

**RANCANG BANGUN *RESTFUL API* UNTUK PLATFORM KAMUS
DIGITAL BAHASA LAMPUNG BERBASIS KOMUNITAS**

(Skripsi)

Oleh:

NYOMAN EKA SWARDITA

NPM 2155061006



**FAKULTAS TEKNIK
UNIVERSITAS LAMPUNG
BANDAR LAMPUNG
2025**

**RANCANG BANGUN *RESTFUL API* UNTUK PLATFORM KAMUS
DIGITAL BAHASA LAMPUNG BERBASIS KOMUNITAS**

**Oleh
Nyoman Eka Swardita**

Skripsi

Sebagai Salah Satu Syarat untuk Mencapai Gelar
SARJANA TEKNIK

**Pada
Program Studi Teknik Informatika
Jurusan Teknik Elektro
Fakultas Teknik Universitas Lampung**



**FAKULTAS TEKNIK
UNIVERSITAS LAMPUNG
BANDAR LAMPUNG**

2025

ABSTRAK

RANCANG BANGUN *RESTFUL API* UNTUK PLATFORM KAMUS DIGITAL BAHASA LAMPUNG BERBASIS KOMUNITAS

Oleh

NYOMAN EKA SWARDITA

Bahasa Lampung merupakan salah satu bahasa daerah yang memiliki nilai budaya tinggi, namun keberadaannya semakin terancam oleh dominasi bahasa nasional dan asing serta minimnya dokumentasi digital. Kondisi ini menyebabkan penurunan jumlah penutur aktif dan potensi hilangnya kosakata lokal. Penelitian ini bertujuan mengembangkan sistem *backend* kamus digital bahasa Lampung menggunakan bahasa pemrograman Golang. Tujuan utama pengembangan ini meliputi penyediaan layanan pencarian dan manajemen kosakata yang efisien dengan Redis sebagai *caching* untuk meningkatkan waktu respons, pengelolaan peran pengguna (admin, kontributor, dan peninjau), serta optimalisasi kinerja sistem agar mampu menangani permintaan secara paralel. Proses pengembangan menggunakan metodologi Scrum selama enam sprint, masing-masing berdurasi dua minggu, untuk memastikan penyampaian dan evaluasi secara iteratif. Pengujian performa menggunakan Apache JMeter menunjukkan bahwa sistem mampu menangani hingga 2.000 pengguna virtual dengan waktu respons rata-rata 365,60 ms di localhost dan 1.128,50 ms di VPS, dengan tingkat kesalahan 0%. Pengujian dengan hingga 10.000 pengguna juga dilakukan, menghasilkan tingkat kesalahan 23%, yang menunjukkan bahwa peningkatan sumber daya server diperlukan untuk skalabilitas yang optimal.

Kata Kunci: bahasa Lampung, backend, Golang, kamus digital, Scrum.

ABSTRACT

DESIGN AND DEVELOPMENT OF A RESTFUL API FOR A COMMUNITY-BASED DIGITAL LAMPUNG DICTIONARY PLATFORM

By

NYOMAN EKA SWARDITA

The Lampung language is one of Indonesia's regional languages with high cultural value, but its existence is increasingly threatened by the dominance of national and foreign languages as well as the lack of digital documentation. This condition has led to a decline in the number of active speakers and the potential loss of local vocabulary. This study aims to develop a backend system for a digital Lampung language dictionary using the Go programming language. The main objectives of this development include providing efficient vocabulary search and management services with Redis as caching to improve response time, managing user roles (admin, contributor, and reviewer), and optimizing system performance to handle parallel requests. The development process adopted the Scrum methodology over six sprints, each lasting two weeks, to ensure iterative delivery and evaluation. Performance testing using Apache JMeter showed that the system was able to handle up to 2,000 virtual users with an average response time of 365.60 ms on localhost and 1,128.50 ms on a VPS, with a 0% error rate. Testing with up to 10,000 users was also conducted, resulting in a 23% error rate, indicating that additional server resources are needed for optimal scalability.

Keywords: Lampung language, backend, Golang, digital dictionary, Scrum.

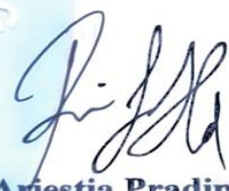
Judul : **Rancang Bangun Restful API Untuk Platform Kamus Digital Bahasa Lampung Berbasis Komunitas**

Nama Mahasiswa : **Nyoman Eka Swardita**
Nomor Pokok Mahasiswa : 2155061006
Program Studi : Teknik Informatika
Fakultas : Teknik

MENYETUJUI

1. Komisi Pembimbing


Mahendra Pratama, S.T., M.Eng.
NIP. 199112152019031013


Rio Ariestia Pradipta, S.Kom, M.T.I.
NIP. 198603232019031013

2. Mengetahui

Ketua Jurusan
Teknik Elektro


Herlinawati, S.T., M.T.
NIP. 197103141999032001

Ketua Program Studi
Teknik Informatika


Yessi Mulyani, S.T., M.T.
NIP. 197312262000122001

MENGESAHKAN

1. Tim Penguji

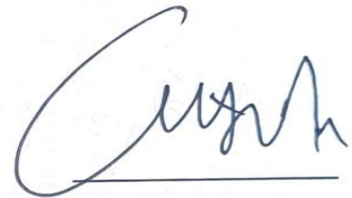
Ketua : **Mahendra Pratama, S.T., M.Eng.**



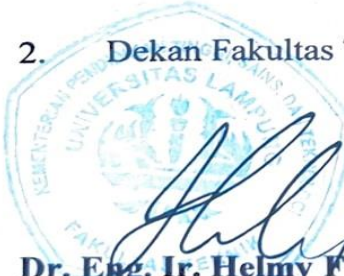
Sekretaris : **Rio Ariesta Pradipta, S.Kom., M.T.I.**



Penguji : **Ir. Gigih Forda Nama, S.T., M.T.I., IPM**



2. Dekan Fakultas Teknik



Dr. Eng. Ir. Helmy Fitriawan, S.T., M.Sc. ¶
NIP. 197509282001121002

Tanggal Lulus Ujian Skripsi: **07 Mei 2025**

SURAT PERNYATAAN

Saya yang bertandatangan dibawah ini, menyatakan bahwa skripsi saya yang berjudul “RANCANG BANGUN RESTFUL API UNTUK PLATFORM KAMUS DIGITAL BAHASA LAMPUNG BERBASIS KOMUNITAS” dengan ini menyatakan bahwa skripsi saya dibuat oleh saya sendiri. Semua hasil yang tertuang dalam skripsi ini telah mengikuti kaidah penulisan karya ilmiah Universitas Lampung. Apabila di kemudian hari terbukti bahwa skripsi ini merupakan salinan atau dibuat oleh orang lain, maka saya bersedia menerima sanksi sesuai dengan ketentuan hukum atau akademik yang berlaku.

Bandar Lampung, 20 Mei 2025

Pembuat Pernyataan,



Nyoman Eka Swardita

NPM 2155061006

RIWAYAT HIDUP



Penulis Bernama Nyoman Eka Swardita yang merupakan anak ketiga dari pasangan I Kadek Sana dan Ni Komang Ayu, Penulis Lahir di desa Labuhan Jaya pada tanggal 22 februari 2003. Penulis telah menyelesaikan sekolah dasar di SD Negeri 1 Labuhan Jaya pada tahun 2015. SMP Negeri 3 Mesuji pada tahun 2018 dan SMA YP Unila pada tahun 2021. Penulis juga melanjutkan

Pendidikan di Fakultas Teknik Jurusan Teknik Elektro Program Studi Teknik Informatika Universitas Lampung pada tahun 2021. Selama menempuh Pendidikan di Jurusan Teknik Elektro Program Studi Teknik Informatika Universitas Lampung, penulis juga aktif mengikuti beberapa kegiatan sebagai berikut:

1. Peserta program Msib Magang sebagai Software Engineer di Gits Indonesia pada tahun 2023
2. Peserta Program Msib Magang sebagai Software Engineer di Telkom Indonesia pada tahun 2024

MOTO

“Kamu hanya berhak untuk melakukan tindakanmu, tetapi tidak pada hasilnya. Jangan biarkan hasil tindakan menjadi motivasi untuk bertindak, dan jangan terikat pada ketidakaktifan”

(Bhagavad Gita II.47)

“Cara terbaik untuk meramalkan masa depan adalah dengan menciptakannya.”

(Alan Kay)

“Kedamaian sejati datang ketika kita mampu menerima diri sepenuhnya, tanpa terpengaruh oleh penilaian orang lain, dan menjalani hidup dengan penuh ketulusan dan kebebasan.”

(Nyoman Eka Swardita)

PERSEMBAHAN

Segala puji syukur kepada Ida Sang Hyang Widhi Wasa berkat karunia, kesehatan serta rahmat-Nya yang telah diberikan. Dengan segala kerendahan hati, saya persembahkan skripsi ini kepada:

Ibuku, yang telah melahirkanku, merawatku, membesarkanku, dan yang telah sepenuh hati mendidikku.

Ayahku tercinta, yang telah membesarkanku dengan seluruh kasih dan sayangnya, memberikan pengetahuannya, dan selalu mendukung serta mendoakan untuk keberhasilanku.

Serta, almamater yang saya sangat banggakan
UNIVERSITAS LAMPUNG

SANWACANA

Puji syukur penulis ucapkan kepada Ida Sang Hyang Widhi Wasa yang telah memberikan karunia-Nya sehingga penulis dapat melaksanakan dan menyelesaikan penelitian ini yang berjudul “RANCANG BANGUN RESTFUL API UNTUK PLATFORM KAMUS DIGITAL BAHASA LAMPUNG BERBASIS KOMUNITAS”. Penelitian ini merupakan salah satu syarat untuk menyelesaikan kurikulum mata kuliah penelitian skripsi pada Program Studi Teknik Informatika, Jurusan Teknik Elektro, Fakultas Teknik, Universitas Lampung. Pelaksanaan penelitian ini, penulis mendapatkan bantuan, bimbingan serta pengarahan dari berbagai pihak. Oleh karena itu, penulis mengucapkan terima kasih sebanyak-banyaknya kepada:

1. Ida Sang Hyang Widhi Wasa yang senantiasa memberikan kemudahan dan kelancaran kepada penulis dalam melakukan penelitian.
2. Ibu dan Ayah penulis yang senantiasa memberikan dukungan serta motivasi kepada penulis .
3. Bapak Dr. Eng. Helmy Fitriawan, S.T., M.Sc selaku Dekan Fakultas Teknik Universitas Lampung.
4. Ibu Herlinawati, S.T.,M.T. selaku Ketua Jurusan Teknik Elektro Universitas Lampung.
5. Ibu Yessi Mulyani, S.T., M.T. selaku Ketua Program Studi Teknik Informatika Universitas Lampung.
6. Bapak Mahendra Pratama, S.T., M.Eng. selaku Pembimbing Utama yang selalu memberikan bimbingan serta dukungan kepada penulis.
7. Bapak Rio Ariestia Pradipta, S.Kom, M.T.I. selaku Pembimbing Pendamping yang selalu memberikan motivasi serta bimbingan.

8. Bapak Puput Budi Wintoro, S. Kom, M.T.I selaku Pembimbing Akademik yang selalu memberikan motivasi kepada penulis.
9. Bapak Ir. Gigih Forda Nama, S.T., M.T.I., IPM selaku Penguji yang memberikan banyak arahan kepada penulis.
10. Teman-teman Lamban Bahasa yang selalu memberi dukungan, inspirasi dan kerja sama sepanjang pengerjaan skripsi ini.
11. Teman-teman PSTI 2021 yang selalu memberi dukungan kepada penulis.
12. Seluruh pihak yang telah berkontribusi dalam penyelesaian penelitian ini dan tidak dapat disebutkan satu per satu oleh penulis.

Penulis menyadari bahwa laporan penelitian ini masih memiliki kekurangan dan dapat disempurnakan di masa mendatang. Oleh karena itu, penulis sangat mengharapkan kritik serta saran dari para pembaca. Semoga penelitian ini dapat memberikan manfaat bagi banyak pihak.

Bandar Lampung
Penulis,

Nyoman Eka Swardita

DAFTAR ISI

	Halaman
DAFTAR ISI.....	xiii
DAFTAR GAMBAR.....	xv
DAFTAR TABEL	xvi
I. PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	4
1.3 Tujuan Penelitian	4
1.4 Manfaat Penelitian	4
1.5 Batasan Masalah	5
1.6 Sistematika Penulisan Skripsi	5
II. TINJAUAN PUSTAKA.....	7
2.1 Bahasa Lampung.....	7
2.2 Kamus	7
2.3 Golang	8
2.4 <i>Three-layered architecture</i>	9
2.5 PostgreSQL	10
2.6 Redis	11
2.7 Docker	12
2.8 Jalur Integrasi dan Pengiriman Berkelanjutan (<i>CI/CD</i>).....	12

2.9 Typesense.....	14
2.10 Swagger.....	14
2.11 Nginx.....	16
2.12 Agile	17
2.13 Scrum	18
2.14 <i>Application Programming Interface (API)</i>	19
2.15 Penelitian Terkait	20
III. METODE PENELITIAN.....	23
3.1 Waktu dan Tempat.....	23
3.2 Alat dan Bahan.....	24
3.3 Tahapan Penelitian.....	25
3.3.1 Analisis Kebutuhan.....	26
3.3.2 Perancangan sistem	32
3.3.3 Perencanaan	35
3.3.4 Product Backlog.....	36
3.3.5 Sprint Backlog	36
3.3.6 Sprint.....	36
3.3.7 Sprint Review.....	37
3.3.8 Retrospective.....	37
V. KESIMPULAN DAN SARAN	38
5.1 Kesimpulan	38
5.2 Saran	38
DAFTAR PUSTAKA.....	41

DAFTAR GAMBAR

	Halaman
Gambar 2. 1 <i>Three layered architecture</i> golang	10
Gambar 2. 2 Continues Integration/Continuous Deployment (CI/CD)	13
Gambar 2. 3 Contoh Penggunaan Swagger	16
Gambar 3. 1 Metode Pengembangan <i>Scrum</i>	25
Gambar 3. 2 Metode Pengembangan Scrum (lanjutan)	26
Gambar 3. 3 Arsitektur sistem	33
Gambar 3. 4 Bagan Dari Tim.....	36

DAFTAR TABEL

	Halaman
Tabel 3. 1 jadwal penelitian	23
Tabel 3. 2 Alat yang digunakan dalam penelitian.....	24
Tabel 3. 3 Kebutuhan Fungsional	28
Tabel 3. 4 Kebutuhan Non Fungsional	31

I. PENDAHULUAN

1.1 Latar Belakang

Bahasa Lampung adalah salah satu bahasa daerah di Indonesia yang memiliki peran penting dalam kehidupan masyarakat Lampung. Sebagai alat komunikasi, bahasa ini tidak hanya berfungsi untuk menjaga keberlangsungan nilai-nilai tradisional, tetapi juga sebagai media pelestarian warisan leluhur yang turun-temurun. Bahasa Lampung memainkan peran yang sangat signifikan dalam menjaga identitas budaya masyarakat Lampung. Namun, seiring dengan pesatnya perkembangan globalisasi dan teknologi, bahasa Lampung menghadapi tantangan besar dalam kelangsungan penggunaannya. Globalisasi telah memperkenalkan bahasa Indonesia dan bahasa asing sebagai bahasa utama dalam berbagai bidang, yang mengakibatkan semakin sedikitnya generasi muda yang menggunakan bahasa Lampung dalam interaksi sehari-hari.

Fenomena ini diperburuk oleh pengaruh budaya populer yang mengutamakan bahasa Indonesia dan bahasa Inggris, yang dianggap lebih penting dan relevan dalam kehidupan modern. Akibatnya, bahasa Lampung yang memiliki kekayaan kosakata dan tata bahasa yang unik, perlahan mulai terpinggirkan. Selain itu, terdapat kurangnya dokumentasi dan aksesibilitas terhadap bahasa Lampung yang menjadi kendala dalam pelestariannya. Banyak kosakata dan tata bahasa tradisional yang tidak terdokumentasi dengan baik, sehingga sulit dipelajari oleh generasi muda.

Untuk itu, pelestarian bahasa Lampung perlu mendapatkan perhatian serius. Salah satu cara yang dapat ditempuh adalah dengan memanfaatkan teknologi digital untuk mendokumentasikan dan mempromosikan kembali bahasa ini. Teknologi memungkinkan pembuatan kamus digital, aplikasi pencarian kata, dan platform berbasis komunitas yang mempermudah masyarakat untuk mengakses dan berkontribusi dalam pelestarian bahasa Lampung.

Beberapa aplikasi telah dikembangkan untuk mendukung pelestarian bahasa Lampung, namun masih memiliki keterbatasan signifikan. Misalnya, aplikasi Translate Bahasa Lampung menyediakan terjemahan untuk dua dialek utama bahasa Lampung, tetapi tidak menyertakan fitur contoh kalimat, sistem kolaborasi, atau pencarian berbasis dialek tertentu. Hal serupa terjadi pada aplikasi Kamus Lampung-Indonesia yang meskipun menyediakan daftar kata cukup lengkap, tetapi bersifat statis dan tidak mendukung pembaruan data secara interaktif.

Mengatasi kekurangan tersebut, penelitian ini berfokus pada pengembangan sistem backend untuk aplikasi pencarian kata bahasa Lampung. Sistem yang akan dikembangkan memiliki fitur utama untuk mencari kata dalam bahasa Lampung berdasarkan dialek. Setiap kata yang dicari akan menampilkan detail informasi yang terdiri dari penulisan aksara dari kata tersebut, contoh kalimat yang menggunakan kata tersebut dalam bahasa Lampung dan bahasa Indonesia, serta penulisan aksara dari kalimat contoh tersebut.

Proyek ini mencakup pengembangan backend, yang bertugas untuk mengelola data kata, kalimat, serta proses verifikasi dan validasi kata dan kalimat yang dimasukkan oleh pengguna dengan peran kontributor dan peninjau. Pengguna dengan peran kontributor dapat menambahkan kata dan kalimat contoh dalam bahasa Lampung, yang kemudian akan diperiksa oleh pengguna dengan peran peninjau. Setelah melalui proses verifikasi, kata dan kalimat tersebut baru dapat ditampilkan dalam fitur pencarian, yang dapat diakses oleh pengguna publik.

Sistem ini memungkinkan masyarakat untuk berperan aktif dalam menambah kosakata bahasa Lampung, sekaligus memperkenalkan aksara Lampung dan tata bahasa yang mungkin sudah tidak lagi digunakan dalam kehidupan sehari-hari. Aplikasi ini juga memungkinkan masyarakat untuk ikut berperan dalam mengembangkan dan memvalidasi data yang ada melalui peran kontributor dan peninjau.

Seiring dengan berkembangnya aplikasi ini, diharapkan dapat menciptakan komunitas digital yang aktif dalam pelestarian bahasa Lampung. Masyarakat dapat berkolaborasi untuk menambah kata dan kalimat contoh dalam bahasa Lampung, sehingga memperkaya korpus bahasa Lampung yang ada. Dengan demikian, aplikasi ini tidak hanya berfungsi sebagai alat pencarian kata, tetapi juga sebagai platform pelestarian bahasa Lampung yang berbasis komunitas.

Upaya pelestarian bahasa Lampung melalui teknologi digital ini diharapkan dapat memberikan kontribusi positif dalam menjaga eksistensi bahasa Lampung di tengah era globalisasi yang semakin pesat. Dengan aplikasi ini, generasi muda diharapkan dapat lebih tertarik untuk mempelajari bahasa Lampung, sekaligus memperkenalkan dan melestarikan bahasa ini kepada masyarakat luas.

Selain itu, aplikasi ini juga menjadi langkah awal dalam mendigitalisasi kamus bahasa Lampung yang lebih lengkap dan mudah diakses oleh publik. Dengan demikian, pelestarian bahasa Lampung tidak hanya bergantung pada tradisi lisan, tetapi juga bisa diakses secara digital, yang lebih relevan dengan kebutuhan zaman sekarang.

Pentingnya pelestarian bahasa daerah, khususnya bahasa Lampung, perlu didukung oleh semua pihak. Masyarakat, pemerintah daerah, dan lembaga pendidikan perlu bekerja sama untuk menjaga keberlangsungan bahasa daerah melalui berbagai inovasi, salah satunya adalah pemanfaatan teknologi digital dalam proses dokumentasi dan edukasi. Melalui kolaborasi ini, diharapkan bahasa Lampung dapat terus berkembang dan dilestarikan untuk generasi mendatang.

Untuk itu, upaya revitalisasi bahasa Lampung dapat dipercepat dengan pemanfaatan teknologi digital, yang memberikan akses yang lebih mudah dan fleksibel untuk pembelajaran dan pelestarian bahasa daerah. Penggunaan aplikasi berbasis komunitas, seperti sistem pencarian kata bahasa Lampung ini, menjadi salah satu cara efektif dalam mendigitalisasi bahasa Lampung dan membuatnya lebih mudah diakses oleh masyarakat luas [1].

1.2 Rumusan Masalah

Adapun rumusan masalah dari penelitian ini adalah sebagai berikut:

1. Bagaimana membangun sebuah Sistem *Restful Api* kamus Bahasa Lampung berbasis komunitas yang dapat mendukung fitur pencarian bahasa Lampung secara efisien?
2. Bagaimana membangun sistem *backend* untuk *platform* bahasa Lampung yang dapat mendukung pengelolaan peran admin, kontributor, dan peninjau dalam proses verifikasi dan publikasi data bahasa Lampung?
3. Bagaimana memastikan performa sistem *backend* yang dibangun memiliki respons *API* yang cepat dan menyediakan data dengan latensi rendah untuk meningkatkan pengalaman pengguna secara keseluruhan?

1.3 Tujuan Penelitian

Adapun tujuan dari penelitian ini adalah sebagai berikut:

1. Mengembangkan sistem *backend* berbasis Golang yang efisien untuk mendukung layanan pencarian dan manajemen kosakata bahasa Lampung dengan menggunakan Redis untuk *caching* guna meningkatkan waktu respons.
2. Mengembangkan sistem manajemen peran dalam backend yang memungkinkan admin, kontributor, dan peninjau untuk berkontribusi dalam proses verifikasi dan publikasi data bahasa Lampung.
3. Mengoptimalkan kinerja dan skalabilitas sistem *backend* untuk memastikan sistem dapat menangani banyak permintaan pengguna secara bersamaan dengan menggunakan teknik penyimpanan sementara dan pemrosesan paralel.

1.4 Manfaat Penelitian

Adapun manfaat yang diharapkan dari penelitian ini adalah sebagai berikut:

1. Penelitian ini memberikan kemudahan bagi masyarakat, khususnya generasi muda, untuk mengakses dan mempelajari bahasa Lampung melalui platform digital.
2. Penelitian ini memungkinkan penutur asli atau ahli bahasa Lampung untuk berkontribusi dalam memperkaya kosakata bahasa Lampung melalui fitur kontributor.
3. Platform ini diharapkan dapat menjadi alat yang efektif dalam melestarikan bahasa Lampung sebagai bagian dari upaya pelestarian warisan budaya tak benda di Indonesia.
4. Penelitian ini berkontribusi pada pengembangan aplikasi dan web berbasis komunitas yang mendukung pelestarian budaya lokal dengan teknologi informasi

1.5 Batasan Masalah

Adapun Batasan masalah dalam penelitian ini adalah sebagai berikut:

1. Penelitian ini hanya berfokus pada pengembangan sistem backend menggunakan Golang, tidak mencakup pengembangan *frontend* atau *mobile application* secara mendalam.
2. Pengembangan sistem ini dibatasi pada bahasa Lampung dan fitur-fitur terkait, seperti pencarian kata, dan penambahan kosakata bahasa Lampung.
3. Tidak mencakup pengembangan fitur pencarian Bahasa lampung dengan menggunakan kalimat.

1.6 Sistematika Penulisan Skripsi

Laporan ini dibagi menjadi beberapa bab untuk memudahkan dalam penguraian, antara lain:

BAB I: PENDAHULUAN

Bab ini berisi latar belakang masalah, rumusan masalah, tujuan penelitian, manfaat penelitian, serta batasan masalah. Bab ini

memberikan pengantar mengenai pentingnya pelestarian bahasa Lampung serta kebutuhan akan teknologi dalam mendukung pelestarian tersebut.

BAB II: TINJAUAN PUSTAKA

Bab ini menguraikan teori-teori yang mendukung penelitian, termasuk penjelasan mengenai bahasa Lampung, teknologi *backend*, serta konsep-konsep yang berkaitan dengan pengembangan sistem *backend* menggunakan *Golang*. Selain itu, akan dijelaskan pula mengenai metodologi *Scrum* yang digunakan dalam pengembangan proyek ini.

BAB III: METODOLOGI PENELITIAN

Bab ini menjelaskan metode penelitian yang digunakan dalam pengembangan sistem *backend* untuk pelestarian bahasa Lampung. Pembahasan mencakup pengenalan *Scrum* sebagai metode kerja yang melibatkan *product backlog*, *sprint backlog*, *sprint*, serta *sprint review*.

BAB IV: HASIL DAN PEMBAHASAN

Bab ini berisi penjelasan detail mengenai hasil implementasi dari metode yang dijelaskan pada bab sebelumnya. Dijelaskan bagaimana *product backlog*, *sprint backlog*, dan *sprint* diterapkan dalam pengembangan sistem *backend*. Selain itu, akan dijelaskan hasil pengujian sistem.

BAB V: KESIMPULAN DAN SARAN

Bab ini berisi Kesimpulan dari hasil penelitian yang telah dilaksanakan serta saran-saran sebagai masukan untuk penelitian lebih lanjut di masa mendatang.

DAFTAR PUSTAKA

LAMPIRAN

II. TINJAUAN PUSTAKA

2.1 Bahasa Lampung

Bahasa Lampung merupakan salah satu bahasa daerah di Indonesia yang memiliki sejarah panjang dan memainkan peranan penting dalam kehidupan masyarakat Lampung. Sebagai bahasa daerah, bahasa ini tidak hanya berfungsi sebagai alat komunikasi, tetapi juga sebagai wadah yang menyimpan nilai-nilai kearifan lokal, tradisi, dan identitas budaya masyarakat Lampung. Melalui bahasa Lampung, nilai-nilai budaya, cerita rakyat, adat istiadat, serta norma-norma sosial diwariskan dari satu generasi ke generasi berikutnya.

Bahasa Lampung terbagi menjadi dua dialek utama, yaitu Dialek Api dan Dialek Nyo [2], yang masing-masing memiliki variasi penggunaan dan penutur. Meskipun bahasa ini pernah mengalami masa kejayaannya, *modernisasi* dan perkembangan zaman yang pesat telah menyebabkan terjadinya penurunan jumlah penutur asli, terutama di kalangan generasi muda. Oleh karena itu, upaya pelestarian bahasa Lampung menjadi penting, bukan hanya untuk mempertahankan warisan budaya tak benda, tetapi juga untuk menjaga identitas masyarakat Lampung di tengah arus globalisasi.

2.2 Kamus

Kamus adalah kumpulan kata atau frasa dalam suatu bahasa yang disertai dengan definisi, terjemahan, atau penjelasan mengenai makna dan penggunaannya. Kamus berperan penting dalam dokumentasi dan pelestarian bahasa, serta membantu pengguna memahami makna dan penggunaan kata dalam konteks tertentu.

Menurut *Jackson (2002)* dalam bukunya *Lexicography: An Introduction*, kamus tidak hanya berfungsi sebagai alat untuk menerjemahkan kata, tetapi juga sebagai sarana

untuk memahami perkembangan bahasa dan budaya masyarakat. Kamus yang baik harus mencakup aspek semantik, sintaksis, dan etimologi dari setiap kata yang dimuat [3].

Dalam konteks digital, kamus telah mengalami transformasi dengan hadirnya kamus elektronik (e-dictionary) yang memungkinkan pengguna melakukan pencarian kata secara instan melalui aplikasi berbasis web atau mobile. Menurut Zgusta (1971) dalam bukunya *Manual of Lexicography*, kamus elektronik memberikan aksesibilitas yang lebih tinggi, terutama untuk bahasa-bahasa yang terancam punah [4].

Dalam penelitian ini, sistem kamus yang dikembangkan bertujuan untuk mendokumentasikan dan memperkenalkan kembali bahasa Lampung kepada masyarakat luas. Penerapan teknologi informasi pada kamus digital diharapkan dapat mendukung upaya pelestarian bahasa daerah yang semakin tergerus oleh globalisasi.

2.3 Golang

Golang atau Go adalah bahasa pemrograman yang dikembangkan oleh Google pada tahun 2007 dan diumumkan secara resmi pada tahun 2009 [5]. Golang dirancang untuk mengatasi keterbatasan bahasa pemrograman lain dalam hal performa, terutama untuk aplikasi yang membutuhkan skala besar dan proses paralel yang tinggi [6]. Salah satu keunggulan utama Golang adalah kemampuannya untuk menangani banyak proses secara bersamaan, yang menjadikannya pilihan yang sangat baik untuk pengembangan sistem backend yang membutuhkan kecepatan dan kinerja tinggi [6].

Berdasarkan dari penelitian yang dipublikasikan dalam prosiding *The 7th International Conference on Information Technology* yang berjudul "*Performance Evaluation of Programming Languages as API Services for Cloud Environments: A Comparative Study of PHP, Python, Node.js and Golang*", hasil eksperimen menunjukkan bahwa pada skenario dengan 100 pengguna secara bersamaan, Golang memberikan respons tercepat dibandingkan bahasa lain seperti PHP dan Node.js. Dalam pengujian pengambilan data melalui API dengan ukuran data $n = 10.000$, Golang mencatat waktu

respons hanya 6,47 detik, jauh lebih cepat dibandingkan NGINX + PHP yang memerlukan rata-rata 33,08 detik. Pengujian juga menunjukkan bahwa Golang lebih unggul dalam pemrosesan bersamaan dengan performa yang 3,5 kali lebih cepat dibandingkan Node.js, meskipun penggunaan CPU mendekati 100%. Selain itu, pada eksperimen unggah file berukuran 50 KB dengan 100 pengguna, Golang mampu menjaga nilai throughput yang konsisten di angka 22,90 req/detik, setara dengan sistem lain seperti Apache dan Node.js [7].

Pemilihan bahasa pemrograman yang tepat sangat penting dalam pengembangan sistem yang berkinerja tinggi [8], terutama untuk aplikasi berbasis *API RESTful* yang akan mengelola dan memproses data secara besar-besaran. Golang, dengan kemampuannya dalam menangani banyak koneksi secara simultan, sangat cocok untuk mendukung aplikasi pencarian kata bahasa Lampung yang memerlukan efisiensi dan performa tinggi.

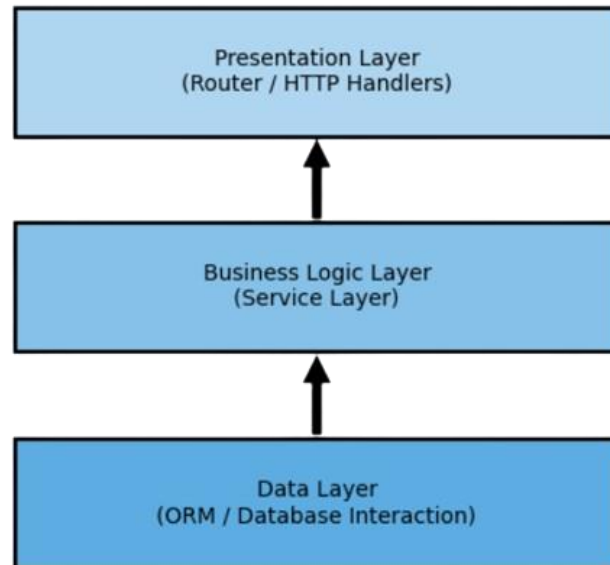
Dengan Golang, pengembang dapat membuat *API* yang dapat menangani ribuan permintaan per detik tanpa mengorbankan kecepatan atau efisiensi [9]. Ini menjadikan Golang sebagai pilihan yang tepat untuk proyek ini, di mana *backend* harus menangani pencarian kata dalam bahasa Lampung dan memastikan bahwa sistem dapat merespons dengan cepat dan efisien terhadap permintaan pengguna.

2.4 Three-layered architecture

Three-layered architecture adalah pendekatan umum dalam pengembangan aplikasi yang membagi sistem menjadi tiga lapisan utama: *Presentation Layer*, *Business Logic Layer*, dan *Data Layer*. Pendekatan ini bertujuan untuk memisahkan tanggung jawab di setiap bagian aplikasi sehingga lebih modular, mudah dipelihara, dan skalabel. Dalam konteks pengembangan menggunakan Golang, *three-layered architecture* sering diadopsi untuk memastikan struktur kode yang jelas dan pengelolaan komponen yang efisien.

Setiap lapisan memiliki fokus khusus, terutama dalam hal pengelolaan dan kontrol data. Lapisan pertama bertugas menangani data dan persisten, sedangkan logika bisnis ditempatkan pada lapisan di atasnya. Lapisan paling atas dikhususkan untuk interaksi dengan pihak eksternal. Prinsip utama arsitektur ini adalah setiap lapisan tidak memiliki ketergantungan pada lapisan yang berada di atasnya. Sebagai contoh, lapisan data tidak mengetahui logika bisnis, sedangkan logika bisnis tidak memiliki informasi tentang API atau mekanisme interaksi eksternal lainnya [10]. Gambar 2.1 merupakan contoh ilustrasi *three-layered architecture* pada *golang*.

Three-Layered Architecture in Golang



Gambar 2. 1 *Three layered architecture* golang

2.5 PostgreSQL

PostgreSQL adalah sistem manajemen basis data relasional berorientasi objek yang bersifat open source dan telah dikembangkan secara aktif selama lebih dari 35 tahun. Sistem ini dikenal karena kehandalannya, fitur yang kaya, dan kinerja yang tinggi. PostgreSQL mendukung berbagai sistem operasi, termasuk Linux, UNIX, dan

Windows, serta menawarkan skalabilitas, fleksibilitas, dan kinerja yang tinggi bagi pengguna. [11]

Selain itu, PostgreSQL menawarkan skalabilitas yang tinggi, memungkinkan penanganan beban kerja yang besar dan kompleks. Fitur seperti partisi tabel, sharding, dan eksekusi kueri paralel memungkinkan penanganan dataset besar secara efisien. [11]

Beberapa penelitian telah membandingkan performa PostgreSQL dengan sistem basis data lainnya. Misalnya, dalam sebuah studi yang membandingkan kemampuan PostgreSQL, MariaDB, dan MongoDB dalam memproses data AIS satelit LAPAN, ditemukan bahwa PostgreSQL memiliki keunggulan dalam hal kecepatan pemrosesan query tertentu. [12] Penelitian lain yang membandingkan performa MySQL dan PostgreSQL menunjukkan bahwa PostgreSQL cenderung lebih cepat dalam mengeksekusi beberapa jenis query, terutama pada jumlah data yang besar. [13]

2.6 Redis

Redis (*Remote Dictionary Server*) adalah penyimpanan struktur data dalam memori yang bersifat open source dan sering digunakan sebagai *cache*, *message broker*, dan database key-value. Redis menyimpan data sepenuhnya di memori utama, sehingga memungkinkan akses data yang sangat cepat dibandingkan dengan penyimpanan berbasis disk tradisional. Redis mendukung berbagai struktur data seperti string, list, set, hash, dan sorted set [14].

Secara teknis, Redis sering digunakan untuk meningkatkan performa sistem dengan menyimpan data yang sering diakses (*caching*) atau menyimpan session pengguna. Dalam konteks pengembangan aplikasi backend, Redis sangat efektif untuk menangani skenario seperti penyimpanan token sementara, throttle request API, dan menyimpan data yang sering diminta agar mengurangi beban pada basis data utama.

Dalam implementasi pada sistem ini, Redis digunakan untuk menyimpan cache dari data statistik dan kontribusi teratas, sehingga mengurangi permintaan langsung ke database dan meningkatkan performa sistem.

2.7 Docker

Docker adalah platform sumber terbuka yang mempermudah pengembang untuk membuat, mengelola, dan menjalankan aplikasi dalam unit yang disebut *container*. *Container* ini berfungsi sebagai unit yang ringan dan mandiri, mengemas aplikasi bersama semua dependensi, pustaka, dan konfigurasi yang diperlukan sehingga dapat dijalankan secara konsisten di berbagai lingkungan. Berbeda dengan mesin virtual, *container* berbagi kernel sistem operasi dengan host tetapi tetap memiliki isolasi antar aplikasi, membuatnya lebih hemat sumber daya dan cepat dibandingkan *virtual machine*.

Dalam konteks penelitian ini, Docker digunakan untuk menciptakan lingkungan *backend* yang konsisten dari pengembangan hingga produksi. Dengan Docker, risiko *dependency conflict* yang sering muncul saat aplikasi dikembangkan dan dijalankan di berbagai lingkungan dapat diminimalkan. Docker juga mempermudah pengelolaan *CI/CD*, di mana *image container* yang sudah diuji dapat langsung dikirimkan ke *server* produksi melalui pipeline otomatis [15].

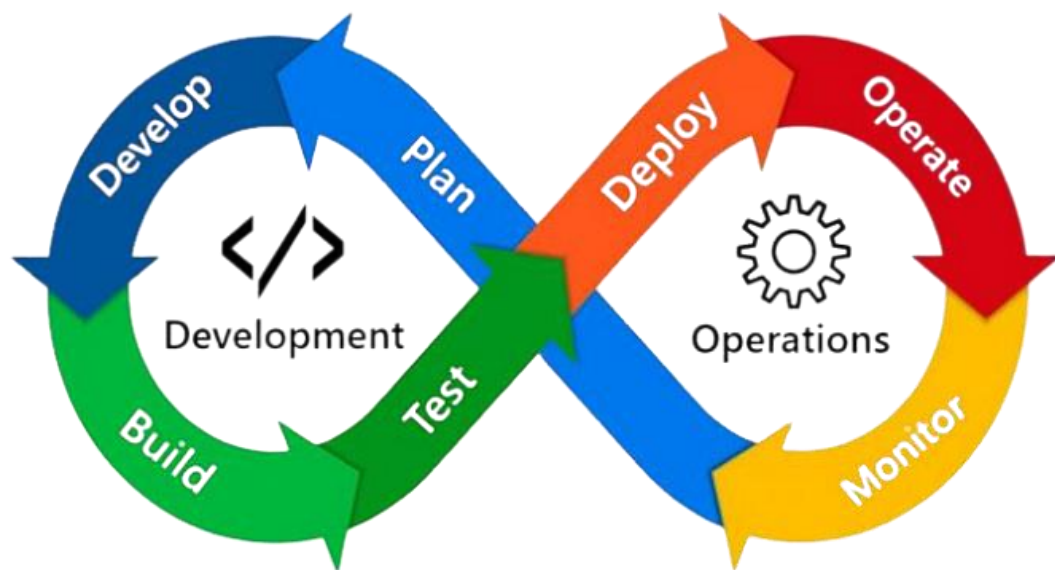
Selain itu, penggunaan Docker meningkatkan skalabilitas sistem. Jika permintaan pengguna meningkat, *kontainer* dapat dengan mudah direplikasi untuk menangani beban tambahan. Docker juga mendukung integrasi dengan Redis sebagai *caching layer*, PostgreSQL sebagai database utama, serta NGINX untuk *reverse proxy*. Secara keseluruhan, Docker tidak hanya meningkatkan efisiensi dalam pengembangan tetapi juga memungkinkan *deployment* aplikasi yang lebih cepat dan andal [15].

2.8 Jalur Integrasi dan Pengiriman Berkelanjutan (CI/CD)

Dalam pengembangan perangkat lunak modern, jalur integrasi dan pengiriman berkelanjutan (*CI/CD*) memainkan peran penting dalam memastikan bahwa integrasi kode dan pengiriman aplikasi dapat dilakukan secara otomatis, efisien, dan tanpa gangguan [16]. Dengan menggunakan *CI/CD*, proses pengujian dan pengiriman aplikasi dapat dilakukan setiap kali ada perubahan pada kode sumber, yang membantu

pengembang untuk menemukan bug lebih awal dan mengurangi kemungkinan kesalahan dalam pengiriman aplikasi ke lingkungan produksi.

GitLab, sebagai platform pengembangan berbasis Git, menyediakan alat yang kuat untuk mengimplementasikan jalur *CI/CD*. Dengan GitLab CI, pengembang dapat menyiapkan rangkaian otomatisasi untuk menguji, membangun, dan mengirim aplikasi ke server produksi tanpa perlu intervensi manual. Penggunaan jalur *CI/CD* dalam pengembangan backend tidak hanya mempercepat siklus pengembangan, tetapi juga meningkatkan kualitas perangkat lunak dengan mengurangi risiko kesalahan manusia dalam proses pengiriman [17]. Gambar 2.2 merupakan ilustrasi penerapan *CI/CD* pada pengembangan project.



Gambar 2. 2 Continuous Integration/Continuous Deployment (CI/CD)

Selain itu, penggunaan Docker dalam proses pengiriman memastikan bahwa aplikasi dapat dijalankan dalam lingkungan yang konsisten, baik pada lokal maupun di server produksi. Docker memungkinkan pengembang untuk mengemas aplikasi beserta semua ketergantungannya dalam kontainer yang dapat dipindahkan dengan mudah

antara lingkungan pengembangan dan produksi, sehingga memastikan aplikasi berfungsi dengan cara yang sama di berbagai platform.

2.9 Typesense

Typesense adalah sebuah search engine open-source yang dirancang untuk memberikan pengalaman pencarian teks yang cepat, relevan, dan mudah diintegrasikan ke dalam berbagai aplikasi. Berbeda dengan search engine tradisional yang cenderung kompleks dan sulit dikonfigurasi, Typesense menekankan kemudahan penggunaan tanpa mengorbankan performa pencarian [18].

Typesense cocok digunakan dalam berbagai skenario pencarian, seperti katalog produk, kamus digital, sistem informasi, dan lain sebagainya. Salah satu keunggulan utama Typesense adalah kemampuannya untuk menangani *fuzzy search* secara efisien, yang memungkinkan pengguna menemukan hasil pencarian meskipun terdapat kesalahan ketik atau variasi dalam kata kunci.

Secara teknis, Typesense dibangun menggunakan bahasa pemrograman C++ dan mendukung pencarian berbasis *inverted index* yang dioptimalkan untuk kecepatan dan efisiensi memori. Typesense menawarkan *RESTful API* yang memungkinkan pengembang berinteraksi dengan search engine melalui *HTTP* requests.

Typesense dapat dijalankan secara lokal menggunakan binary atau container Docker, maupun di-hosting secara cloud. Pada umumnya, proses pencarian dilakukan melalui koleksi data yang didefinisikan oleh pengguna, yang disebut dengan **collection**. Masing-masing koleksi memiliki skema data yang harus ditentukan di awal, mirip dengan skema tabel pada database.

2.10 Swagger

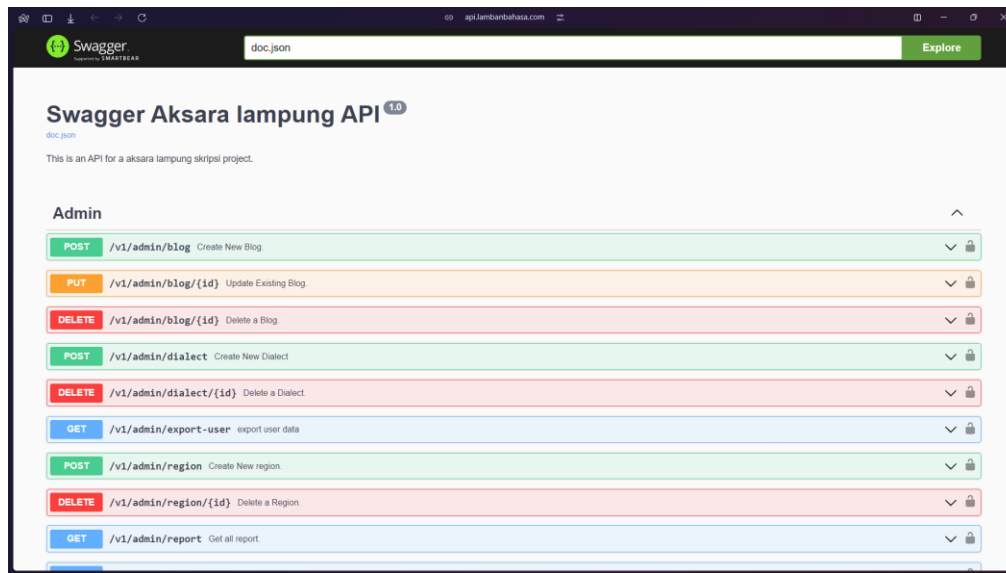
Swagger adalah sebuah kerangka kerja yang digunakan untuk merancang, membangun, mendokumentasikan, dan mengonsumsi *API* berbasis *REST*. *Swagger* sangat membantu dalam pengembangan aplikasi modern yang berkomunikasi melalui

API karena memberikan alat otomatis untuk mendokumentasikan seluruh endpoint *API* yang ada. Dalam proyek pengembangan *backend*, *Swagger* digunakan untuk mendokumentasikan *API* yang berfungsi dalam pencarian dan terjemahan bahasa Lampung, serta fitur untuk kontributor, peninjau dan admin.

Salah satu keunggulan *Swagger* adalah kemampuannya menghasilkan dokumentasi interaktif secara otomatis. Dokumentasi ini memungkinkan pengembang dan pengguna *API* untuk memahami bagaimana setiap endpoint bekerja, termasuk parameter yang dibutuhkan, format respon, serta status kode *HTTP* yang mungkin dikembalikan. Selain itu, *Swagger* memungkinkan pengguna untuk mencoba setiap endpoint langsung melalui antarmuka dokumentasi yang dihasilkan.

Swagger juga mendukung standar *OpenAPI*, yang menjadi acuan dalam mendefinisikan spesifikasi *API*. Hal ini memudahkan integrasi dan pengembangan lebih lanjut karena *API* yang terdokumentasi dengan baik akan lebih mudah dipahami oleh tim pengembang maupun pihak ketiga yang ingin menggunakan *API* tersebut.

Penggunaan *Swagger* dalam pengembangan *backend* ini memastikan bahwa *API* yang dikembangkan terdokumentasi dengan baik dan mudah diakses oleh tim pengembang lain, termasuk tim front-end dan pihak yang terlibat dalam pengujian. Gambar 2.3 merupakan contoh hasil penggunaan *swagger* dalam pengembangan sistem *backend*.



Gambar 2. 3 Contoh Penggunaan Swagger

2.11 Nginx

Nginx adalah perangkat lunak open-source yang berfungsi sebagai web server, reverse proxy, load balancer, dan HTTP cache. Nginx dikenal karena performanya yang tinggi, penggunaan memori yang rendah, serta kemampuannya untuk menangani ribuan koneksi secara bersamaan secara efisien. Nginx banyak digunakan untuk menyajikan konten web statis, meneruskan permintaan ke aplikasi backend, dan mengelola lalu lintas jaringan dalam skala besar.

Secara teknis, Nginx bekerja menggunakan pendekatan *event-driven architecture*, berbeda dengan pendekatan thread-based yang digunakan oleh server web seperti Apache. Pendekatan ini memungkinkan Nginx melayani banyak koneksi secara bersamaan dengan sumber daya minimum. Nginx juga sangat kompatibel untuk digunakan dalam arsitektur microservices, karena kemampuannya sebagai reverse proxy dan load balancer [19].

Dalam konteks pengembangan sistem ini, Nginx digunakan sebagai *reverse proxy server* yang mengarahkan permintaan dari pengguna ke backend yang dikembangkan

menggunakan Golang. Nginx juga berperan dalam manajemen routing dan keamanan seperti SSL termination dan rate limiting.

2.12 Agile

Agile adalah metodologi pengembangan perangkat lunak yang bersifat kolaboratif dan adaptif, dirancang untuk menangani perubahan kebutuhan secara fleksibel selama siklus pengembangan. Pendekatan ini berpusat pada kolaborasi tim, pengiriman produk yang terus-menerus, dan tanggapan cepat terhadap perubahan. Dalam *Agile*, pengembangan perangkat lunak dipecah menjadi iterasi atau siklus pendek yang memungkinkan tim untuk merilis fitur baru secara bertahap dan memperoleh umpan balik lebih awal.

Sebagai salah satu pendekatan yang paling banyak digunakan dalam pengembangan perangkat lunak modern, *Agile* memiliki beberapa keunggulan utama. Menurut jurnal “*Game Development with Scrum Methodology*” (2022), *Agile* memungkinkan kolaborasi yang lebih baik antara tim lintas disiplin. Dalam pengembangan gim, misalnya, *Agile* digunakan untuk menyelaraskan pekerjaan antara pengembang, desainer, dan artis visual dengan menggunakan kerangka kerja seperti *Scrum*. Jurnal ini menekankan bahwa *Agile* mendukung komunikasi yang transparan dan meningkatkan efisiensi penyelesaian tugas dalam proyek yang sangat dinamis [20].

Selain itu, jurnal “*A Comparative Study of Implementing Agile Methodology and Scrum Framework for Software Development*” (2020), menyebutkan bahwa *Agile* memberikan keuntungan utama berupa peningkatan kualitas perangkat lunak dan kepuasan pelanggan melalui iterasi reguler dan pembaruan yang cepat. Dalam studi ini, *Agile* terbukti efektif dalam proyek perangkat lunak kompleks karena kemampuannya untuk menyesuaikan diri dengan kebutuhan yang terus berubah tanpa mengorbankan produktivitas tim [21].

Dengan mengintegrasikan *Agile* ke dalam proyek pengembangan, seperti yang didokumentasikan oleh penelitian-penelitian tersebut, tim dapat meningkatkan

fleksibilitas dan memastikan produk akhir lebih sesuai dengan kebutuhan pengguna. Pendekatan ini relevan untuk mendukung pengembangan sistem backend aplikasi bahasa Lampung yang kompleks dan membutuhkan pembaruan berkelanjutan sesuai kebutuhan budaya dan bahasa lokal.

2.13 Scrum

Scrum adalah salah satu kerangka kerja dari metodologi *Agile* yang digunakan dalam pengembangan perangkat lunak. *Scrum* dirancang untuk membantu tim bekerja secara kolaboratif dan adaptif dalam mengelola proyek yang kompleks. Dalam proyek ini, *Scrum* diterapkan sebagai metodologi pengembangan agar tim dapat bekerja secara efisien dan fleksibel dalam mengembangkan backend aplikasi bahasa Lampung.

Dalam *Scrum*, pekerjaan diatur dalam siklus singkat yang disebut sprint, biasanya berlangsung selama dua hingga empat minggu. Setiap sprint diawali dengan perencanaan sprint di mana tim menentukan fitur-fitur apa saja yang akan dikerjakan berdasarkan product backlog, daftar prioritas fitur yang harus dikembangkan. Pada akhir sprint, dilakukan sprint review untuk meninjau hasil pekerjaan, diikuti dengan retrospektif untuk mengevaluasi proses dan perbaikan yang bisa dilakukan di sprint berikutnya.

Scrum, sebagai salah satu pendekatan dalam metodologi *Agile*, memiliki keunggulan dalam memecah proyek besar menjadi iterasi pendek dan terfokus. Dalam konteks *SDLC (Software Development Life Cycle)*, *Scrum* memungkinkan tim untuk secara terus-menerus mengembangkan perangkat lunak dalam bentuk *increment* yang dapat diuji dan digunakan setelah setiap sprint. Keunggulan utama *Scrum* dibandingkan dengan pendekatan *Agile* lain secara umum adalah adanya struktur yang lebih jelas dengan peran-peran yang spesifik, seperti *Scrum Master*, *Product Owner*, dan tim pengembang, yang mendukung koordinasi dan komunikasi yang lebih baik antar anggota tim. *Scrum* memastikan bahwa setiap iterasi menghasilkan fitur yang siap pakai dan relevan dengan kebutuhan pengguna atau pemangku kepentingan, sehingga memungkinkan pengembangan yang lebih responsif terhadap perubahan.

Berdasarkan jurnal *A Systematic Review of SCRUM in Software Development* (JOIV, 2021), Scrum sangat efektif dalam menangani proyek perangkat lunak yang kompleks, karena sifatnya yang terstruktur dan memungkinkan penyesuaian yang cepat terhadap kebutuhan yang berubah. *Scrum* lebih unggul dibandingkan dengan pendekatan Agile yang lebih umum karena memberikan pedoman yang lebih terperinci dalam manajemen proyek, serta fokus pada pengembangan secara iteratif dan peningkatan berkelanjutan. Metode ini tidak hanya mempercepat proses pengembangan tetapi juga meningkatkan kolaborasi dalam tim, sehingga memungkinkan deteksi masalah lebih cepat dan memberikan solusi yang lebih tepat. Dengan demikian, *Scrum* menjadi pilihan yang lebih tepat dalam pengembangan sistem seperti backend aplikasi bahasa Lampung yang membutuhkan pengelolaan proyek yang lebih terorganisir dan efisien [22].

2.14 Application Programming Interface (API)

API, atau *Application Programming Interface*, adalah sebuah mekanisme yang memungkinkan dua sistem berbeda untuk saling berkomunikasi dan bertukar data. Dalam konteks pengembangan perangkat lunak, *API* merupakan elemen penting dalam pengembangan aplikasi modern, terutama untuk integrasi antara front-end dan back-end. *API* biasanya digunakan untuk menghubungkan aplikasi *front-end* dengan *back-end* [23], di mana data dikirim dan diterima melalui protokol *HTTP* dalam format *JSON* atau *XML*.

API REST (Representational State Transfer) adalah salah satu arsitektur yang paling umum digunakan dalam pengembangan *API*. *RESTful API* memungkinkan komunikasi yang lebih sederhana antara klien (misalnya aplikasi web atau mobile) dengan server melalui metode *HTTP* seperti *GET*, *POST*, *PUT*, dan *DELETE*. Setiap permintaan *API* membawa informasi tentang data yang dibutuhkan, dan server merespon dengan menyediakan data atau melakukan operasi sesuai permintaan.

2.15 Penelitian Terkait

Terkait penelitian ini, Muljono, Nur, Junta, Raden, Valentina, & Bayu melakukan penelitian yang berjudul *“Perancangan Aplikasi kamus istilah Jawa berbasis Android sebagai Upaya Pelestarian Budaya jawa”*. Penelitian tersebut bertujuan untuk merancang aplikasi kamus istilah jawa, yang memiliki fitur interaktif yang dapat digunakan oleh penyandang tunanetra, selain itu pengembangan fitur pada penelitian ini juga menggunakan beberapa *role* untuk pengelolaan data istilah jawa [24].

Penelitian lainnya yaitu, Riri Ardyaningtyas melakukan penelitian dengan judul *“Basabali wiki penggunaan Website Kamus Digital sebagai Langkah untuk melestarikan Bahasa Bali”* [25]. Penelitian tersebut menjelaskan tentang pentingnya pemanfaatan website untuk melestarikan Bahasa, aksara, dan literatur kebudayaan bali.

Selanjutnya penelitian yang dipublikasikan dalam prosiding *The 7th International Conference on Information Technology* yang berjudul *“Performance Evaluation of Programming Languages as API Services for Cloud Environments: A Comparative Study of PHP, Python, Node.js and Golang”*. Menjelaskan bahwa Golang dan Node.js memiliki performa yang hampir setara dalam skenario dengan satu pengguna. Namun, ketika jumlah pengguna yang bersamaan meningkat, Golang menunjukkan kemampuan yang lebih unggul dalam menangani permintaan dibandingkan dengan bahasa lain. Dalam pengembangan layanan *API* untuk kebutuhan Internet, kemampuan mendukung proses bersamaan adalah hal yang penting. Oleh karena itu, hasil eksperimen menunjukkan bahwa Golang merupakan pilihan yang lebih baik untuk membangun layanan *API* di lingkungan Cloud dibandingkan PHP, Node.js, dan Python [7].

Selanjutnya penelitian terkait lainnya yang berjudul *“Analisa Performa Backend Framework: Studi Komparasi Framework Golang dan Node.js”* Menjelaskan bahwa Golang adalah Bahasa pemrograman yang mempunyai waktu respon yang cepat dibandingkan dengan Node.js Ketika melakukan perhitungan kriptografi yang

kompleks, Golang juga lebih efisien dalam penggunaan sumber daya seperti cpu, dibandingkan dengan Node.js [9].

Selanjutnya penelitian terkait lainnya dilakukan oleh para peneliti dalam jurnal *A Systematic Review of SCRUM in Software Development* (JOIV, 2021), yang menyoroti penerapan metodologi *Scrum* dalam pengembangan perangkat lunak, termasuk dalam pelestarian bahasa dan budaya. Penelitian tersebut menjelaskan bahwa penggunaan *Scrum* dapat meningkatkan efisiensi dalam mengelola proyek perangkat lunak yang kompleks dan memungkinkan tim untuk bekerja secara iteratif dan fleksibel. Pendekatan ini juga memungkinkan perbaikan berkelanjutan dengan setiap iterasi, memastikan bahwa hasilnya selalu relevan dan dapat disesuaikan dengan kebutuhan pengguna. Dalam konteks pelestarian bahasa daerah seperti aplikasi kamus bahasa Lampung, penggunaan *Scrum* dapat membantu pengembangan sistem secara lebih terstruktur, dengan memperhatikan kebutuhan dan masukan dari masyarakat atau pemangku kepentingan secara terus-menerus. Penelitian ini juga sejalan dengan pentingnya pengembangan aplikasi berbasis teknologi untuk mendokumentasikan dan melestarikan bahasa daerah. Seperti yang dijelaskan dalam penelitian JOIV, penerapan *Scrum* dapat mempercepat proses pengembangan aplikasi dan mempermudah kolaborasi antar tim dalam menangani tantangan teknis dan kebutuhan pengguna yang berkembang. Dengan pendekatan ini, aplikasi kamus bahasa Lampung diharapkan dapat lebih cepat diselesaikan dan memiliki fitur yang lebih baik, sesuai dengan ekspektasi penggunanya [22].

Selanjutnya penelitian dalam jurnal berjudul "*Game Development with Scrum Methodology*" membahas penerapan *Scrum* dalam pengembangan perangkat lunak, khususnya dalam pengembangan game. Penelitian ini menunjukkan bahwa *Scrum*, sebagai salah satu metode *Agile*, dapat meningkatkan kolaborasi tim, mempercepat siklus pengembangan, serta memberikan fleksibilitas dalam menghadapi perubahan kebutuhan proyek. Studi ini juga menekankan pentingnya perencanaan dan pengujian yang terstruktur untuk menghasilkan produk berkualitas tinggi. Relevansi penelitian ini terhadap penelitian yang sedang dilakukan adalah penerapan metodologi *Scrum*

untuk meningkatkan efisiensi dalam pengembangan sistem *backend*, memastikan kolaborasi tim yang lebih baik, serta mempercepat proses penyelesaian fitur-fitur yang dibutuhkan [20].

Selanjutnya penelitian yang dilakukan oleh Arifin (2018) dalam perancangan *RESTful API* pada sistem informasi akademik berbasis web, ditemukan bahwa penerapan *API REST* membantu meningkatkan efisiensi komunikasi antara komponen-komponen sistem yang terpisah. Selain itu, *API* juga memberikan fleksibilitas untuk integrasi antar-platform, baik di sisi web maupun mobile, yang sangat relevan dengan sistem Aksara Lampung yang juga akan diakses melalui antarmuka web dan mobile.

Selanjutnya Penelitian oleh Anderson dan Gapon (2019) menegaskan bahwa *Golang* sangat cocok untuk pengembangan aplikasi backend karena memiliki kemampuan *concurrency* yang baik dan memanfaatkan fitur *goroutines* untuk menjalankan proses secara bersamaan tanpa menghabiskan banyak memori. Ini memberikan efisiensi yang lebih baik dibandingkan dengan bahasa pemrograman lainnya, terutama ketika menangani banyak permintaan dari klien secara simultan, yang sangat penting dalam pengembangan sistem seperti Aksara Lampung.

III. METODE PENELITIAN

3.1 Waktu dan Tempat

Pelaksanaan penelitian dilakukan pada bulan Desember 2024 sampai dengan bulan Mei 2025 yang dilakukan di universitas lampung, Adapun jadwal penelitian sebagai berikut:

Tabel 3. 1 jadwal penelitian

No	Aktivitas	Des	Jan	Feb	Mar	April	Mei
1	Analisis kebutuhan						
2	Perencanaan						
3	Product Backlog						
4	Sprint Backlog						
5	Sprint						
6	Sprint Review						
8	Restropective						
7	Penulisan Laporan						

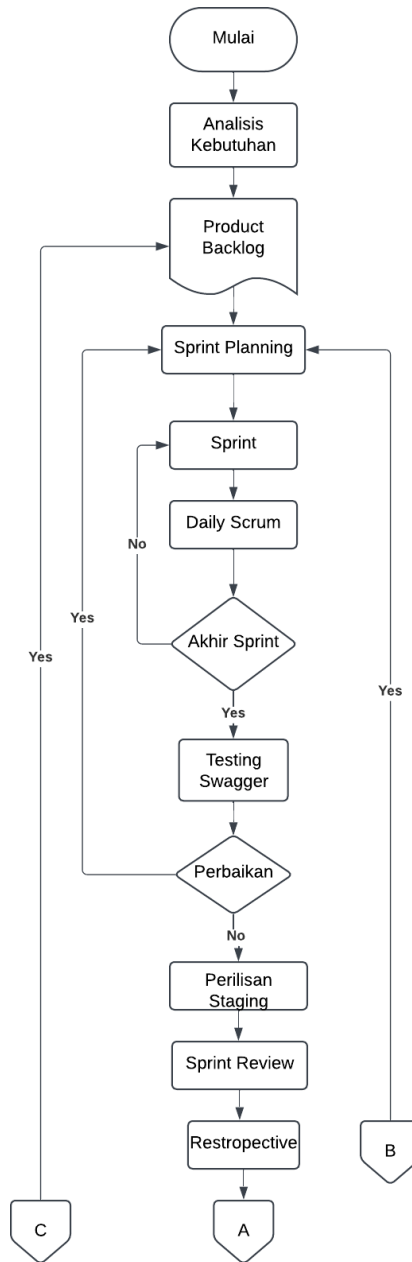
3.2 Alat dan Bahan

Beberapa alat dan bahan yang digunakan dalam penelitian ini adalah sebagai berikut:

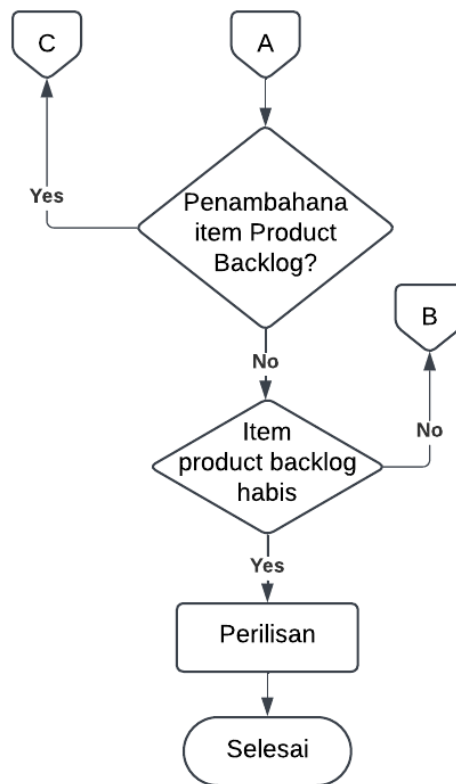
Tabel 3. 2 Alat yang digunakan dalam penelitian

No	Nama Alat	Spesifikasi	Deskripsi
1	Laptop Lenovo Ideapad Slim 3	AMD Ryzen 5 dengan RAM 8GB dan sistem operasi Windows 11	Perangkat keras yang digunakan untuk melakukan proses <i>development</i>
2	Visual Studio Code	Versi 1.95.3	Perangkat lunak yang digunakan untuk proses <i>development</i>
3	Postman	Versi 9.1.3	Perangkat lunak yang digunakan untuk melakukan testing <i>API</i>

3.3 Tahapan Penelitian



Gambar 3. 1 Metode Pengembangan *Scrum*



Gambar 3. 2 Metode Pengembangan Scrum (lanjutan)

3.3.1 Analisis Kebutuhan

Mengumpulkan data dari berbagai sumber melalui metode observasi dan studi literatur sebagai bahan untuk penyusunan Spesifikasi Kebutuhan Perangkat Lunak (SKPL). Data yang dikumpulkan dikompilasi untuk digunakan sebagai acuan dalam menentukan kebutuhan pengguna dari sistem yang dikembangkan.

Selain itu, penelitian ini juga menggunakan metode observasi untuk pengumpulan data. Metode observasi menjadi dasar penentuan isu yang dibahas dalam penelitian ini yang mengangkat isu punahnya bahasa Lampung. Kemudian, observasi juga digunakan untuk menganalisis situs web serupa seperti Kasa Talk (Kamus Bahasa Sasak) dan Kamus Bahasa Lampung yang dikembangkan oleh Balai Bahasa Provinsi Lampung.

3.3.1.1 Analisis Aplikasi Serupa

Dalam tahap analisis kebutuhan, dilakukan studi terhadap aplikasi serupa yang sudah ada untuk mengevaluasi kelebihan, kekurangan, peluang, dan tantangan yang dihadapi. Tujuan dari analisis ini adalah untuk mengarahkan pengembangan *Restful Api* kamus Bahasa Lampung agar dapat menghadirkan fitur yang lebih relevan, meningkatkan pengalaman pengguna, serta memanfaatkan peluang dalam bersaing dengan produk sejenis. Selain itu, analisis ini juga membantu mengidentifikasi aspek yang perlu diperbaiki atau ditambahkan, sekaligus menjadi dasar strategis dalam merancang solusi yang lebih tepat.

a. Kamus Lampung-Indonesia KBPL

Aplikasi ini unggul dalam menyediakan data yang cukup lengkap serta mendukung pencarian berdasarkan kata tertentu. Namun, kekurangannya terletak pada sifat data yang statis dan jarang diperbarui, serta tampilan yang kurang menarik. Peluang utama yang dapat dimanfaatkan adalah peningkatan desain antarmuka dan pemberian akses bagi masyarakat untuk berkontribusi dalam memperkaya data. Sementara itu, tantangan yang dihadapi mencakup persaingan dengan aplikasi serupa yang lebih populer serta risiko menurunnya relevansi data jika tidak dilakukan pembaruan secara rutin.

b. Kamus Bahasa Lampung

Keunggulan aplikasi ini meliputi fitur pembelajaran aksara, budaya, dan sastra Lampung, serta kemampuan menerjemahkan teks hingga kalimat. Kelemahannya termasuk alur penambahan kata yang tidak jelas dan hasil terjemahan yang masih per

kata. Peluang aplikasi ini terletak pada pengembangan fitur pembelajaran interaktif dan penggunaan teknologi AI untuk meningkatkan akurasi. Namun, ancaman berasal dari persaingan aplikasi dengan teknologi lebih canggih dan segmentasi pasar yang terbatas.

c. Kamus Lampung (Aplikasi Android)

Aplikasi ini unggul dalam menampilkan daftar kata secara lengkap dan memiliki fitur untuk menyimpan kata favorit. Namun, kekurangannya mencakup iklan yang mengganggu, kurangnya kategori dialek, dan tampilan aplikasi yang tidak menarik. Peluangnya adalah memperbaiki tampilan dan menambahkan fitur kategori dialek untuk memperluas fungsionalitas. Ancaman datang dari minimnya daya tarik di tengah kompetisi aplikasi serupa.

3.3.1.2 Kebutuhan Fungsional

Kebutuhan fungsional menggambarkan fungsionalitas apa saja yang akan dimiliki oleh sistem. Kebutuhan fungsional mencakup fitur-fitur yang akan diterapkan untuk memenuhi kebutuhan pengguna. Berikut adalah tabel kebutuhan fungsionalitas sistem Kamus Bahasa Lampung:

Tabel 3. 3 Kebutuhan Fungsional

No	Kebutuhan	Penjelasan
1	Mencari Kata	Sistem dapat mencari kata dari Bahasa Lampung ke Bahasa Indonesia dan sebaliknya
2	Melihat hasil pencarian kata	Sistem dapat menampilkan hasil pencarian kata termasuk aksara, dialek, wilayah penuturan, dan kontributor.
3	Melaporkan kata	Sistem mengizinkan pengguna untuk melaporkan kata

Tabel 3. 4 Kebutuhan Fungsional (lanjutan)

No	Kebutuhan	Penjelasan
4	Melihat blog	Sistem menampilkan blog yang ada
5	Daftar akun	Sistem menyediakan fitur pendaftaran akun untuk pengguna baru.
6	Login akun	Pengguna dapat masuk ke dalam sistem dengan kredensial yang dimiliki
7	Lupa Kata Sandi	Sistem menyediakan fitur lupa sandi jika pengguna tidak dapat masuk ke sistem
8	Atur Ulang Kata Sandi	Sistem memiliki mekanisme untuk mengatur ulang kata sandi pengguna.
9	Melihat riwayat penambahan kata	Sistem mengizinkan pengguna dapat melihat riwayat kata yang telah mereka tambahkan ke sistem.
10	Sunting Profil	Pengguna dapat menyunting informasi profil seperti nama, email, dan domisili
11	Kelola Draf	Pengguna dapat mengelola draf kata yang disimpan seperti menghapus, menyunting, dan mengirimkan kata untuk ditinjau
12	Menambahkan kata baru	Pengguna dapat menambahkan kata baru ke dalam sistem.
13	Meninjau Kata	Peninjau dapat memeriksa kata yang ditambahkan oleh kontributor.

Tabel 3. 5 Kebutuhan Fungsional (lanjutan 1)

14	Meninjau Laporan	Peninjau dapat meninjau laporan yang dikirimkan oleh pengguna lain.
15	Melihat riwayat tinjauan	Peninjau dapat melihat riwayat laporan dan kata yang sudah mereka tinjau.
16	Melihat daftar kata	Admin dapat melihat seluruh kata yang ada dalam sistem.
17	Kelola data blog	Admin dapat menambah, menyunting, dan menghapus data blog.
18	Kelola data pengguna	Admin dapat menambah, menyunting, dan menghapus data pengguna.
19	Kelola data dialek dan wilayah	Admin dapat mengelola data terkait dialek dan wilayah untuk mendukung penerjemahan kata.

3.3.1.4 Kebutuhan Non Fungsional

Kebutuhan non-fungsional adalah kebutuhan yang menggambarkan kualitas dan karakteristik sistem yang tidak terkait langsung dengan fungsi atau fitur sistem, tetapi lebih kepada aspek kinerja, keamanan, dan keandalan sistem. Berikut merupakan daftar kebutuhan non fungsional sistem:

Tabel 3. 6 Kebutuhan Non Fungsional

No	Parameter	Penjelasan
1	Ketersediaan	1. Sistem harus dapat diakses 24 jam dan harus memiliki uptime minimal 99,5%.
2	Skalabilitas	1. Mendukung caching untuk mengurangi beban pada database
3	Keamanan	1. Menggunakan autentikasi dan otorisasi untuk akses API. 2. Menerapkan rate limiting untuk mencegah penyalahgunaan API.
4	Kinerja	1. Backend harus merespons permintaan dalam waktu kurang dari 2000ms.
5	Portabilitas	1. Dapat dijalankan menggunakan container (Docker) untuk kemudahan deployment. 2. Backend dapat dijalankan di berbagai lingkungan seperti DigitalOcean, AWS, atau VPS lainnya.
6	Kompatibilitas	1. Menggunakan format data JSON untuk komunikasi antar sistem.

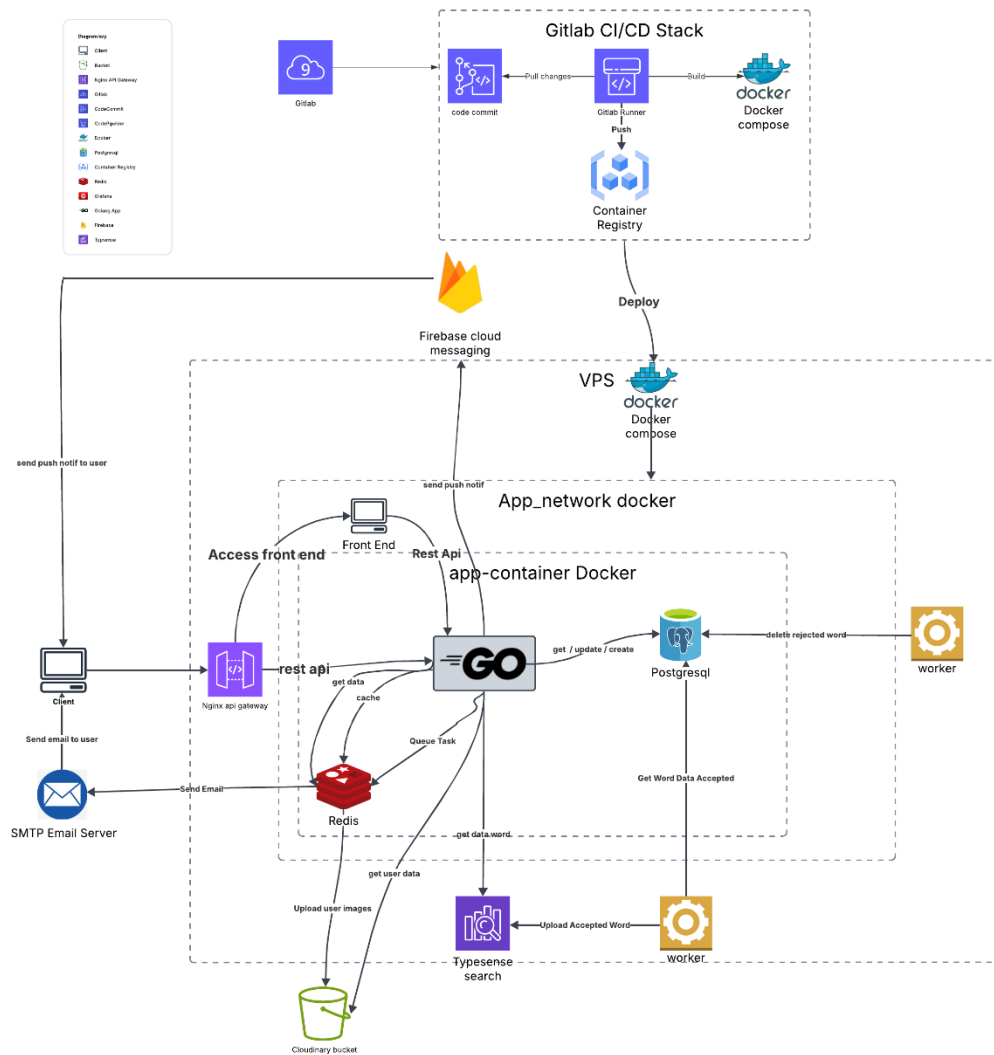
Tabel 3. 7 Kebutuhan Non-Fungsional (lanjutan)

No	Parameter	Penjelasan
7	Pemeliharaan	1. Restful api memiliki dokumentasi swagger.

3.3.2 Perancangan sistem

3.3.2.1 Arsitektur sistem

Arsitektur sistem merupakan representasi desain infrastruktur dari aplikasi yang dikembangkan untuk mendukung fitur pencarian, dan manajemen data bahasa Lampung. Sistem ini dibangun dengan pendekatan berbasis layanan menggunakan teknologi monolitik dan terintegrasi dengan berbagai komponen eksternal.



Gambar 3. 3 Arsitektur Sistem

Berdasarkan dari gambar 3.3 Arsitektur Sistem, terdapat beberapa komponen yang digunakan sebagai berikut:

Tabel 3. 8 Deskripsi Arsitektur Sistem

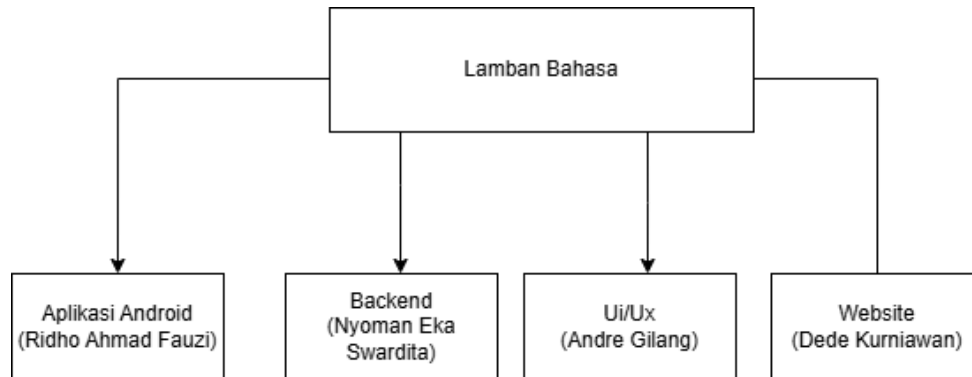
No	Komponen	Deskripsi
1	Client	Client adalah perangkat pengguna yang berinteraksi dengan sistem melalui aplikasi web dan mobile.

No	Komponen	Deskripsi
		Permintaan dari client akan dikirimkan melalui API Gateway sebelum diteruskan ke server backend.
2	<i>Nginx</i> <i>Api</i> <i>Gateway</i>	Nginx API Gateway bertindak sebagai pintu gerbang utama untuk semua permintaan API dari client. Komponen ini berfungsi untuk melakukan routing permintaan, dan pengaturan batasan akses.
3	<i>Backend</i> <i>(golang)</i>	Backend dibangun menggunakan bahasa pemrograman Golang dengan framework Fiber. Backend bertanggung jawab untuk menangani logika bisnis, proses pencarian kata,, serta penyimpanan dan pengelolaan data. Komponen backend terhubung dengan berbagai layanan eksternal seperti database, cache, firebase, dan object storage.
4	Database	Database yang digunakan adalah PostgreSQL untuk menyimpan data kata bahasa Lampung, arti terjemahan, dan informasi pengguna. Komunikasi antara backend dan database menggunakan ORM (Object Relational Mapping) GORM.
5	Redis	Redis digunakan sebagai cache penyimpanan sementara untuk mempercepat proses pengambilan data blog, statistic, top kontributor dan <i>queue task</i> .
6	Cloudinary Object Storage	Object storage digunakan untuk menyimpan file gambar atau media lainnya yang diunggah oleh pengguna.

No	Komponen	Deskripsi
7	Firebase Cloud Messaging	Firebase cloud messaging digunakan untuk mengirimkan push notifikasi kepada pengguna yang melakukan instalasi aplikasi.
8	CI/CD	Proses Continuous Integration dan Continuous Deployment (CI/CD) dilakukan menggunakan GitLab CI/CD untuk otomatisasi build, dan deployment aplikasi.
9	Containerization	Aplikasi backend dikemas dalam container menggunakan Docker untuk memastikan portabilitas dan konsistensi lingkungan pengembangan hingga produksi.
10	Infrastructure as a Service (IaaS)	Server aplikasi di-deploy menggunakan DigitalOcean sebagai layanan cloud untuk menjalankan aplikasi dan menyimpan data.
11	Worker	Sebuah cronjob untuk melakukan background job dengan menjalankan program untuk mengirim data kata pada typesense search engine
12	Typesense	Sebuah database search <i>open source</i> untuk menyimpan data kata, serta memiliki fitur pencarian kata.

3.3.3 Perencanaan

Pada tahap ini, dilakukan pemetaan tim dan pembagian tugas berdasarkan data dari analisis kebutuhan. Metode pengembangan yang diterapkan adalah agile development dengan kerangka kerja Scrum. Hasil analisis kebutuhan digunakan sebagai dasar dalam menyusun user stories serta menentukan teknologi yang akan digunakan selama pengembangan. User stories yang telah dibuat kemudian disusun dalam backlog, yang nantinya akan diambil dan dikerjakan dalam setiap sprint.



Gambar 3. 4 Bagan Dari Tim

3.3.4 Product Backlog

Pada tahap ini, *product backlog* disusun berdasarkan kebutuhan penggunaan sistem. Product backlog memuat daftar tugas yang mencakup fitur-fitur yang harus dikembangkan. Product backlog bersifat fleksibel, sehingga dapat diperbarui sepanjang proses pengembangan.

3.3.5 Sprint Backlog

Sprint backlog adalah subset dari product backlog yang dipilih untuk dikerjakan dalam satu sprint (periode kerja yang ditetapkan, biasanya 2-4 minggu). Sprint backlog mencakup item-item yang telah diprioritaskan dan dirinci, serta tugas-tugas yang dibutuhkan untuk menyelesaikan item tersebut. Sprint backlog menjadi panduan bagi tim pengembang dalam satu sprint.

3.3.6 Sprint

Sprint adalah periode waktu tertentu, biasanya antara 1 hingga 4 minggu, di mana tim *Scrum* menyelesaikan serangkaian item dari *sprint backlog*. Setiap *sprint* memiliki tujuan atau *sprint goal* yang jelas, dan hasil akhir dari sprint biasanya berupa potongan fungsional dari produk yang siap untuk digunakan atau diujikan.

3.3.7 Sprint Review

Sprint review adalah pertemuan di akhir sprint di mana tim Scrum menampilkan hasil kerja mereka kepada Product Owner dan pemangku kepentingan. Pada tahap ini, tim memperlihatkan increment (penambahan fungsionalitas produk) yang telah selesai, mendapatkan umpan balik, dan mendiskusikan potensi perubahan atau perbaikan untuk sprint berikutnya.

3.3.8 Retrospective

Retrospective adalah pertemuan internal tim yang diadakan setelah sprint review untuk mengevaluasi proses kerja selama sprint berlangsung. Tujuan dari sprint retrospective adalah untuk mengidentifikasi hal-hal yang berjalan dengan baik, hal-hal yang perlu diperbaiki, dan langkah-langkah konkret yang dapat diambil untuk meningkatkan efektivitas tim di sprint berikutnya. Diskusi dalam retrospective mencakup aspek teknis, kolaborasi tim, dan hambatan yang dihadapi selama sprint, sehingga proses pengembangan perangkat lunak dapat terus ditingkatkan.

V. KESIMPULAN DAN SARAN

5.1 Kesimpulan

Adapun Kesimpulan dari penelitian ini adalah sebagai berikut:

1. Sistem backend yang dibangun menggunakan bahasa pemrograman Golang berhasil menyediakan layanan pencarian dan manajemen kosakata bahasa Lampung secara digital. Sistem ini mendukung proses pengelolaan data bahasa Lampung yang melibatkan komunitas dalam menambah dan memverifikasi data kosakata.
2. Sistem manajemen peran dalam backend telah berhasil diimplementasikan dengan mekanisme autentikasi dan otorisasi yang memastikan akses sesuai dengan fungsi masing-masing peran. Admin memiliki wewenang dalam pengelolaan pengguna, kontributor dapat menambahkan kosakata, sementara peninjau bertanggung jawab dalam proses verifikasi kata.
3. Hasil pengujian performa menggunakan JMeter menunjukkan bahwa sistem mampu menangani 2000 virtual user dengan waktu respon rata-rata 365.60 ms pada localhost dan 1128.50 ms pada VPS tanpa error, serta mampu menangani 10000 virtual user dengan waktu respon rata-rata 6430.77 ms pada localhost dan 8659.61 ms pada VPS. Hal ini menunjukkan bahwa sistem memiliki tingkat stabilitas yang baik dalam menerima permintaan dalam jumlah besar.

5.2 Saran

Berdasarkan hasil penelitian dan pengembangan, terdapat beberapa saran untuk pengembangan lebih lanjut:

1. Menambahkan fitur load balancing dengan menerapkan distributed system yang memungkinkan pembagian permintaan pengguna ke beberapa server secara

otomatis, sehingga meningkatkan skalabilitas dan ketersediaan sistem. Penerapan teknologi ini dapat dilakukan dengan menggunakan proxy seperti Nginx atau layanan cloud yang mendukung load balancing untuk memastikan performa sistem tetap stabil saat menerima permintaan dalam jumlah besar.

2. Melakukan pengujian performa dengan JMeter pada environment yang lebih optimal, seperti penggunaan server dengan spesifikasi lebih tinggi dan jaringan yang lebih stabil, untuk memperoleh hasil pengujian yang lebih relevan dan representatif terhadap kondisi nyata.

DAFTAR PUSTAKA

DAFTAR PUSTAKA

- [1] E. Andina, A. Keparlemenan, B. Keahlian, D. Ri, and J. G. Subroto, “Implementasi dan Tantangan Revitalisasi Bahasa Daerah di Provinsi Lampung
Implementation and Challenges of Revitalizing Local Languages in Lampung
Province,” *J. Masal. Sos. |*, vol. 14, no. 1, pp. 2614–5863, 2023, [Online].
Available:
<https://doi.org/10.22212/aspirasi.v14i1.3859>[linkonline:http://jurnal.dpr.go.id/index.php/aspirasi/index](http://jurnal.dpr.go.id/index.php/aspirasi/index)
- [2] Z. Abidin and P. Permata, “Pengaruh Penambahan Korpus Paralel Pada Mesin
Penerjemah Statistik Bahasa Indonesia Ke Bahasa Lampung Dialek Nyo,” *J.
Teknoinfo*, vol. 15, no. 1, p. 13, 2021, doi: 10.33365/jti.v15i1.889.
- [3] H. Jackson, *Lexicography: An Introduction*. Psychology Press, 2002, 2002.
- [4] L. Zgusta, “Manual of lexicography,” *Man. Lexicogr.*, 1971, doi:
10.1515/9783111349183.
- [5] P. Rob, “Go at Google: Language Design in the Service of Software
Engineering.” Accessed: Nov. 24, 2024. [Online]. Available: <https://go.dev>
- [6] S. bin Uzayr, *Mastering GoLang*. 2022. doi: 10.1201/9781003310457.
- [7] T. Tanadechopon and B. Kasemsontitum, “Performance Evaluation of
Programming Languages as API Services for Cloud Environments: A
Comparative Study of PHP, Python, Node.js and Golang,” *7th Int. Conf. Inf.
Technol. InCIT* 2023, pp. 293–297, 2023, doi:
10.1109/InCIT60207.2023.10413079.

- [8] M. R. S. Alfarizi *et al.*, “Menggali Bahasa Pemrograman Populer: Karakteristik Utama dan Penggunaan yang Luas,” *Karimah Tauhid*, vol. 2, no. 4, pp. 1191–1197, 2023, [Online]. Available: <https://ojs.unida.ac.id/karimahtauhid/article/view/8816>
- [9] Suwarno and A. Putri Yulandi, “Analisis Performa Backend Framework: Studi Komparasi Framework Golang dan Node.js,” *J. Ris. Sist. Inf. Dan Tek. Inform.*, vol. 8, pp. 155–168, 2023, [Online]. Available: <https://tunasbangsa.ac.id/ejurnal/index.php/jurasik>
- [10] A. Mohhamad, “Applying Layered Architecture in Golang.” Accessed: Nov. 24, 2024. [Online]. Available: <https://www.clear-cut.io/golang/applying-layered-architecture-in-golang/>
- [11] The PostgreSQL Global Development Group, “What Is PostgreSQL?,” *Postgresql.Org*. [Online]. Available: <https://www.postgresql.org/about/>
- [12] N. Ferdiansyah, D. A. Rahayu, and R. Permala, “Comparison of Postgresql, Mariadb and MongoDB Capabilities in Processing Lapan Satellite Ais Data,” no. December, pp. 231–238, 2019, doi: 10.30536/p.siptekgan.2019.v23.23.
- [13] A. D. Praba and M. Safitri, “Studi Perbandingan Performansi Antara Mysql Dan Postgresql,” *J. Khatulistiwa Inform.*, vol. 8, no. 2, pp. 88–93, 2020, doi: 10.31294/jki.v8i2.8851.
- [14] M. I. Zulfa, A. Fadli, and A. W. Wardhana, “Application caching strategy based on in-memory using Redis server to accelerate relational data access,” *J. Teknol. dan Sist. Komput.*, vol. 8, no. 2, pp. 157–163, 2020, doi: 10.14710/jtsiskom.8.2.2020.157-163.
- [15] A. Alperly and M. A. F. Ridha, “Implementasi CI/CD Dalam Pengembangan Aplikasi Web Menggunakan Docker dan Jenkins,” *Appl. Bus. Eng. Conf.*, pp. 287–296, 2021.

- [16] F. Sethi, “Automating Software Code Deployment Using Continuous Integration and Continuous Delivery Pipeline for Business Intelligence Solutions,” *Int. J. Innov. Sci. Res. Rev.*, vol. 02, no. 10, pp. 445–449, 2020, doi: 10.22541/au.160373745.57814465/v1.
- [17] S. Wellem Taju, S. Richard Pungus, R. Junius Lontaan, R. Rotikan, and M. Timothy Tombeng, “Mengakselerasi Keterampilan Rekayasa Perangkat Lunak: Peranan DevOps, SDLC, dan CI/CD dalam Meningkatkan Kompetensi Siswa SMK N 1 Pusomaen,” *Servitium Smart J.*, vol. 2, no. 1, pp. 119–128, 2023, doi: 10.31154/servitium.v2i1.24.
- [18] Typesense, “What is Typesense?” [Online]. Available: <https://typesense.org/docs/overview/what-is-typesense.html>
- [19] E. K. D. P. I. K. Najibullah, “Perbandingan performa Load Balancing pada Nginx dan Gobetween,” no. January, pp. 0–3, 2020.
- [20] D. P. Kristiadi, F. Sudarto, D. Sugiarto, R. Sambera, H. L. H. S. Warnars, and K. Hashimoto, “Game Development with Scrum methodology,” *2019 Int. Congr. Appl. Inf. Technol. AIT 2019*, 2019, doi: 10.1109/AIT49014.2019.9144963.
- [21] N. Gupta, H. Sharma, S. Kumar, A. Kumar, and R. Kumar, “A Comparative Study of Implementing Agile Methodology and Scrum Framework for Software Development,” *Proc. 2022 11th Int. Conf. Syst. Model. Adv. Res. Trends, SMART 2022*, pp. 1088–1092, 2022, doi: 10.1109/SMART55829.2022.10047477.
- [22] E. P. Wonohardjo, R. F. Sunaryo, Y. Sudiyono, N. Surantha, and Suharjito, “A systematic review of scrum in software development,” *Int. J. Informatics Vis.*, vol. 3, no. 2, pp. 108–112, 2019, doi: 10.30630/joiv.3.2.167.
- [23] T. Bratakusuma, I. U. Azmi, and S. Ayuningtiyas, “Pengembangan Back End Pada Aplikasi Alat Tulis Kantor Bank Indonesia Perwakilan Purwokerto

Menggunakan Nodejs,” *Semin. Nas. Inov. dan Pengemb. Teknol. Terap. Cilacap*, pp. 119–127, 2022, [Online]. Available: <https://ejournal.pnc.ac.id/index.php/senovtek>

- [24] M. Muljono, N. Rokhman, J. Zeniarja, R. A. Nugroho, V. W. Suryaningtyas, and B. Aryanto, “Perancangan Aplikasi Kamus Istilah Jawa Berbasis Android sebagai Upaya Pelestarian Budaya Jawa,” *ANDHARUPA J. Desain Komun. Vis. Multimed.*, vol. 9, no. 4, pp. 553–564, 2024, doi: 10.33633/andharupa.v9i4.9282.
- [25] R. Ardyaningtyas, “Basabali Wiki, Penggunaan Website Kamus Digital Sebagai Langkah untuk Melestarikan Bahasa Bali,” *J. Educ. Instr.*, vol. 6, no. 1, pp. 205–222, 2023, doi: 10.31539/joeai.v6i1.6052.