan <i>fetching</i> data ke platform Android melalui <i>endpoint</i> yang telah disediakan. Proses <i>fetching</i> data ini dilakukan dalam pengembangan kode aplikasi 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



Data Penduduk	
Laki-laki	20
Perempuan	20
Usia0sd15	7
Usia15sd65	27
Usia65keatas	6
TK	5
SD	8
SMP	9
SMA	15
D3	1
Sarjana	2
PascaSarjana	0
mayoritasPekerjaan	Karyawan Swasta

Gambar 4.19 Hasil Pengujian Platform Situs Web

Pengujian pengambilan data dari basis data terpadu monografi dengan ID 3A01 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 4.19.



The screenshot shows an Android emulator interface with a status bar at the top displaying the time 2:12 and various icons. Below the status bar is a table with two columns: 'Kriteria' and 'Jumlah'. The table contains the following data:

Kriteria	Jumlah
Laki-laki	20
Perempuan	20
Usia 0-15	7
Usia 15-65	27
Usia 65 ke atas	6
TK	5
SD	8
SMP	9
SMA	15
D3	1
Sarjana	2
Pasca Sarjana	0
Mayoritas Pekerjaan	Karyawan Swasta

Gambar 4.20 Hasil Pengujian Platform Aplikasi Android

Pengujian 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 perminataan 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. Hasil pengujian terdapat pada Gambar 4.20

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.

4.2.4 Pengujian Syntax

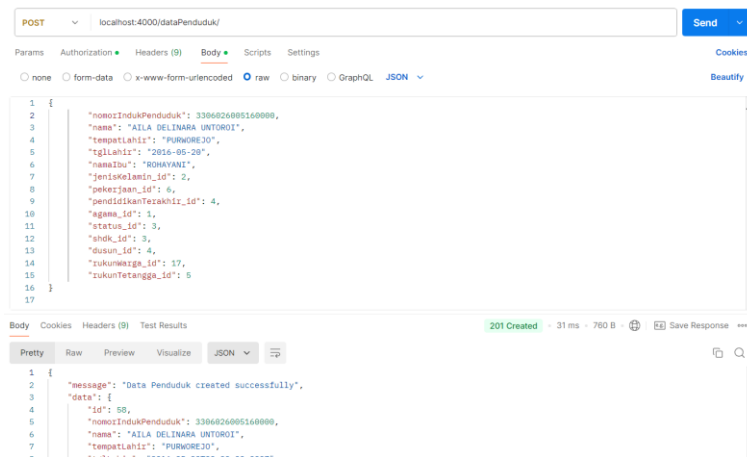
Pengujian sintaks dalam penelitian ini dilakukan untuk memastikan bahwa sistem dapat menangani format input yang tidak sesuai tanpa mengalami gangguan atau crash. Format utama yang digunakan dalam sistem ini adalah JSON, dan untuk menguji keabsahan serta ketahanan sistem terhadap kesalahan format, peneliti melakukan tiga jenis pengujian. Pertama, dengan memberikan input dalam bentuk JSON Object, kemudian JSON Array, dan akhirnya format

non-JSON seperti XML. Pengujian ini dilakukan menggunakan alat bantu Postman untuk memastikan setiap skenario pengujian dapat divalidasi secara akurat. Tujuan utama dari pengujian ini adalah untuk memverifikasi bahwa sistem memiliki ketahanan terhadap masukan data yang tidak sesuai dengan format yang ditetapkan, sehingga menghindari potensi kesalahan atau kegagalan sistem yang mungkin muncul akibat input yang tidak valid. Hasil pengujian tersebut dapat dilihat pada Tabel 4.28 sebagai referensi lebih lanjut mengenai performa sistem dalam menangani berbagai format input.

Tabel 4.28 Pengujian Syntax

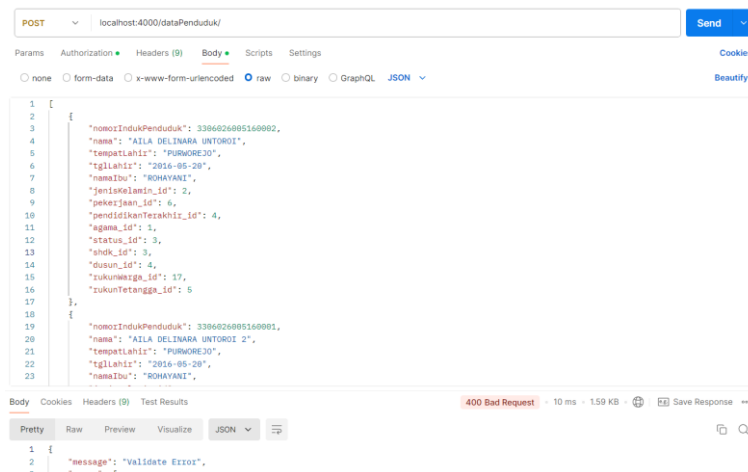
Id	Deskripsi Pengujian	Perkiraan Hasil	Hasil Pengujian	Kecepatan Respon	Kesimpulan
4A01	Pengguna melakukan request operasi post sesuai dengan endpoint yang akan di akses, menyertakan data dengan format JSON Object sesuai dengan field yang disediakan JSON Web Token yang telah di dapat	Sistem akan mencatat data dan merespon dengan message “Data Created Succesfully”	Sistem mencatat data dan merespon dengan message “Data Created Succesfully”	31 ms	Berhasil
4A02	Pengguna melakukan request operasi post sesuai dengan endpoint yang	Sistem akan mencatat data dan merespon dengan message	Sistem merespon dengan kode error dan message	10 ms	Gagal

	akan di akses, menyertakan data dengan format JSON Array sesuai degan field yang disediakan JSON Web Token yang telah di dapat	“Data Created Succesfully”	“Validate Error”		
4A03	Pengguna melakukan request operasi post sesuai dengan endpoint yang akan di akses, menyertakan data dengan format XML sesuai degan field yang disediakan JSON Web Token yang telah di dapat	Sistem akan merespon dengan kode error	Sistem merespon dengan kode error	7 ms	Berhasil



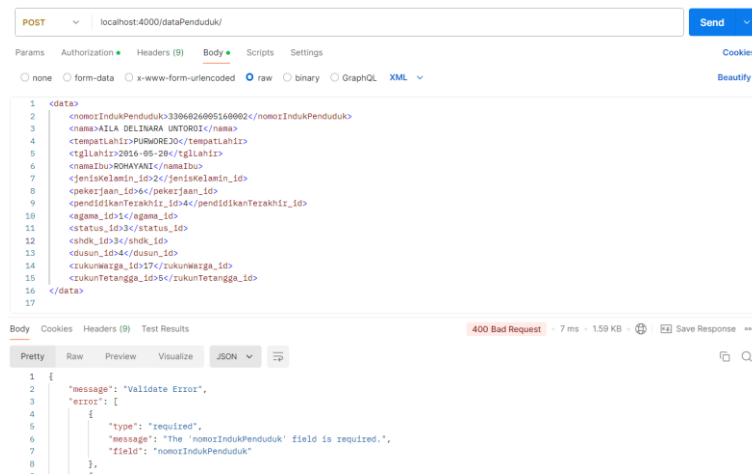
Gambar 4.21 Pengujian Format JSON Object

Pengujian dengan ID 4A01 dilakukan melalui salah satu *endpoint* yang telah disediakan, yaitu `localhost:4000/dataPenduduk`, dengan operasi *POST*. Pada bagian *body*, peneliti memasukkan data dengan format JSON Object 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. Berdasarkan hasil pada Gambar 4.21 Pengujian Format JSON Object, sistem menerima request tersebut dan dapat data dalam format JSON Object ke dalam basis data. Proses request dan respon terjadi selama 31 ms.



Gambar 4.22 Pengujian Format JSON Array

Pengujian dengan ID 4A02 dilakukan melalui salah satu *endpoint* yang telah disediakan, yaitu `localhost:4000/dataPenduduk`, dengan operasi *POST*. Pada bagian *body*, peneliti memasukkan data dengan format JSON Array yang terdiri dari dua object 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. Berdasarkan hasil pada Gambar 4.22 Pengujian Format JSON Array, sistem gagal untuk memproses data input dengan format JSON Array dan mengembalikan respon dengan pesan "validate error, proses request dan respon terjadi selama 10 ms.



Gambar 4.23 Pengujian Format XML

Pengujian dengan ID 4A03 dilakukan melalui salah satu *endpoint* yang telah disediakan, yaitu `localhost:4000/dataPenduduk`, dengan operasi *POST*. Pada bagian *body*, peneliti memasukkan data dengan format XML 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*. Dikarenakan sistem ini tidak mendukung input dengan format XML maka sistem akan merespon dengan kode error. Berdasarkan hasil pengujian pada Gambar 4.23 Pengujian Format XML, sistem merespon request dengan kode error 400 dengan rentan waktu 7 ms.

Berdasarkan hasil pengujian terhadap sintaks yang telah dilakukan, sistem berhasil melaksanakan operasi dengan tingkat keberhasilan sebesar 66%, sesuai dengan harapan yang telah ditetapkan. Pengujian ini mengungkapkan bahwa sistem hanya menerima input dalam format JSON Object, sementara format input lainnya memerlukan pengembangan lebih lanjut agar dapat didukung. Selanjutnya, pengujian menunjukkan bahwa kecepatan respon sistem tercatat dengan rata-rata waktu 16 ms, waktu tercepat 7 ms, dan waktu terlama 31 ms. Dengan demikian, dapat disimpulkan bahwa sistem mampu menangani pengujian sintaks dengan cukup efektif, meskipun terdapat batasan terkait format input yang diterima.

4.2.5 Boundery Value Analisys

Boundery Value Analisys merupakan bagian dari metode black box yang melibatkan pengujian dalam rentang batas atas dan bawah dari suatu nilai yang dimasukkan ke dalam sistem (Ijudin & Saifudin, 2020). Teknik ini merupakan metode yang efektif untuk menguji kecacatan maupun kelemahan dalam sistem (Ijudin & Saifudin, 2020). Dalam penelitian ini, teknik pengujian *Boundary Value Analysis* diterapkan pada operasi POST yang melibatkan

input nilai JSON ke dalam sistem. Pengujian dilakukan pada endpoint /penduduk untuk menjalankan operasi POST, di mana teknik ini digunakan untuk menguji nilai batas (*boundary value*) pada atribut nomorIndukKependudukan, yang merupakan salah satu atribut terpenting dalam data kependudukan. Value yang telah ditetapkan pada field nomorIndukKependudukan adalah 16 digit integer dan bersifat unik (*UU No. 24 Tahun 2013, t.t.*). Dari kondisi tersebut kemudian field nomorIndukKependudukan akan diuji dalam beberapa batasan yaitu:

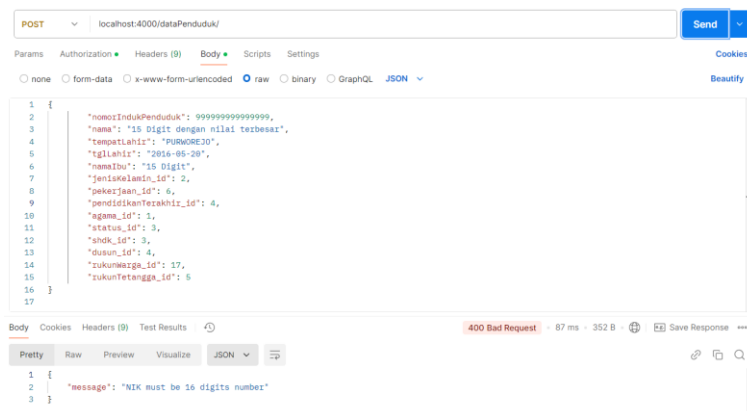
- Field nomorIndukKependudukan dengan nilai 15 digit integer dengan nilai yang terbesar.
- Field nomorIndukKependudukan dengan nilai 16 digit integer yang belum terdaftar dengan nilai terkecil.
- Field nomorIndukKependudukan dengan nilai 16 digit integer yang belum terdaftar dengan nilai terbesar.
- Field nomorIndukKependudukan dengan nilai 16 digit integer yang sudah terdaftar.
- Field nomorIndukKependudukan dengan nilai 17 digit integer dengan nilai terkecil.
- Field nomorIndukKependudukan dengan nilai 16 digit string.

Selama proses pengujian, JSON Web Token (JWT) disertakan dalam header menggunakan alat uji *Postman*. Detail dan hasil penelitian terdapat pada Tabel 4.29

Tabel 4.29 Pengujian Boundery Value Analisis

Id	Deskripsi Pengujian	Perkiraan Hasil	Hasil Pengujian	Kecepatan Respon	Kesimpulan
5A01	Field nomorIndukKependudukan diisi dengan 15 digit integer terbesar	Sistem akan merespon dengan kode error	Sistem merespon dengan kode error	87 ms	Berhasil
5A02	Field nomorIndukKependudukan diisi dengan 16 digit integer terkecil yang belum terdaftar.	Sistem akan mencatat data dan mengembalikan data dengan kode berhasil	Sistem mencatat data dan mengembalikan data dengan kode berhasil	164 ms	Berhasil
5A03	Field nomorIndukKependudukan diisi dengan 16 digit	Sistem akan mencatat data dan mengembalikan	Sistem merespon dengan kode error	66 ms	Gagal

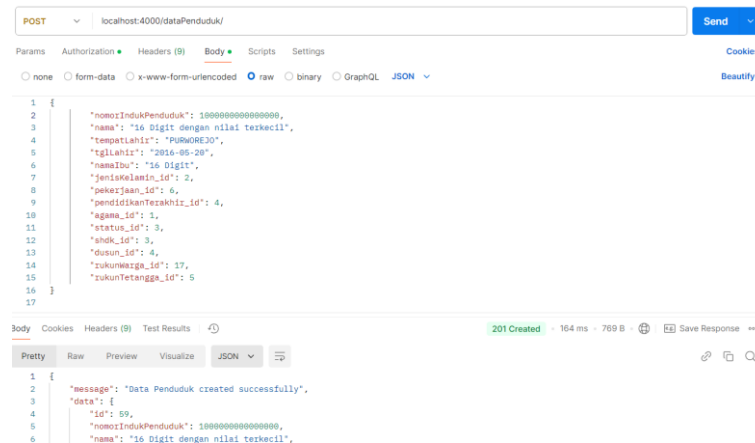
	integer terbesar yang belum terdaftar.	n data dengan kode berhasil			
5A04	Field nomorIndukKependudukan diisi dengan 16 digit integer yang sudah terdaftar.	Sistem akan merespon dengan kode error	Sistem merespon dengan kode error	14 ms	Berhasil
5A05	Field nomorIndukKependudukan diisi dengan 17 digit integer terkecil	Sistem akan merespon dengan kode error	Sistem merespon dengan kode error	9 ms	Berhasil
5A06	Field nomorIndukKependudukan diisi dengan 16 digit string	Sistem akan merespon dengan kode error	Sistem merespon dengan kode error	8 ms	Berhasil
5A07	Field nomorIndukKependudukan diisi dengan 16 digit Char	Sistem akan merespon dengan kode error	Sistem merespon dengan kode error	19 ms	Berhasil



Gambar 4.24 Pengujian Nilai 15 Digit Integer

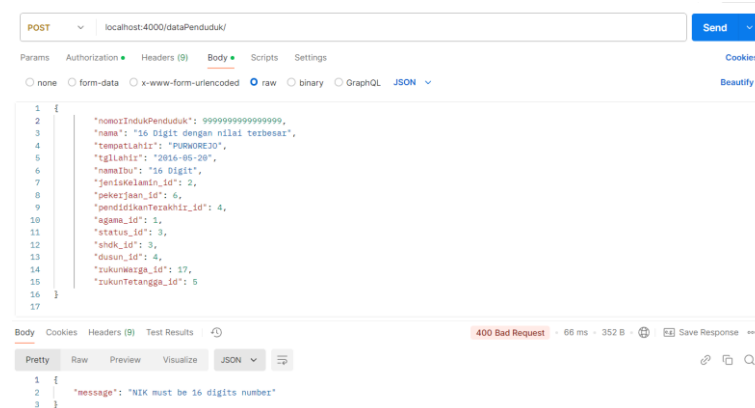
Pengujian dengan ID 5A01 dilakukan untuk menguji nilai input atribut nomorIndukKependudukan menggunakan nilai integer sebanyak 15 digit. Penentuan nilai input ini merupakan penerapan teknik analisis nilai batas (boundary value analysis), di mana batas maksimal untuk nomor induk penduduk yang seharusnya terdiri dari 16 digit adalah

nomor dengan 15 digit dan nilai tertinggi, yaitu "999999999999999". Berdasarkan hasil yang ditampilkan pada Gambar 4.24, sistem merespons dengan pesan "NIK must be 16 digits number" sesuai dengan yang diharapkan. Proses permintaan dan respons berlangsung dalam waktu 87 ms.



Gambar 4.25 Pengujian Nilai 16 Digit Unik Terkecil

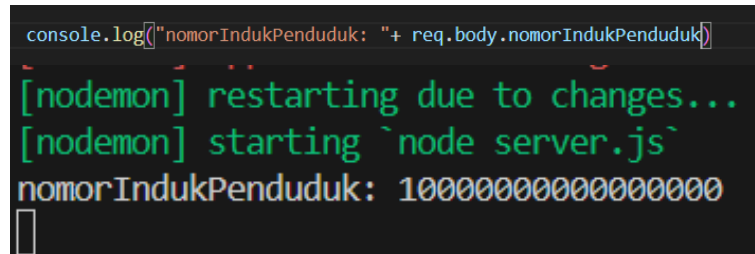
Pengujian dengan ID 5A02 dilakukan untuk menguji nilai input pada atribut nomorIndukKependudukan dengan nilai integer yang terdiri dari 16 digit. Penentuan nilai input ini merupakan penerapan teknik analisis batas nilai (boundary value analysis), di mana batasan nomor induk kependudukan adalah 16 digit, dan 16 digit nilai terkecil yang dapat diterima, yaitu "1000000000000000". Berdasarkan hasil yang ditampilkan pada Gambar 4.25, terlihat bahwa sistem dapat mencatat sesuai dengan yang diharapkan. Proses permintaan dan respons berlangsung dalam waktu 164 ms.



Gambar 4.26 Pengujian Nilai 16 Digit Unik Terbesar

Pengujian dengan ID 5A03 dilakukan untuk menguji nilai input pada atribut nomorIndukKependudukan dengan nilai integer yang terdiri dari 16 digit. Penentuan nilai input ini merupakan penerapan teknik analisis batas nilai (boundary value analysis), di mana batasan nomor induk kependudukan adalah 16 digit, dan 16 digit nilai terbesar yang dapat diterima,

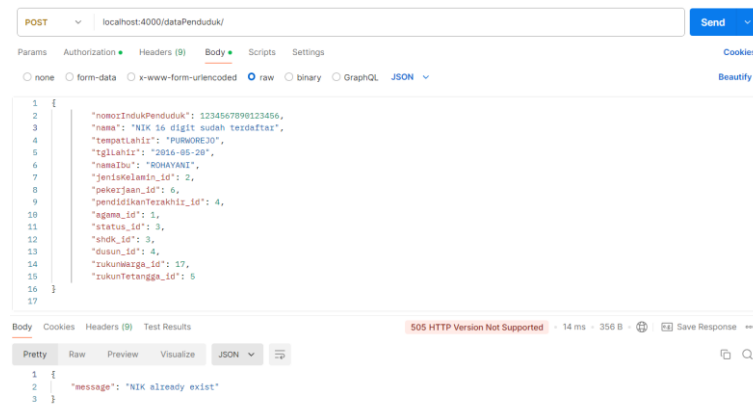
yaitu "9999999999999999". Berdasarkan hasil yang ditampilkan pada Gambar 4.26, terlihat bahwa sistem merespon dengan pesan "NIK must be 16 digits number" dalam waktu 66 ms. Hal tersebut tidak sesuai dengan perkiraan hasil yang sudah ditetapkan, yang seharusnya sistem dapat mencatat data dengan baik apabila dimasukkan data 16 digit terbesar.



```
console.log("nomorIndukPenduduk: " + req.body.nomorIndukPenduduk)
[nodemon] restarting due to changes...
[nodemon] starting `node server.js`
nomorIndukPenduduk: 1000000000000000
```

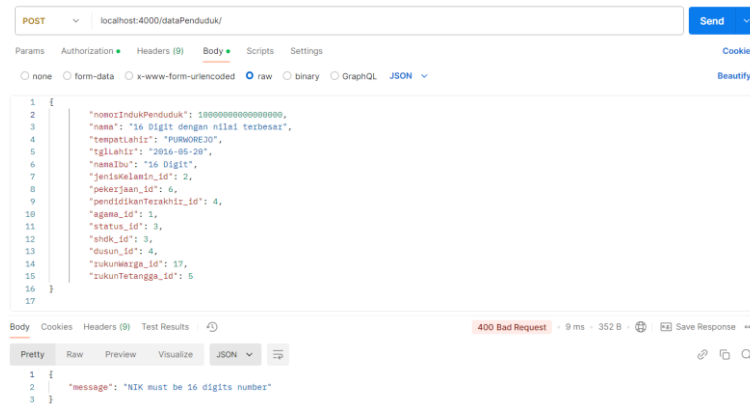
Gambar 4.27 Debugging Nilai 16 Digit Unik Terbesar

Pada tahap ini, peneliti melakukan proses debugging untuk mengidentifikasi sumber kesalahan. Seperti yang terlihat pada Gambar 4.27, nilai input mengalami perubahan dalam sistem. Perubahan ini disebabkan oleh kapasitas maksimal dari tipe data yang digunakan pada atribut nomorIndukPenduduk, yaitu BigInt, yang hanya mampu menangani nilai hingga $2^{53} - 1$ atau 9.007.199.254.740.991. Akibatnya, nilai yang diinputkan berubah menjadi "1000000000000000" dan tidak dapat disimpan, karena melebihi batas maksimum 16 digit.



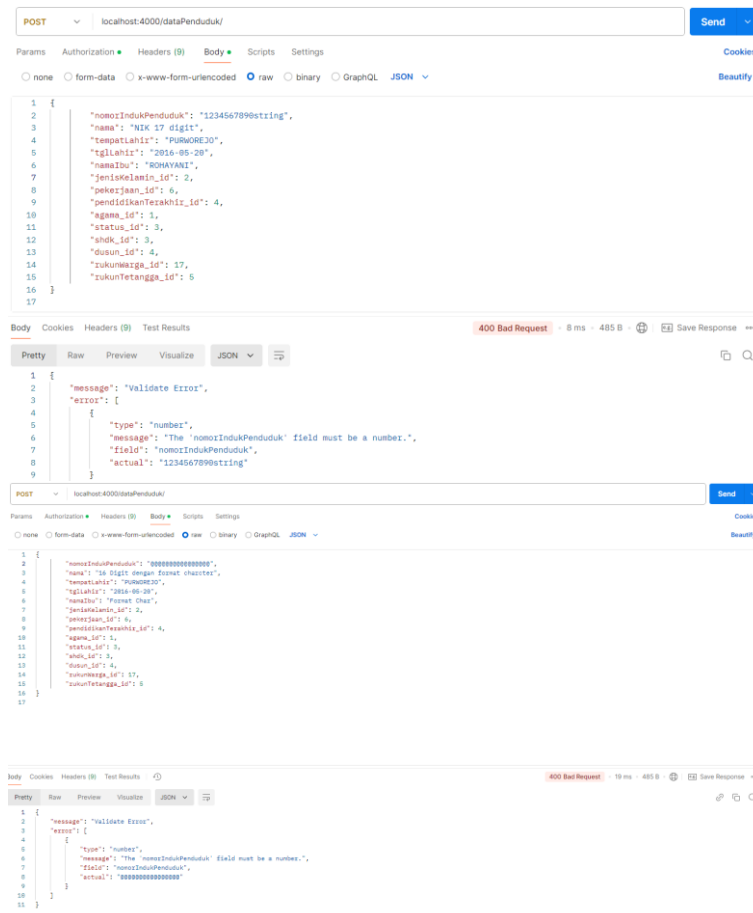
Gambar 4.28 Pengujian Nilai 16 Digit Tidak Unik

Pengujian dengan ID 5A04 dilakukan untuk menguji nilai input pada atribut nomorIndukKependudukan yang unik. Penentuan nilai input ini merupakan penerapan teknik analisis batas nilai (boundary value analysis), di mana batasan nomor induk kependudukan 16 digit yang unik adalah 16 digit nomorIndukKependudukan yang sudah terdaftar sebelumnya. Berdasarkan hasil yang ditampilkan pada Gambar 4.28, terlihat bahwa sistem merespon dengan pesan "NIK already exist" dalam waktu 14 ms sesuai dengan yang diharapkan.



Gambar 4.29 Pengujian Nilai 17 Digit Terkecil

Pengujian dengan ID 5A05 dilakukan untuk menguji nilai input pada atribut nomorIndukKependudukan dengan nilai integer yang terdiri dari 16 digit. Penentuan nilai input ini merupakan penerapan teknik analisis batas nilai (boundary value analysis), di mana batas minimal untuk nomor induk penduduk yang seharusnya terdiri dari 16 digit adalah nomor dengan 17 digit dan nilai terendah, yaitu "100000000000000000".. Berdasarkan hasil yang ditampilkan pada Gambar 4.29, sistem merespons dengan pesan "NIK must be 16 digits number" sesuai dengan yang diharapkan. Proses permintaan dan respons berlangsung dalam waktu 9 ms.



Gambar 4.30 Pengujian Nilai Format Tidak Integer

Pengujian dengan ID 5A06 dan 5A07 dilakukan untuk menguji nilai input pada atribut nomorIndukKependudukan dengan nilai integer yang terdiri dari 16 digit. Penentuan nilai input ini merupakan penerapan teknik analisis batas nilai (boundary value analysis), di mana batas minimal untuk nomor induk penduduk merupakan data dengan tipe integer. Dan pengujian batas dilakukan dengan data dalam format selain integer yaitu string dan Char. Berdasarkan hasil yang ditampilkan pada Gambar 4.30, sistem merespons dengan pesan "validate error" sesuai dengan yang diharapkan. Proses permintaan dan respons berlangsung dalam waktu 8 ms dan 19 ms.

Berdasarkan pengujian dengan teknik *boundary value analysis* yang dilakukan pada field nomorIndukKependudukan, diperoleh tingkat keberhasilan pengujian sebesar 85%. Sistem mencatat kecepatan respons rata-rata sebesar 52 ms, dengan waktu tercepat 8 ms dan waktu terlama 164 ms. Namun, terdapat kesalahan pada pengujian dengan ID 5A03, yang disebabkan oleh limitasi tipe nilai atribut. Berdasarkan tingkat keberhasilan ini, dapat disimpulkan bahwa sistem mampu berfungsi secara efektif sesuai dengan harapan pengembang, meskipun masih terdapat kesalahan yang perlu diperbaiki.

4.2.6 Equivalent Partitioning

Equivalent partitioning merupakan teknik pengujian dalam metode black box yang menguji data input dalam setiap form (Melani & Mahmud, 2021). Menurut Melani, pengujian ini membagi kelas data yang mewakili kelompok input tertentu. Teknik pengujian ini dapat menemukan bug logika dalam DBMS dengan lebih efektif (Rigger & Su, 2020). Dalam penelitian ini, teknik *equivalent partitioning* digunakan untuk menguji logika basis data relasional dalam REST API yang telah dikembangkan. Pengujian dilakukan pada endpoint /penduduk guna memverifikasi logika atribut yang mencerminkan penggunaan logika relasional basis data. Di dalam endpoint ini terdapat atribut agama_id, yang memiliki relasi dengan tabel agama. Nilai agama_id ditetapkan berdasarkan ketersediaan ID dalam tabel agama.

Data input dalam pengujian kemudian dikelompokkan menjadi dua partisi yaitu :

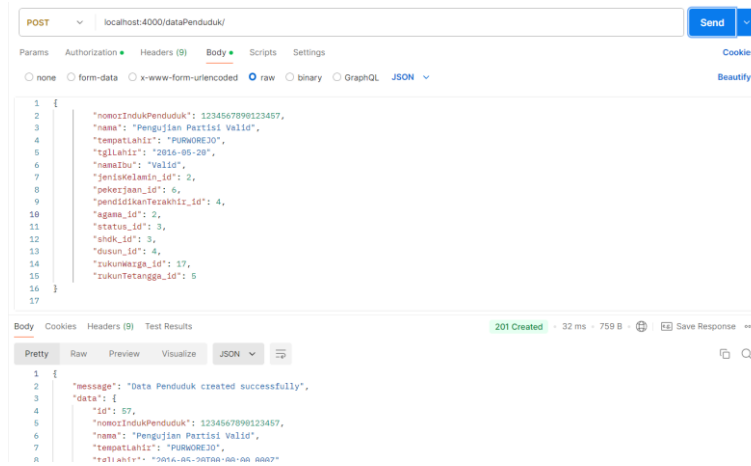
- a. Valid : nilai atribut agama_id telah terdaftar sebagai id pada tabel agama.
- b. Tidak Valid : nilai atribut agama_id tidak terdaftar sebagai id pada tabel agama.

Atribut dan nilai yang ada pada tabel agama dapat dilihat pada Tabel 4.5. Pengujian ini dilakukan menggunakan *Postman* sebagai alat uji, dengan menyertakan JSON Web Token pada bagian *header*. Rincian dan hasil pengujian dapat dilihat pada Tabel 4.30.

Tabel 4.30 Pengujian Equivalent Partitioning

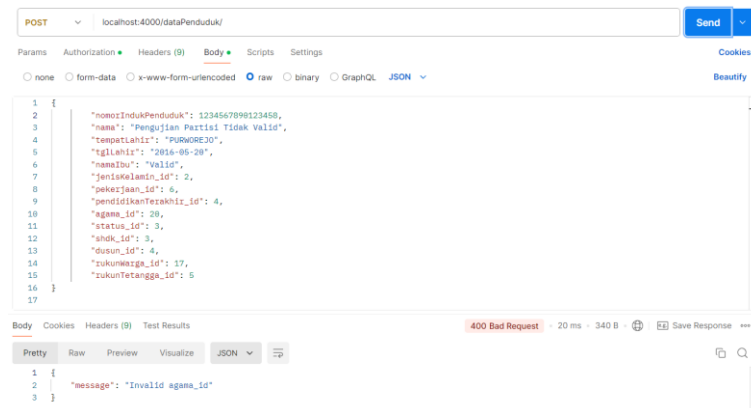
Id	Deskripsi Pengujian	Perkiraan Hasil	Hasil Pengujian	Kecepatan Respon	Kesimpulan
6A01	Field agama_id diisi dengan perwakilan nilai yang valid, yaitu “2” dengan format integer.	Sistem akan mencatat data dan mengembalikan data dengan kode berhasil	Sistem akan mencatat data dan mengembalikan data dengan kode berhasil	32 ms	Berhasil
6A02	Field agama_id diisi dengan perwakilan nilai yang tidak valid, yaitu	Sistem akan merespon dengan kode error dan message	Sistem akan merespon dengan kode error dan message	20 ms	Berhasil

	“20” dengan format integer.	“Invalid agama_id”	“Invalid agama_id”		
--	-----------------------------	--------------------	--------------------	--	--



Gambar 4.31 Pengujian Partisi Valid

Pengujian 6A01 dilakukan untuk menguji fungsionalitas sistem dalam memvalidasi nilai atribut agama_id yang terdaftar pada tabel agama. Dalam pengujian ini, digunakan partisi valid dengan memilih nilai "2", yang merujuk pada ID agama Kristen dalam Tabel 4.5 yang berisi daftar agama. Sistem kemudian memvalidasi apakah nilai tersebut terdapat dalam tabel agama. Berdasarkan hasil yang ditampilkan pada Gambar 4.31, sistem berhasil memvalidasi nilai atribut agama dan mencatat data tersebut ke dalam basis data. Proses permintaan hingga respons berlangsung dalam waktu 32 ms.



Gambar 4.32 Pengujian Partisi Tidak Valid

Pengujian 6A02 dilakukan untuk menguji fungsionalitas sistem dalam memvalidasi nilai atribut agama_id yang terdaftar pada tabel agama. Dalam pengujian ini, digunakan partisi valid dengan memilih nilai "20", yang tidak terdaftar dalam Gambar 4.32 yang berisi daftar agama. Sistem kemudian memvalidasi apakah nilai tersebut terdapat dalam tabel agama. Berdasarkan

hasil yang ditampilkan pada Gambar 4.32, sistem berhasil memvalidasi nilai atribut agama dan memberikan respon “Invalid agama_id”. Proses permintaan hingga respons berlangsung dalam waktu 32 ms.

Berdasarkan hasil pengujian menggunakan metode *equivalent partitioning* pada logika DBMS yang diterapkan pada tabel data penduduk dan agama, sistem berhasil menjalankan seluruh operasi yang diminta dengan sukses. Rata-rata waktu yang dicatat sistem untuk menjalankan operasi dan fungsi adalah 26 ms. Hal ini menunjukkan bahwa pengujian *equivalent partitioning* membuktikan kemampuan sistem dalam menjalankan logika DBMS secara efektif dan efisien.

BAB V

PENUTUP

5.1 Kesimpulan

Penelitian ini bertujuan mengembangkan basis data monografi desa yang dapat terintegrasi dengan aman menggunakan autentifikasi dan otorisasi ke berbagai platform. Dengan membangun basis data monografi desa dengan basis REST API dan keamanan JSON Web Token diharapkan sistem ini menciptakan konsistensi informasi di seluruh platform publikasi yang berkaitan dengan desa. Selain itu, sistem ini diharapkan dapat membantu pemerintah desa dalam mengelola data secara terpusat. Pengembangan sistem dilakukan menggunakan metode prototipe, yang melibatkan diskusi antara peneliti, pemerintah desa, dan pengembang IT untuk mengidentifikasi masalah dan merancang sistem. Rancangan ini mencakup perancangan tabel, Entity Relationship Diagram (ERD), arsitektur sistem, dan endpoint API. Implementasi dilakukan menggunakan framework Node.js dan ORM Sequelize untuk pengembangan basis data, yang kemudian diuji melalui metode black box untuk memastikan fungsionalitasnya sesuai dengan kebutuhan klien.

Berdasarkan hasil pengujian yang telah dilakukan, sistem diuji dengan mengacu pada sejumlah aspek penting, antara lain autentikasi, operasi CRUD (Create, Read, Update, Delete), sintaksis, kompatibilitas dengan multiplatform, serta logika Database Management System (DBMS). Hasil pengujian menunjukkan bahwa sistem berhasil melewati 92% dari seluruh rangkaian pengujian dengan tingkat kesalahan sebesar 8%, yang mengindikasikan tingkat keandalan yang cukup tinggi. Selain itu, sistem mampu menunjukkan kinerja yang optimal dalam hal kecepatan operasional, dengan rata-rata waktu respons hanya 38 ms. Melalui pemanfaatan sistem ini, pemerintah desa memiliki kemampuan untuk mempublikasikan informasi yang tercatat dalam monografi desa dengan aman, efisiensi dalam pengelolaan data, dan efektivitas dalam penyebaran informasi ke berbagai platform. Hal ini tidak hanya mendukung transparansi pengelolaan data desa, tetapi juga memperluas jangkauan akses masyarakat terhadap informasi yang relevan. Dengan demikian, sistem ini berkontribusi secara signifikan dalam memperkuat komunikasi antara pemerintah desa dan masyarakat, mendorong partisipasi aktif warga, serta meningkatkan akuntabilitas pengelolaan informasi yang terkait dengan pembangunan dan layanan di tingkat desa.

5.2 Saran

Berdasarkan penelitian yang sudah dilakukan peneliti memberikan beberapa saran yaitu:

a. Pengembangan Keamanan

Pengembangan fungsionalitas JSON Web Token (JWT) untuk menjaga keamanan hak akses dalam pengelolaan data melalui REST API memungkinkan penerapan fitur peran (Role) bagi para pemangku kepentingan di pemerintahan dan pengembang TI secara umum. Dengan fitur peran ini, pengelolaan data hanya dapat dilakukan oleh pemilik hak akses, sehingga dapat meningkatkan keamanan.

b. Pengembangan Efektifitas

Dalam implementasi yang dilakukan oleh peneliti, proses penginputan data saat ini terbatas pada satu atribut dengan satu nilai. Untuk meningkatkan efisiensi dan efektivitas, pengembangan di masa depan dapat mempertimbangkan untuk menggunakan JSON array dalam proses penginputan. Dengan penerapan JSON array, diharapkan pengguna dapat memasukkan beberapa data sekaligus dalam satu waktu, sehingga mempercepat proses pengumpulan dan pengelolaan data.

c. Perubahan Tipe Atribut

Pada pengujian yang telah dilakukan, ditemukan satu celah berupa bug pada sistem dalam pencatatan nomor induk penduduk. Tipe data yang digunakan untuk atribut nomorIndukPenduduk saat ini adalah *bigint*, yang memiliki batas maksimum hingga 2^{53} . Hal ini menyebabkan nilai nomor induk penduduk yang terdiri dari 16 digit melebihi batas kapasitas tipe data tersebut. Mengingat bahwa nomorIndukPenduduk tidak bergantung pada fungsi matematis apapun, disarankan untuk mengubah tipe data atribut ini menjadi *string*. Dengan penggunaan tipe data *string*, fungsi dari nomor induk penduduk tetap terjaga, dan sistem dapat menyimpan nilai hingga 16 digit terbesar tanpa kendala.

DAFTAR PUSTAKA

- Ahmad, I., Suwarni, E., Rossi, F., Rohmat, I., Borman, Jusman, Y., Indra, R., & Asmawati, A. (2022). *Implementation of RESTful API Web Services Architecture in Takeaway Application Development*.
- Ardiansyah, A., Yahya, F. Y., Irawati, A. R., & Yusman, M. (2021). PENGEMBANGAN SISTEM INFORMASI TERPADU FMIPA UNIVERSITAS LAMPUNG (SIMIPA) MENGGUNAKAN METODE SCRUM. *Jurnal Teknoinfo*, 15(2), Article 2. <https://doi.org/10.33365/jti.v15i2.1041>
- Armas, J., Navas, P., Mayorga, T., Rengifo, P., & Arevalo, B. (2017). Optimization of code lines and time of access to information through object-relational mapping (ORM) using alternative tools of connection to database management systems (DBMS). *2017 2nd International Conference on System Reliability and Safety (ICSRS)*, 500–504. 2017 2nd International Conference on System Reliability and Safety (ICSRS). <https://doi.org/10.1109/ICSRS.2017.8272872>
- Astowo, U. B., & Sujarwo, A. (2023). Penerapan JSON Web Token sebagai Strategi Pengamanan Data pada Aplikasi MultiMasjid. *Innovative: Journal Of Social Science Research*, 3(6), Article 6. <https://doi.org/10.31004/innovative.v3i6.6908>
- Atmojo, M. E., Zuhriyati, E., & Hanif, N. A. (2022). Peningkatan Kualitas Pemerintah Desa Melalui Penyusunan Profil Desa Trimulyo. *Society : Jurnal Pengabdian Masyarakat*, 1(3), Article 3. <https://doi.org/10.55824/jpm.v1i3.88>
- Azkarin, V., Guntara, R. G., & Herdiana, O. (2023). Development of a REST API for Human Resource Information System for Employee Referral Management Domain Using the Express JS Framework and Node.js. *Journal of Scientific Research, Education, and Technology (JSRET)*, 2(3), Article 3. <https://doi.org/10.58526/jsret.v2i3.199>

- Bourhis, P., Reutter, J. L., Suárez, F., & Vrgoč, D. (2017). JSON: Data model, Query languages and Schema specification. *Proceedings of the 36th ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems*, 123–135.
<https://doi.org/10.1145/3034786.3056120>
- Brady, M., & Loonam, J. (2010). Exploring the use of entity-relationship diagramming as a technique to support grounded theory inquiry. *Qualitative Research in Organizations and Management: An International Journal*, 5, 224–237.
<https://doi.org/10.1108/17465641011089854>
- Brew, A. (2003). *Pengantar JavaScript*.
https://www.academia.edu/3279350/Pengantar_JavaScript
- Crow's Foot Notation*. (2016, Maret 31). Vertabelo Data Modeler.
<https://vertabelo.com/blog/crow-s-foot-notation/>
- Darnis, F., & Azdy, R. A. (2018). Perancangan Basis Data Website Promosi Usaha Kecil Dan Menengah (UKM) Desa Pedado. *Sinkron : Jurnal Dan Penelitian Teknik Informatika*, 3(1), Article 1.
- Dwita Mentari. (2014). *IMPLEMENTASI NORMALISASI DATABASE AKADEMIK TERHADAP MySQL*.
- Elmasri, R., & Navathe, S. (2011). *Fundamentals of database systems* (6th ed). Addison-Wesley.
- Express—Node.js web application framework*. (t.t.). Diambil 21 Oktober 2024, dari
<https://expressjs.com>
- Fathansyah. (2012). *Basis Data*. Informatika Bandung.
- Febriansyah, F. D. K., A, L. R., Alfianto, A., Lathoif, M. I., & Adilah, N. (2022).
 PERANCANGAN SISTEM INFORMASI E-ARSIP BERBASIS WEBSITE PADA

- DESA WATESWINANGUN. *ABDI MASSA: Jurnal Pengabdian Nasional* (e-ISSN: 2797-0493), 2(06), Article 06.
- Flanagan, D. (2012). *Java Script: O guia essencial*. Bookman.
- Groppe, S., & Groppe, J. (2024). *Hybrid Multi-model Multi-platform (HM3P) Databases*. 177–184. <https://www.scitepress.org/Link.aspx?doi=10.5220/0009802401770184>
- Gunawan, R., & Rahmatulloh, A. (2019). JSON Web Token (JWT) untuk Authentication pada Interoperabilitas Arsitektur berbasis RESTful Web Service. *JEPIN (Jurnal Edukasi dan Penelitian Informatika)*, 5(1), 74–79. <https://doi.org/10.26418/jp.v5i1.27232>
- Harrington, J. L. (2016). Entities and Relationships. Dalam *Relational Database Design and Implementation* (hlm. 55–88). Elsevier. <https://doi.org/10.1016/B978-0-12-804399-8.00004-1>
- Haryadi, H. L., Sujjada, A., & Simatupang, D. S. (2023). Perbandingan Rest Api Menggunakan Node Js Dan Php Pada Aplikasi Pemilihan Umum. *Jurasik (Jurnal Riset Sistem Informasi Dan Teknik Informatika)*, 8(2), Article 2. <https://doi.org/10.30645/jurasik.v8i2.631>
- Ichwani, A., Anwar, N., Karsono, K., & Alrifqi, M. (2021). Sistem Informasi Penjualan Berbasis Website dengan Pendekatan Metode Prototype. *Prosiding SISFOTEK*, 5(1), Article 1.
- Ijudin, A., & Saifudin, A. (2020). Pengujian Black Box pada Aplikasi Berita Online dengan Menggunakan Metode Boundary Value Analysis. *Jurnal Informatika Universitas Pamulang*, 5(1), Article 1. <https://doi.org/10.32493/informatika.v5i1.3717>
- Isnanda, Y. R. (t.t.). *PENGEMBANGAN APLIKASI KONVERSI OTOMATIS DATA JSON KE OWL ONTOLOGY*.
- Jaya, T. S. (2018). Pengujian Aplikasi dengan Metode Blackbox Testing Boundary Value Analysis (Studi Kasus: Kantor Digital Politeknik Negeri Lampung). *Jurnal*

- Informatika: Jurnal Pengembangan IT*, 3(1), Article 1.
<https://doi.org/10.30591/jpit.v3i1.647>
- Jayadi, P., & Nugraha, T. (2019). *Tutorial Dasar Framework Laravel*.
https://www.academia.edu/13433223/Tutorial_Dasar_Framework_Laravel
- Juneau, J., & Juneau, J. (2013). *Object-Relational Mapping*. 369–408.
https://doi.org/10.1007/978-1-4302-4426-4_8
- Kaprocki, Z., Pekovic, V., & Velikic, G. (2015). Combined testing approach: Increased efficiency of black box testing. *2015 IEEE 1st International Workshop on Consumer Electronics (CE WS)*, 76–78. <https://doi.org/10.1109/CEWS.2015.7867160>
- Kelvin. (2019). *Simulasi e-voting pemilu untuk penyandang tuna netra menggunakan google cloud speech berbasis raspberry pi (uji coba yayasan peduli kesejahteraan tuna netra)* [bachelorThesis, Fakultas Sains dan Teknologi Universitas Islam Negeri Syarif Hidayatullah Jakarta]. <https://repository.uinjkt.ac.id/dspace/handle/123456789/48647>
- Kementerian Komunikasi dan Informasi Republik Indonesia. (2003). *Panduan Penyelenggaraan Situs Web Pemerintah Daerah*. Kementerian Komunikasi dan Informasi Republik Indonesia.
- Kurniawan, I., Humaira, & Rozi, F. (2020). REST API Menggunakan NodeJS pada Aplikasi Transaksi Jasa Elektronik Berbasis Android. *JITSI : Jurnal Ilmiah Teknologi Sistem Informasi*, 1(4), Article 4.
- Lorenz, M., Hesse, G., & Rudolph, J.-P. (2024). *Object-relational Mapping Revised—A Guideline Review and Consolidation*. 157–168.
<https://www.scitepress.org/Link.aspx?doi=10.5220/0005974201570168>
- Mardan, A. (2014). Building Node.js REST API Servers with Express.js and Hapi. Dalam A. Mardan (Ed.), *Practical Node.js: Building Real-World Scalable Web Apps* (hlm. 173–194). Apress. https://doi.org/10.1007/978-1-4302-6596-2_8

- Masse, M. (2011). *REST API Design Rulebook: Designing Consistent RESTful Web Service Interfaces*. O'Reilly Media, Inc.
- Mazurova, O., Andrushchenko, M., & Shirokopetleva, M. (2023). ДОСЛІДЖЕННЯ МЕТОДІВ ПРОГРАМНОЇ РЕАЛІЗАЦІЇ COSMOS DB API НА ПЛАТФОРМИ .NET. *INNOVATIVE TECHNOLOGIES AND SCIENTIFIC SOLUTIONS FOR INDUSTRIES*, 2 (24), Article 2 (24). <https://doi.org/10.30837/ITSSI.2023.24.118>
- Melani, Y. I., & Mahmud. (2021). *Black Box Testing Using Equivalence Partition Method in Sintana Application*. 529–535. <https://doi.org/10.2991/ahe.k.210205.089>
- Migrations | Sequelize*. (2024, Agustus 3). <https://sequelize.org/docs/v6/other-topics/migrations/>
- Muhammad, I. R. D., & Paputungan, I. V. (2024). Development of Backend Server Based on REST API Architecture in E-Wallet Transfer System. *Jurnal Sains, Nalar, Dan Aplikasi Teknologi Informasi*, 3(2), Article 2. <https://doi.org/10.20885/snati.v3.i2.35>
- Nashikhuddin, A. Y., Karaman, J., & Litanianda, Y. (2023). 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*, 7(2), Article 2. <https://doi.org/10.46880/jmika.Vol7No2.pp239-246>
- NASUTION. (2022). *Implementasi Mongo Db, Express Js, React Js Dan Node Js (Mern) Pada Pengembangan Aplikasi Formulir, Kuis, Dan Survei Online*. <https://dspace.uui.ac.id/handle/123456789/38607>
- Nugraha, E., Ariyana, R., & Nurnawati, E. (2022). Uji BLACK BOX TES APLIKASI SOFTWARE DEVELOPMENT SYSTEM INFORMATION (SODEVI) PT. DIMATA SORA JAYATE MENGGUNAKAN KATALON STUDIO. *PROSIDING SNAST*, E60-65. <https://doi.org/10.34151/prosidingsnast.v8i1.4172>

- Nurnawati, E. K., & Ermawati, E. (2018). Design of Integrated Database on Mobile Information System: A Study of Yogyakarta Smart City App. *IOP Conference Series: Materials Science and Engineering*, 306, 012036. <https://doi.org/10.1088/1757-899X/306/1/012036>
- Painem, P., & Soetanto, H. (2020). Sistem Presensi Pegawai Berbasis Web Service Menggunakan Metode Restfull Dengan Keamanan JWT Dan Algoritma Haversine. *Fountain of Informatics Journal*, 5(3), Article 3. <https://doi.org/10.21111/fij.v5i3.4906>
- Penetapan Presiden Nomor 1 Tahun 1965—Pusat Data Hukumonline. (t.t.). Diambil 9 Agustus 2024, dari <https://www.hukumonline.com/pusatdata/detail/26390/penetapan-presiden-nomor-1-tahun-1965/>
- PERMENDAGRI No. 1 Tahun 2017. (t.t.). Database Peraturan | JDIH BPK. Diambil 12 Agustus 2024, dari <http://peraturan.bpk.go.id/Details/111304/permendagri-no-1-tahun-2017>
- Permendagri No. 13 Tahun 2012 Tentang Monografi Desa Dan Kelurahan. (t.t.). Database Peraturan Perundang-Undangan Indonesia - [PERATURAN.GO.ID]. Diambil 3 Juni 2024, dari <https://peraturan.go.id/id/permendagri-no-13-tahun-2012>
- PERMENDAGRI No. 18 Tahun 2018. (t.t.). Database Peraturan | JDIH BPK. Diambil 12 Agustus 2024, dari <http://peraturan.bpk.go.id/Details/143587/permendagri-no-18-tahun-2018>
- PERMENDAGRI No. 19 Tahun 2010. (t.t.). Database Peraturan | JDIH BPK. Diambil 12 Agustus 2024, dari <http://peraturan.bpk.go.id/Details/138576/permendagri-no-19-tahun-2010>
- PERMENDAGRI No. 20 Tahun 2018. (t.t.). Database Peraturan | JDIH BPK. Diambil 9 Agustus 2024, dari <http://peraturan.bpk.go.id/Details/139714/permendagri-no-20-tahun-2018>

- PERMENDAGRI No. 73 Tahun 2020*. (t.t.). Database Peraturan | JDIH BPK. Diambil 12 Agustus 2024, dari <http://peraturan.bpk.go.id/Details/163278/permendagri-no-73-tahun-2020>
- PERMENDAGRI No. 95 Tahun 2019*. (t.t.). Database Peraturan | JDIH BPK. Diambil 12 Agustus 2024, dari <http://peraturan.bpk.go.id/Details/138603/permendagri-no-95-tahun-2019>
- Permendikbudriset No. 8 Tahun 2024*. (t.t.). Database Peraturan | JDIH BPK. Diambil 12 Agustus 2024, dari <http://peraturan.bpk.go.id/Details/281843/permendikbudriset-no-8-tahun-2024>
- Pezoa, F., Reutter, J. L., Suarez, F., Ugarte, M., & Vrgoč, D. (2016). Foundations of JSON Schema. *Proceedings of the 25th International Conference on World Wide Web*, 263–273. <https://doi.org/10.1145/2872427.2883029>
- PP No. 43 Tahun 2014*. (t.t.). Database Peraturan | JDIH BPK. Diambil 9 Agustus 2024, dari <http://peraturan.bpk.go.id/Details/5482/pp-no-43-tahun-2014>
- Prayogi, A. A., Niswar, M., Indrabayu, & Rijal, M. (2020). Design and Implementation of REST API for Academic Information System. *IOP Conference Series: Materials Science and Engineering*, 875(1), 012047. <https://doi.org/10.1088/1757-899X/875/1/012047>
- Pulungan, S. M., Febrianti, R., Lestari, T., Gurning, N., & Fitriana, N. (2023). Analisis Teknik Entity-Relationship Diagram Dalam Perancangan Database. *Jurnal Ekonomi Manajemen Dan Bisnis (JEMB)*, 1(2), Article 2. <https://doi.org/10.47233/jemb.v1i2.533>
- Puspitasari, D., Rahmad, C., & Astiningrum, M. (2016). *Normalisasi Tabel Pada Basisdata Relasional*. 8.

- Qi, M., & Wang, J. (2021). Using the Internet of Things E-Government Platform to Optimize the Administrative Management Mode. *Wireless Communications and Mobile Computing*, 2021(1), 2224957. <https://doi.org/10.1155/2021/2224957>
- Rahmawati, D. E., & Rasyid, H. A. N. (2022). Pendampingan Penyusunan Buku Profil Desa. *JCOMENT (Journal of Community Empowerment)*, 3, 1–11.
- Ramakrishnan, R. (2023). Enhancing Data Management with MongoDB and its Rest API. *International Journal for Research in Applied Science and Engineering Technology*, 11(7), 189–193. <https://doi.org/10.22214/ijraset.2023.54593>
- Rigger, M., & Su, Z. (2020). Finding bugs in database systems via query partitioning. *OOPSLA 20 Artifact for “Finding Bugs in Database Systems via Query Partitioning,”* 4(OOPSLA), 211:1-211:30. <https://doi.org/10.1145/3428279>
- Rodriguez, C., Baez, M., Daniel, F., Casati, F., Trabucco, J., Canali, L., & Percannella, G. (2016). *REST APIs: A Large-Scale Analysis of Compliance with Principles and Best Practices*. 21–39. https://doi.org/10.1007/978-3-319-38791-8_2
- Septima, R., & Zulfa, I. (2020). Pelatihan Ilmu Teknologi Komputer Dalam Meningkatkan Kemampuan Administrasi Perangkat Desa Tan Saril. *JPKMI (Jurnal Pengabdian Kepada Masyarakat Indonesia)*, 1(1), Article 1. <https://doi.org/10.36596/jpkmi.v1i1.10>
- Sequelize v6 | Sequelize. (2024, Oktober 19). <https://sequelize.org/docs/v6/>
- Setiawan, A., & Purnamasari, A. I. (2020). Implementasi JSON Web Token Berbasis Algoritma SHA-512 untuk Otentikasi Aplikasi BatikKita. *Jurnal RESTI (Rekayasa Sistem Dan Teknologi Informasi)*, 4(6), Article 6. <https://doi.org/10.29207/resti.v4i6.2533>
- Setiyowati, & Siswanti, Sri. (2021). *Perancangan Basis Data dengan Menggunakan SQL Server Management Studio*.

- Shah, H., & Soomro, T. (2017). Node.js Challenges in Implementation. *Global Journal of Computer Science and Technology*, 17, 72–83.
- Storey, V. C. (1991). Relational database design based on the entity-relationship model. *Data & Knowledge Engineering*, 7(1), 47–83. [https://doi.org/10.1016/0169-023X\(91\)90033-T](https://doi.org/10.1016/0169-023X(91)90033-T)
- Suehring, S., & Valade, J. (2013). *PHP, MySQL, JavaScript & HTML5 All-in-One For Dummies*. John Wiley & Sons.
- Susniwati, S., & Zamili, M. (2022). Acceleration of One Indonesian Data through Collaborative Governance in Indonesia. *Publik (Jurnal Ilmu Administrasi)*, 11(2), 166–177. <https://doi.org/10.31314/pjia.11.2.166-177.2022>
- Syahbana, S. N., & Tejawati, A. (2018). *Sistem Informasi Geografis Kriminalitas Di Kota Samarinda*. 3(1).
- Syahputri, K., & Nasution, M. I. P. (2023). Peran Database Dalam Sistem Informasi Manajemen. *Jurnal Akuntansi Keuangan Dan Bisnis*, 1(2), Article 2.
- Thung, F., Kim, K., Zhang, T., Irsan, I. C., Widyasari, R., Yang, Z., & Lo, D. (2023). *APIHarvest: Harvesting API Information from Various Online Sources* (arXiv:2304.02514). arXiv. <https://doi.org/10.48550/arXiv.2304.02514>
- Turcotte, A., Aldrich, M. W., & Tip, F. (2023). reformulator: Automated Refactoring of the N+1 Problem in Database-Backed Applications. *Proceedings of the 37th IEEE/ACM International Conference on Automated Software Engineering*, 1–12. <https://doi.org/10.1145/3551349.3556911>
- UNDANG-UNDANG REPUBLIK INDONESIA NOMOR 6 TAHUN 2014 TENTANG DESA – *Laboratorium Politik dan Tata Pemerintahan*. (t.t.). Diambil 9 Agustus 2024, dari <https://lptp-fia.ub.ac.id/undang-undang-republik-indonesia-nomor-6-tahun-2014-tentang-desa/>

- UU No. 12 Tahun 2012*. (t.t.). Database Peraturan | JDIH BPK. Diambil 12 Agustus 2024, dari <http://peraturan.bpk.go.id/Details/39063/uu-no-12-tahun-2012>
- UU No. 24 Tahun 2007*. (t.t.). Database Peraturan | JDIH BPK. Diambil 12 Agustus 2024, dari <http://peraturan.bpk.go.id/Details/39901/uu-no-24-tahun-2007>
- UU No. 24 Tahun 2013*. (t.t.). Database Peraturan | JDIH BPK. Diambil 9 Agustus 2024, dari <http://peraturan.bpk.go.id/Details/38985/uu-no-24-tahun-2013>
- Vega-Gorgojo, G. (2022). CRAFTS: Configurable REST APIs for Triple Stores. *IEEE Access*, 10, 32426–32441. <https://doi.org/10.1109/ACCESS.2022.3160610>
- Waqiah, Y. N. (2018). SISTEM INFORMASI PEMERINTAHAN DESA PADA KANTOR DESA TENGGIR PANJI SITUBONDO MENGGUNAKAN PHP DAN MYSQL. *Jurnal Ilmiah Informatika*, 3(2), Article 2. <https://doi.org/10.35316/jimi.v3i2.645>
- Warman, N. S., Syamsir, S., Maldini, M., Nurhasanah, O., Oktariandani, N. R., & Syafikruzi, I. H. (2022). Implementasi Inovasi Kebijakan Dalam Penyelenggaraan Sistem Pemerintahan Berbasis Elektronik (SPBE) di Kota Pekanbaru. *PROSIDING SEMINAR NASIONAL PENDIDIKAN, BAHASA, SASTRA, SENI, DAN BUDAYA*, 1(2), Article 2. <https://doi.org/10.55606/mateandrau.v1i2.161>
- Wicasana, B. A., & Djutalov, R. (2023). PENGEMBANGAN APLIKASI LOKPRO SEBAGAI MEDIA PENCARI KERJA DI LOKPRO MEDIA DENGAN METODE PROTOTYPE. *LOGIC : Jurnal Ilmu Komputer Dan Pendidikan*, 1(3), Article 3.
- Wicha, M., & Pańczyk, B. (2023). Performance analysis of REST API technologies using Spring and Express.js examples. *Journal of Computer Sciences Institute*, 29, 352–359. <https://doi.org/10.35784/jcsi.3796>
- Yusof, K., & Man, M. (2017). Efficiency of JSON for Data Retrieval in Big Data. *Indonesian Journal of Electrical Engineering and Computer Science*, 7, 250–262. <https://doi.org/10.11591/ijeecs.v7.i1.pp250-262>

Yusof, M. K., Man, M., & Ismail, A. (2022). Design and Implement of REST API for Data Integration. *2022 International Conference on Engineering and Emerging Technologies (ICEET)*, 1–4. <https://doi.org/10.1109/ICEET56468.2022.10007414>

LAMPIRAN

a. Surat Keterangan Selesai Penelitian

	PEMERINTAHAN KABUPATEN PURWOREJO KECAMATAN NGOMBOL DESA TUNJUNGAN Jl. Nyi Semendi No.89 RT 003/RW 002 Desa Tunjungan Kode Pos.54172
---	---

Surat Keterangan
Nomor : 625/E...2/97/2/2024

Yang bertanda tangan dibawah ini:

Nama	: Suroto, A.Ma.Pd
Jabatan	: Kepala Desa
Alamat	: Desa Tunjungan, Kecamatan Ngombol, Kabupaten Purworejo

Menerangkan bahwa: .

Nama	: Akhmad Ramadhan
NIM	: 20523049
Prodi/Fakultas/Universitas	: Informatika/Fakultas Teknologi Industri /Univertitas Islam Indonesia

Mahasiswa tersebut benar-benar telah melakukan penelitian di desa Tunjungan sejak tanggal
8 Maret 2024 – 17 Agustus 2024

Demikian surat keterangan ini kami sampaikan untuk dapat digunakan sebagai mana
mestinya.

Purworejo, 17 Agustus 2024
Kepala Desa



Suroto, A.Ma.Pd

b. Dokumentasi

