**IMPLEMENTASI ARSITEKTUR *MICROSERVICES* DALAM PERANCANGAN APLIKASI WEBSITE EDUKASI BAHASA JEPANG UNTUK ANAK DIUSIA DINI**

##### SKRIPSI

Sebagai Salah Satu Syarat Untuk Memperoleh Gelar Sarjana Komputer

Disusun Oleh :

Yahya Pratama

11200910000027

**PROGRAM STUDI TEKNIK INFORMATIKA**

**FAKULTAS SAINS DAN TEKNOLOGI**

**UNIVERSITAS ISLAM NEGERI SYARIF HIDAYATULLAH JAKARTA**

**2023 M / 1444 H**

# **LEMBAR PERSETUJUAN**

Proposal Skripsi berjudul “IMPLEMENTASI ARSITEKTUR *MICROSERVICES* DALAM PERANCANGAN APLIKASI WEBSITE EDUKASI BAHASA JEPANG UNTUK ANAK DIUSIA DINI” yang ditulis oleh Yahya Pratama telah didiskusikan dengan calon pembimbing. Diajukan untuk memenuhi salah satu persyaratan memperoleh gelar sarjana komputer (S.Kom).

|  |  |  |
| --- | --- | --- |
| Calon Dosen Pembimbing 1 |  | Calon Dosen Pembimbing 2 |
| <tanda tangan> |  | <tanda tangan> |
| <nama Dosen Pembimbing> |  | <nama Dosen Pembimbing 2> |
| NIP. <nip Dosen Pembimbing 2> |  | NIP. <nip Dosen Pembimbing 2> |

# **KATA PENGANTAR**

Puji syukur penulis panjatkan kepada Tuhan Yang Maha Esa Allah Swt. atas berkat rahmat dan karunia-nya, penulis dapat menyelesaikan laporan skripsi ini. Tidak lupa, shalawat serta salam semoga tetap tercurah kepada Nabi Muhammad SAW.

Penelitian laporan skripsi ini dibuat dalam rangka memenuhi salah satu syarat pembuatan tugas akhir berupa skripsi untuk syarat mencapai gelar Sarjana Komputer Program Studi Teknik Informatika Fakultas Sains dan Teknologi Universitas Islam Negeri Syarif Hidayatullah Jakarta.

Dengan selesainya penelitian laporan skripsi yang berjudul

“IMPLEMENTASI ARSITEKTUR *MICROSERVICES* DALAM PERANCANGAN APLIKASI WEBSITE EDUKASI BAHASA JEPANG UNTUK ANAK DIUSIA DINI”, penulis ingin mengucapkan terima kasih yang sebesar-besarnya kepada:

1. Allah Swt. yang telah memberikan rahmat dan karunia-nya.
2. Orang tua dan keluarga penulis, yang senantiasa memberikan do’a, dukungan, dan cinta selama proses penulisan penelitian laporan skripsi ini.
3. Bapak Prof. Asep Saepudin Jahar MA Ph.D, selaku Rektor Universitas Islam Negeri Syarif Hidayatullah Jakarta.
4. Bapak Husni Teja Sukmana, S.T., M.Sc, Ph.D, selaku Dekan Fakultas Sains dan Teknologi.
5. Ibu Dewi Khairani, M.Sc, selaku Ketua Program Studi Teknik Informatika Fakultas Sains dan Teknologi.
6. <nama dospem 1>, selaku Dosen Pembimbing I, dan <nama dospem 2>, selaku Dosen Pembimbing II, atas kesabaran, bimbingan, dan masukan yang berharga dalam membimbing penulis sepanjang penulisan penelitian Laporan Skripsi ini.
7. Serta seluruh pihak yang telah membantu penulis yang namanya tidak dapat disebutkan satu per satu.

Penulis juga menyampaikan permohonan maaf yang sebesar-besarnya apabila ada kesalahan dan kekhilafan baik itu perkataan ataupun perbuatan yang dilakukan secara sengaja ataupun tidak disengaja yang kurang berkenan di hati. Dengan segala kerendahan hati, peneliti menyadari bahwa proposal skripsi ini masih jauh dari sempurna karena keterbatasan peneliti. Namun demikian diharapkan skripsi ini dapat bermanfaat sebagaimana mestinya, Aamiin Yaa Robbal’alamin.

|  |
| --- |
| Jakarta, 28 Agustus 2023 |
|  |
| Yahya Pratama |

# **DAFTAR ISI**

[**LEMBAR PERSETUJUAN** 2](#_Toc154603616)

[**KATA PENGANTAR** 3](#_Toc154603617)

[**DAFTAR ISI** 5](#_Toc154603618)

[**BAB I PENDAHULUAN** 8](#_Toc154603619)

[**1.1 Latar Belakang** 8](#_Toc154603620)

[**1.2 Rumusan Masalah** 10](#_Toc154603621)

[**1.3 Batasan Masalah** 10](#_Toc154603622)

[**1.3.1 Metode** 10](#_Toc154603623)

[**1.3.2 Tools** 10](#_Toc154603624)

[**1.3.2 Proses** 11](#_Toc154603625)

[**1.4 Tujuan Penelitian** 11](#_Toc154603626)

[**1.5 Manfaat Penelitian** 11](#_Toc154603627)

[**1.5.1 Bagi Penulis** 11](#_Toc154603628)

[**1.5.2 Bagi Universitas** 11](#_Toc154603629)

[**1.5.3 Bagi Masyarakat** 11](#_Toc154603630)

[**1.6 Metodologi Penelitian** 12](#_Toc154603631)

[**1.6.1 Metode Pengumpulan Data** 12](#_Toc154603632)

[**1.6.2 Metode Pengembangan Sistem** 12](#_Toc154603633)

[**1.7 Sistematika Penulisan** 13](#_Toc154603634)

[**BAB II LANDASAN TEORI** 15](#_Toc154603635)

[**2.1 Aplikasi *Website*** 15](#_Toc154603636)

[**2.2 Arsitektur *Microservices*** 15](#_Toc154603637)

[**2.3 *Application Programming Interface* (API)** 16](#_Toc154603638)

[**2.4 HTML** **Error! Bookmark not defined.**](#_Toc154603639)

[**2.5 CSS** **Error! Bookmark not defined.**](#_Toc154603640)

[**2.6 JavaScript** 17](#_Toc154603641)

[**2.6.1 NodeJS** 17](#_Toc154603642)

[**2.7 Structured Query Language (SQL)** 18](#_Toc154603643)

[**2.7.1 MySQL** 18](#_Toc154603644)

[**2.8 Google Cloud Platform** 18](#_Toc154603645)

[**2.8.1 App Engine** 19](#_Toc154603646)

[**2.8.2 Cloud Storage** 19](#_Toc154603647)

[**2.8.3 Google Cloud SQL** 19](#_Toc154603648)

[**2.8 Tinjauan Pustaka (Literature Review)** 20](#_Toc154603649)

[**BAB III METODOLOGI PENELITIAN** 22](#_Toc154603650)

[**3.1 Metode Pengumpulan data** 22](#_Toc154603651)

[**3.1.1 Studi Pustaka** 22](#_Toc154603652)

[**3.1.2 Observasi** 22](#_Toc154603653)

[**3.1.3 Wawancara** 22](#_Toc154603654)

[**3.2 Metode Pengembangan** 22](#_Toc154603655)

[**3.2.1 Analysis** 23](#_Toc154603656)

[**3.2.2 Design** 23](#_Toc154603657)

[**3.2.3 Implementation** 23](#_Toc154603658)

[**3.2.4 Testing** 23](#_Toc154603659)

[**3.2.5 Maintenance** 24](#_Toc154603660)

[**3.3 Alur Penelitian** 24](#_Toc154603661)

[**DAFTAR PUSTAKA** 25](#_Toc154603662)

# **BAB I PENDAHULUAN**

## **1.1 Latar Belakang**

Bahasa adalah bagian penting dalam interaksi dan komunikasi. Dengan bahasa, individu mampu berinteraksi dan berkomunikasi. Sebagai alat komunikasi, bahasa dapat digunakan setiap orang untuk menyampaikan pendapat, pemikiran, dan perasaan kepada orang lain, baik melalui secara lisan atau tulisan. Ketika tidak menggunakan bahasa, seseorang pasti akan menghadapi kesulitan dalam berinteraksi. Selain itu, Bahasa juga dapat digunakan sebagai tolak ukur individu dalam memahami budaya darimana bahasa itu berasal [1].

Belajar bahasa merupakan hal yang penting terlebih lagi bahasa asing. Dalam era industri 4.0 dimana kita berkomunikasi dan berinterasksi dengan berbagai macam individu dari banyak negara. Belajar bahasa asing seperti bahasa Inggris, Jerman, Prancis, Jepang, China, dan lain-lain menjadi salah aspek yang penting.

Sebagai salah satu contohnya bahasa Jepang. Jepang sendiri membutuhkan pekerja imigram dikarenakan tingkat kelahiran rendah yang mengakibat rendahnya populasi dengan rentang usia muda[2] dan berdasarkan *EF English Proficiency Index* Jepang berada diurutan 87 dari 113 negara[3]. Dengan rendahnya pemahaman bahasa inggris salah satu cara berkomunikasi yang paling efektif adalah dengan menggunakan bahasa Jepang.

Mempelajari bahasa baru bukan suatu hal yang mudah untuk orang dewasa[4] oleh karena itu belajar bahasa baru bisa mulai sejak usia dini. Dalam awal perkembangan anak memperkenalkan bahasa baru dapat mengembangkan *medial temporal lobe* dan meningkatkan *gray matter* yang berfungsi untuk membuat menjadi lebih paham dalam menerima informasi baru dan ingat dalam hal yang dipelajari[5].

Dengan perkembangan teknologi informasi dan komunikasi saat ini, terdapat banyak cara atau metode untuk membantu penggunaan teknologi dalam belajar secara mandiri dan meningkatkan keterampilan mereka.

Sebagai salah satu contohnya aplikasi *website* edukasi bahasa Jepang dapat menjadi salah satu solusi sebagai sarana belajar bahasa Jepang. aplikasi *website* ini dapat memberikan akses kepada seluruh masyarakat untuk belajar bahasa Jepang secara mandiri, kapan saja dan di mana saja selama memiliki akses ke internet. Tapi dengan pesatnya perkembangan teknologi informasi dan komunikasi aplikasi *website* ini perlu memiliki fleksibilitas, dan skalabilitas dalam mengembangkan dan mengelola aplikasi. Masalah ini bisa tanganin dengan *mircoservis*.

*Microservices* merupakan konsep arsitektur yang cocok diimplementasi dalam pengembangan aplikasi *website* dikarenakan microservices membagi sistem besar menjadi layanan-layanan kecil, independen, yang saling bekerja sama dan berkomunikasi menggunakan *Application Programming Interface* (API). Dengan adanya membagi sistem menjadi layanan-layanan kecil dapat memberikan kemudahan dalam melakukan pengembangan, testing, maintenance, dan dapat juga mengimplementasikan teknologi yang berbeda sesuai kebutuhan layanan masing-masing [6].

Aplikasi *website* juga memerlukan aksesibilitas dan ketersedian (*availability*) masalah ini bisa diselesaikan dengan Google Cloud. Google Cloud sendiri memiliki infrastruktur yang luas, dan keamanan yang ketat. Tidak hanya itu pengguna baru Google Cloud juga mendapatkan credit sebesar 300$.

Dengan demikian, penulis betujuan untuk mengembangkan sebuah *website* untuk membantu pengguna dalam belajar bahasa Jepang. penulis memutuskan untuk mengimplementasikan arsitektur *microservices*. Arsitektur *microservices* dipilih karena keunggulannya dalam membuat sistem jadi fleksibel, dan skalabel. Arsitektur ini memungkinkan pengembangan aplikasi dalam bentuk layanan terpisah sehingga dapat lebih mudah dilanjut jika ingin kembangkan pada platform lainnya.

## **1.2 Rumusan Masalah**

Tujuan dari diadakannya penelitian ini adalah untuk implementasikan arsitektur microservice untuk Website Edukasi.

## **1.3 Batasan Masalah**

Berdasarkan latar belakang permasalahan yang telah dipaparkan di atas, perlu adanya batasan masalah agar penelitian yang dilakukan tetap fokus dan dapat menyelesaikan permasalahan yang ada. Maka batasan masalah dalam penelitian ini yaitu:

### **1.3.1 Metode**

1. Metode pengumpulan data yang digunakan adalah observasi, studi pustaka, dan wawancara.
2. Metode pengembangan perangkat lunak menggunakan metode Waterfall

### **1.3.2 Tools**

1. Tools yang digunakan untuk merancang diagram-diagram yaitu menggunakan aplikasi draw.io
2. Tools yang digunakan untuk pengembangan website yaitu menggunakan Visual Studio Code
3. Bahasa pemrograman yang digunakan untuk sisi front-end adalah Html dan CSS.
4. Bahasa pemrograman yang digunakan untuk sisi back-end adalah javascript dengan framework Express.js.
5. Menggunakan Google Cloud Platform (GCP) sebagai penyedia layanan cloud computing.
6. Menggunakan Cloud SQL dari GCP sebagai layanan database relasional.
7. Menggunakan Cloud Storage dari GCP sebagai layanan untuk menyimpan objek.
8. Menggunakan App Engine dari GCP sebagai layanan serverless compute platform untuk deploy aplikasi.
9. Source code aplikasi disimpan dalam remote repository pada platform Github.
10. Desain user interface (UI) aplikasi website menggunakan Figma.

### **1.3.2 Proses**

1. Penelitian ini akan berfokus pada implementasi arsitektur microservice pada aplikasi berbasis website.

## **1.4 Tujuan Penelitian**

Tujuan dari diadakannya penelitian ini adalah untuk pengimplementasikan arsitektur microservice untuk Website Edukasi.

## **1.5 Manfaat Penelitian**

Adapun manfaat yang didapatkan dari penelitian ini adalah sebagai berikut :

### **1.5.1 Bagi Penulis**

Penulis dapat mengimplementasikan ilmu-ilmu yang sudah dipelajari selama perkuliahan ke dalam penelitian ini.

### **1.5.2 Bagi Universitas**

Mengukur tingkat kemampuan dalam menerapkan ilmu akademis maupun non-akademis di lingkungan masyarakat dan Lembaga

### **1.5.3 Bagi Masyarakat**

Aplikasi website edukasi ini menjadi salah satu alternatif media belajar Bahasa jepang untuk anak usia dini

## **1.6 Metodologi Penelitian**

Metode yang digunakan penulis dalam penulisan dan penelitian dibagi menjadi dua, yaitu metode pengumpulan data dan metode pengembangan sistem. Berikut penjelasan kedua metode tersebut:

### **1.6.1 Metode Pengumpulan Data**

Pada penelitian ini penulis menggunakan dua buah metode pengumpulan data untuk mendukung kegiatan penelitian ini yaitu:

1. Studi Pustaka

Studi pustaka merupakan metode pengumpulan data dengan cara mencari referensi dari berbagai sumber seperti buku, buku elektronik, jurnal, paper, dan website yang dapat dipercaya kredibilitasnya.

1. Observasi

Metode observasi digunakan untuk mengamati secara langsung objek yang ingin diamati untuk mendapatkan data primer.

1. Wawancara

Wawancara dilakukan kepada Guru atau *Sensei* yang mengajar bahasa jepang. Wawancara tersebut bertujuan untuk mengetahui lebih dalam mengenai Konten atau Materi apa yang akan dimasukan kedalam website ini.

### **1.6.2 Metode Pengembangan Sistem**

Pada penelitian ini digunakan metode pengembangan sistem Waterfall, metode Waterfall adalah salah satu metode pengembangan sistem sekuensial yang terdiri dari beberapa fase atau tahapan yang berurutan [7]. Metode ini memiliki lima fase utama, yaitu:

1. Fase Analysis: Pada fase ini akan dilakukan pengumpulan data dan pengidentifikasian masalah berdasarkan kebutuhan pengguna dan data yang sudah dikumpulkan.
2. Fase Design: Fase ini akan dilakukan pembuatan arsitektur sistem dan tampilanya sesuai dengan kebutuhan spesifikasi sistem. Fase ini bertujuan sebagai acuan untuk fase selanjutnya.
3. Fase Implementation: Fase ini akan dilakukan implementasi dari desain perangkat lunak difase yang sebulumnya menjadi kode program dan juga akan ada pengujian untuk setiap komponen perangkat lunak untuk memastikan perangkat lunak yang dihasilkan berkualitas.
4. Fase Testing: Fase ini akan dilakukan pengujian sistem secara keseluruhan untuk memastikan sistem berkerja dengan baik dan aman.
5. Fase Maintenance: Fase ini akan dilakukan perawatan sistem selama masa operasi untuk mengatasi masalah, kesalahan, dan perubahan kebutuhan yang perlu.

## **1.7 Sistematika Penulisan**

Sistematika penulisan yang akan dilakukan dalam penelitian ini terdiri dari 6 bab, yaitu:

**BAB 1 PENDAHULUAN**

Dalam bab ini akan dibahas mengenai latar belakang, Batasan masalah, tujuan, dan manfaat serta sistematika penulisan.

**BAB II TINJAUAN PUSTAKA DAN LANDASAR TEORI**

Dalam bab ini akan dibahas mengenai berbagai teori yang mendasari analisis permasalahan dan berhubungan dengan topik yang dibahas.

**BAB III METODOLOGI PENELITIAN**

Bab ini membahas mengenai metode penelitian yang akan digunakan dalam merancang dan membangun prototipe sistem.

**BAB IV ANALISIS, DESAIN, IMPLEMENTASI, DAN PENGUJIAN SISTEM**

Pada bab ini membahas mengenai hasil dari analisis, perancangan, implementasi sesuai dengan metode yang dilakukan pada sistem yang dibuat serta hasil dari pengujian.

**BAB V HASIL DAN PEMBAHASAN**

Bab ini berisi hasil dari pembahasan yang didapat dari penelitian.

**BAB VI KESIMPULAN DAN SARAN**

Bab ini berisi tentang kesimpulan dari penelitian yang telah dilakukan dan saran yang diusulkan untuk pengembangan lebih lanjut agar tercapai hasil yang lebih baik.

# **BAB II LANDASAN TEORI**

## **2.1 Aplikasi *Website***

Aplikasi web adalah perangkat lunak yang berjalan di browser web Anda. Bisnis harus bertukar informasi dan memberikan layanan dari jarak jauh. Bisnis menggunakan aplikasi web untuk terhubung dengan pelanggan secara nyaman dan aman. Fitur situs web yang paling umum, seperti keranjang belanja, pencarian dan pemfilteran produk, pesan instan, serta umpan berita media sosial adalah aplikasi web dalam desainnya. Dengan fitur-fitur tersebut, Anda dapat mengakses fungsionalitas yang kompleks tanpa menginstal atau mengonfigurasi perangkat lunak [8].

## **2.2 Arsitektur *Microservices***

Microservices adalah arsitektur perangkat lunak yang bersifat independen dan tidak memiliki tergantungan terhadap servis yang lain. *Microservices* itu sendiri memiliki arsitektur berorientasi layanan atau *Services Oriented Architechture* (SOA) [6].

Berdasarkan buku yang ditulis oleh Newman [6] Mircroservis berberapa keuntungan dari arsitektur perangkat lunak yang lain, yaitu sebagai berikut :

1. Technology Heterogeneity

Dalam pengembangan sistem *microservices*, pengembang sistem dapat memilihi teknologi yang tepat sesuai dengan kebutuhan layanan atau *services*.

1. Resilience

*Mircoservices* adalah sistem yang terdiri dari beberapa layanan. Oleh karena itu, jika satu dari layanan dalam sistem *mircoservices* ini mengalami kegagalan, sistem akan terus berjalan dan mengisolasi layanan yang sedang mengalami kegagalan tersebut sehigga sistem tetap berejalan lancer.

1. Scaling

Dalam sistem *microservices,* pengembang sistem hanya perlu mengukur CPU dan Ram yang diperlukan oleh layanan, tanpa harus mengukur seluruh layanan seperti arsitektur monolitik.

1. Ease of Deployment

Dengan *microservices,* Pengembang sistem dapat melakukan perubahan baris kode dalam salah satu layanan tanpa melibatkan layanan lain yang tidak memerlukan perubahan.

1. Organizational Alignment

Dalam sistem *microservices*, pembagian tugas menjadi lebih mudah karena satu pengembang hanya perlu focus pada setiap layanan sehingga hasil kerja menjadi lebih produktif.

1. Composability

Dengan *microservices,* komposisi layanan dapat dilakukan dengan mudah, karena proses pemilihan dan penyusunan setiap layanan hanya perlu untuk memperhatikan interface antar layanan.

1. Optimizing for Replaceability

Dengan layanan *microservices* yang berukuran kecil, biaya untuk mengganti layanan dengan implementasi yang lebih baik, atau penghapusan layanan, jauh lebih mudah untuk dikelola. Hambatan untuk melakukan penulisan ulang atau penghapusan layanan seluruhnya juga menjadi sangat rendah.

## **2.3 *Application Programming Interface* (API)**

*Application Programming Interface* atau bisa disebut sebagai API ini adalah layanan atau data yang disediakan oleh suatu aplikasi perangkat lunak melalui serangkaian sumber daya yang telah ditentukan sebelumnya, seperti metode, objek, atau *URI* [9]. Dengan menggunakan sumber daya ini, aplikasi lain dapat mengakses data atau layanan tanpa harus membuatnya sendiri. API menjadi pusat dalam banyak arsitektur perangkat lunak modern, karena mereka menyediakan abstraksi tingkat tinggi yang memudahkan tugas pengembang, mendukung desain aplikasi perangkat lunak yang terdistribusi dan modular, serta memfasilitasi penggunaan ulang kode [10].

### **2.3.1** ***Representational State Transfer* (REST)**

*Representational State Transfer* atau bisa disebut sebagai REST merupakan arsitektur untuk aplikasi layanan web yang menerapkan konsep peralihan antara *states*. *states* di sini dapat dijelaskan sebagai browser yang meminta halaman web, di sisi server akan mengirimkan keadaan terkini dari halaman web ke browser. REST API memungkinkan berbagai sistem berkomunikasi dan mengirim/menerima data dengan cara yang sangat sederhana[11].

## **2.4 JavaScript**

JavaScript adalah bahasa pemrograman berorientasi objek yang menggunakan pewarisan objek prototipe[12]. JavaScript juga merupakan salah satu bahasa pemrograman yang popular dan digunakan secara luas tidak hanya untuk aplikasi web dalam sisi klien, tetapi juga untuk aplikasi sisi server yang berjalan menggunakan Nodejs[13].

### **2.4.1 NodeJS**

Node.js adalah lingkungan runtime JavaScript yang umumnya digunakan untuk membangun platform *server-side*. Node.js sendiri dibangun diatas V8 milik Chrome.

Node.js juga sangat populer di kalangan komunitas *open-source* dan perusahaan-perusahaan besar. Banyak dari perusahaan-perusahaan besar seperti Microsoft, IBM, PayPal, dan lainnya menggunakan Node.js dalam produk mereka. Salah satu alasan popularitasnya terletak pada pilihan arsitektur Node.js. Node.js menggunakan arsitektur *non-blocking event-based*. arsitektur ini memberikan kemampuan kepada pengembang untuk dengan mudah mengembangkan skala aplikasi Node.js[14].

## **2.5 Structured Query Language (SQL)**

*Structured Query Language* (SQL) adalah mekanisme yang diterima secara universal untuk mengakses dan memanipulasi data yang tersimpan dalam sistem manajemen basis data relasional. SQL adalah bahasa berbasis teks yang memungkinkan pengguna untuk mendeskripsikan struktur hirarkis database relasional dalam sebuah kueri, sehingga memungkinkan untuk membuat pertanyaan yang rumit dan kuat secara langsung[15].

### **2.5.1 MySQL**

MySQL adalah perangkat lunak database *open-source* yang paling banyak digunakan di dunia, dengan lebih dari 100 juta pengguna di seluruh dunia. MySQL telah menjadi *database* pilihan bagi banyak pengembang perangkat lunak dan aplikasi pada platform online dan desktop karena keandalan, kecepatan, dan kemudahan penggunaannya. Individu dan bisnis kecil bukanlah satu-satunya yang menggunakan MySQL banyak perusahaan seperti Yahoo!, Alcatel-Lucent, Google, Nokia, Youtube, WordPress, dan Facebook menggunakan MySQL[16].

## **2.6 Google Cloud Platform (GCP)**

Google Cloud Platform (GCP) saat ini merupakan salah satu yang paling penting dan berkembang di pasar cloud. GCP menyediakan beberapa produk bagi para pengembang untuk membangun berbagai program, mulai dari situs web sederhana hingga aplikasi kompleks yang didistribusikan ke seluruh dunia. GCP menawarkan layanan hosting pada infrastruktur pendukung yang sama dengan yang digunakan Google secara internal untuk produk end-user seperti Google Search dan YouTube. Reliabilitas yang luar biasa ini membuat GCP diadopsi oleh organisasi-organisasi terkemuka seperti Airbus, Coca-Cola, HTC, Spotify, dan lain-lain. Selain itu, jumlah mitra GCP juga meningkat secara substansial, terutama Equinix, Intel, dan Red Hat[17].

### **2.6.1 App Engine**

App Engine adalah salah satu layanan Google Cloud Platform yang bertujuan untuk mengembangkan dan menghosting aplikasi *website* di pusat data yang dikelola Google. App Engine menawarkan *automatic scaling* untuk aplikasi *website* seiring dengan meningkatnya jumlah *requests* untuk sebuah aplikasi, App Engine secara otomatis mengalokasikan lebih banyak sumber daya untuk aplikasi web untuk menangani *requests* tambahan.

App Engine mendukung berbagai bahasa pemrograman seperti Go, PHP, Java, Python, Node.js, .NET, dan Ruby, meskipun begitu App Engine juga dapat mendukung bahasa lain melalui "custom runtime".

### **2.6.2 Cloud Storage**

Cloud Storage adalah layanan untuk menyimpan objek yang ditawarkan oleh Google Cloud. Objek adalah bagian data yang tidak dapat diubah yang terdiri atas file dalam format mp3, mp4, mkv, png, jpg dan lain-lain. Cloud Storage ini menyimpan objek dalam wadah yang disebut bucket. Bucket juga dapat berisi folder terkelola.

### **2.6.3 Cloud SQL**

Cloud SQL adalah layanan basis data relasional yang dikelola sepenuhnya untuk MySQL, PostgreSQL, dan SQL Server.

## **2.7 Tinjauan Pustaka (Literature Review)**

Peneliti juga melakukan studi literatur serupa sebagai referensi penelitian yang akan dilakukan. Berikut penelitianpenelitian yang penulis gunakan sebagai bahan referensi:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| No | Judul | Penulis | Desain Arsitektur | Protokol Komunikasi | Hasil |
| 1 |  |  |  |  |  |
| 2 |  |  |  |  |  |
| 3 |  |  |  |  |  |
| 4 |  |  |  |  |  |
| 5 |  |  |  |  |  |
| 6 | IMPLEMENTASI ARSITEKTUR MICROSERVICES DALAM PERANCANGAN APLIKASI WEBSITE EDUKASI BAHASA JEPANG UNTUK ANAK DIUSIA DINI | Yahya Pratama | *Microservices* | RESTful API |  |

# **BAB II****I METODOLOGI PENELITIAN**

## **3.1 Metode Pengumpulan Data**

Penulis melakukan pengumpulan data sebagai cara untuk mengumpulkan data yang berkaitan dengan penelitian seperti latar belakang penelitian, landasan teori, metodologi penelitian, dan pembahasan. Pada penelitian ini, penulis menggunakan studi pustaka, observasi dan wawancara sebagai metode pengumpulan data.

### **3.1.1 Studi Pustaka**

Dalam studi literatur ini, penulis membaca dan mempelajari berbagai materi yang berkaitan dengan penelitian dari sumber referensi terpercaya, seperti buku, jurnal, e-book, e-learning, dan situs-situs internet. Sumber dari referensi tersebut dapat dilihat pada halaman Daftar Pustaka.

### **3.1.2 Observasi**

penulis akan melakukan observasi aplikasi website serupa yang memiliki konsep yang sama, yaitu sebuah aplikasi website yang menggunakan arsitektur microservices.

### **3.1.3 Wawancara**

Wawancara dilakukan menggunakan aplikasi ZOOM Meeting kepada guru bahasa jepang yang berpengalaman dalam mengajar anak usai dini. Wawancara ini dilakukan penulis untuk mengetahui apa topik belajar yang cocok untuk anak usai dini.

## **3.2 Metode Pengembangan**

Penulis akan menggunakan metode pengembangan metode Waterfall dalam pengembangan aplikasi website. Metode ini terdiri dari lima tahap, yaitu *Analysis*, *Design*, *Implementation*, *Testing*, dan *Maintenance*. Tahap-tahap ini dilakukan secara berurutan dan setiap tahap harus selesai sebelum tahap berikutnya dimulai.

### **3.2.1 *Analysis***

Dalam tahap ini, penulis memahami kebutuhan sistem dan menentukan fitur-fitur yang harus ada pada aplikasi website yang akan dikembangkan. Penulis melakukan analisis dari literatur serupa untuk memperoleh pemahaman yang lebih baik tentang topik penelitian. Dalam melakukan analisis literatur serupa, penulis mempelajari kelebihan, kekurangan, dan fitur-fitur yang disediakan dari hasil penelitian literatur serupa. penulis juga melakukan observasi aplikasi-aplikasi yang memiliki tujuan serupa dengan penelitian penulis. Observasi ini dilakukan untuk memperoleh gambaran yang lebih jelas tentang aplikasi-aplikasi tersebut dan bagaimana aplikasi-aplikasi tersebut dapat membantu dalam pengembangan penelitian penulis. penulis juga melakukan wawancara kepada guru bahasa jepang yang berpengalaman dalam mengajar anak usai dini. Wawancara ini bertujuan agar penulis mengetahui apa topik belajar yang cocok untuk anak usai dini

### **3.2.2 *Design***

Dalam tahap ini, Penulis akan merancang arsitektur aplikasi yang akan dikembangkan. Arsitektur aplikasi harus memenuhi spesifikasi kebutuhan yang telah ditetapkan pada tahap analisis kebutuhan.

### **3.2.3 *Implementation***

Dalam Tahap ini akan dilakukan untuk mengimplementasikan rancangan aplikasi yang sebelumnya telah disusun dalam tahap perancangan.

### **3.2.4 *Testing***

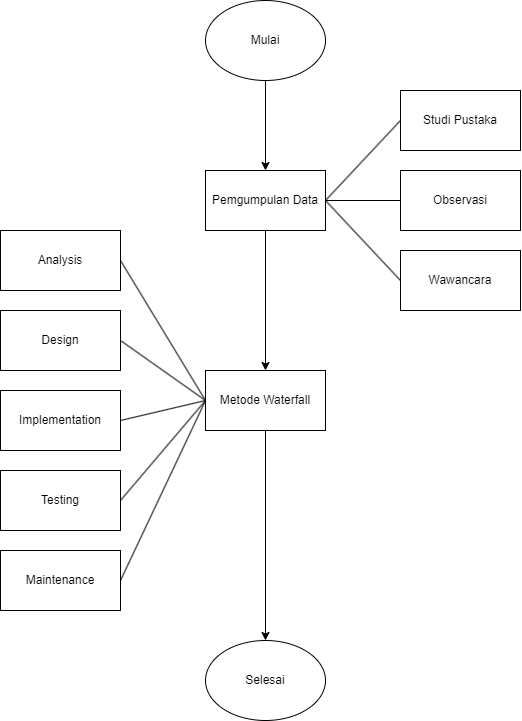
Dalam tahap ini, semua komponen yang telah diimplementasikan akan dilakukan proses pengujian dengan tujuan untuk membuktikan apakah keseluruhan sistem beroperasi secara baik dan sesuai dengan desain yang telah ditetapkan.

### **3.2.5** ***Maintenance***

Setelah *deployment* berhasil pengembang akan melakukan pemeliharaan aplikasi. Dalam tahap ini, pemeliharaan akan dilakukan dengan menghilangkan *bug*, menambahkan fitur, meningkatkan performa dan keamanan, meningkatkan pengalaman pengguna, serta penyesuaian dengan perangkat dan teknologi terbaru.

## **3.3 Alur Penelitian**

Alur penelitian berguna untuk menguraikan proses-proses pada penelitian ini. Alur penelitian dapat dilihat dalam gambar berikut:

******

# **DAFTAR PUSTAKA**

1. Malik, Agung Rinaldy, and Muhammad Nur Ashar Asnur. "Using Social Media As A Learning Media Of Foreign Language Students In Higher Education." Online Submission 18.2 (2019).
2. Iijima S, Yokoyama K. [Socioeconomic Factors and Policies Regarding Declining Birth Rates in Japan]. Nihon Eiseigaku Zasshi. 2018;73(3):305-312. Japanese. doi: 10.1265/jjh.73.305. PMID: 30270298.
3. Education First. (2023). EF English Proficiency Index (2023). https://www.ef.com/assetscdn/WIBIwq6RdJvcD9bc8RMd/cefcom-epi-site/reports/2023/ef-epi-2023-english.pdf.
4. Gullberg, Marianne, et al. "Adult language learning after minimal exposure to an unknown natural language." Language Learning 60 (2010): 5-24.
5. Bektur, Gulnaz, and M. Mustafina. "The importance of effective foreign language teaching methods." Scientific Collection «InterConf» 150 (2023): 87-92.
6. Newman, S. (2015). Building Microservices (First Edit; M. L. and B. MacDonald, Ed.). Sebastopol: O’Reilly Media, Inc.
7. Aroral, Harkirat Kaur. "Waterfall Process Operations in the Fast-paced World: Project Management Exploratory Analysis." International Journal of Applied Business and Management Studies 6.1 (2021): 91-99.
8. Amazon Web Services Inc (2023). Apa itu Aplikasi Web? - Penjelasan tentang Aplikasi Web - AWS (2023). <https://aws.amazon.com/id/what-is/web-application/>.
9. Jeong, Sae Young, et al. "Improving documentation for eSOA APIs through user studies." End-User Development: 2nd International Symposium, IS-EUD 2009, Siegen, Germany, March 2-4, 2009. Proceedings 2. Springer Berlin Heidelberg, 2009.
10. Robillard, Martin P. "What makes APIs hard to learn? Answers from developers." IEEE software 26.6 (2009): 27-34.
11. Ahmad, Imam, et al. "Implementation of RESTful API Web Services Architecture in Takeaway Application Development." 2021 1st International Conference on Electronic and Electrical Engineering and Intelligent System (ICE3IS). IEEE, 2021.
12. Johannes, David, Foutse Khomh, and Giuliano Antoniol. "A large-scale empirical study of code smells in JavaScript projects." Software Quality Journal 27 (2019): 1271-1314.
13. Malik, Rabee Sohail, Jibesh Patra, and Michael Pradel. "NL2Type: inferring JavaScript function types from natural language information." 2019 IEEE/ACM 41st International Conference on Software Engineering (ICSE). IEEE, 2019.
14. Koishybayev, Igibek, and Alexandros Kapravelos. "Mininode: Reducing the attack surface of node. js applications." 23rd International Symposium on Research in Attacks, Intrusions and Defenses (RAID 2020). 2020.
15. Jamison, D. Curtis. "Structured query language (SQL) fundamentals." Current protocols in bioinformatics 1 (2003): 9-2.
16. Rawat, Bhupest, and Suryari Purnama. "MySQL Database Management System (DBMS) On FTP Site LAPAN Bandung." International Journal of Cyber and IT Service Management 1.2 (2021): 173-179.
17. Challita, Stéphanie, et al. "A precise model for google cloud platform." 2018 IEEE international conference on cloud engineering (IC2E). IEEE, 2018.