**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).

|  |  |  |
| --- | --- | --- |
| Dosen Pembimbing 1 |  | Calon Dosen Pembimbing 2 |
| <tanda tangan> |  | <tanda tangan> |
| Herlino Nanang, M.T |  | Victor Amrizal, M.Kom |
| 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. Herlino Nanang, M.T, selaku Dosen Pembimbing I, dan Victor Amrizal, M.Kom, 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 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** ii](#_Toc165974871)

[**KATA PENGANTAR** iii](#_Toc165974872)

[**DAFTAR GAMBAR** viii](#_Toc165974873)

[**DAFTAR TABEL** ix](#_Toc165974874)

[**BAB I PENDAHULUAN** 1](#_Toc165974875)

[**1.1** **Latar Belakang** 1](#_Toc165974876)

[**1.2** **Rumusan Masalah** 4](#_Toc165974877)

[**1.3** **Batasan Masalah** 4](#_Toc165974878)

[**1.3.1** **Metode** 4](#_Toc165974879)

[**1.3.2** **Tools** 4](#_Toc165974880)

[**1.3.3** **Proses** 5](#_Toc165974881)

[**1.4** **Tujuan Penelitian** 5](#_Toc165974882)

[**1.5** **Manfaat Penelitian** 5](#_Toc165974883)

[**1.5.1** **Bagi Penulis** 5](#_Toc165974884)

[**1.5.2** **Bagi Universitas** 5](#_Toc165974885)

[**1.5.3** **Bagi Masyarakat (spesifik)** 6](#_Toc165974886)

[**1.6** **Metodologi Penelitian** 6](#_Toc165974887)

[**1.6.1** **Metode Pengumpulan Data** 6](#_Toc165974888)

[**1.6.2** **Metode Pengembangan Sistem** 6](#_Toc165974889)

[**1.7** **Sistematika Penulisan** 7](#_Toc165974890)

[**BAB II LANDASAN TEORI** 9](#_Toc165974891)

[**2.1** **Bahasa Jepang** 9](#_Toc165974892)

[**2.2** **Client** 9](#_Toc165974893)

[**2.3** **Server** 9](#_Toc165974894)

[**2.4** **Aplikasi *Website*** 10](#_Toc165974895)

[***2.5*** **Arsitektur *Microservices*** 10](#_Toc165974896)

[**2.6** ***Application Programming Interface* (API)** 12](#_Toc165974897)

[**2.6.1** ***Representational State Transfer* (REST)** 13](#_Toc165974898)

[**2.7** **JavaScript** 13](#_Toc165974899)

[**2.7.1** **NodeJS** 14](#_Toc165974900)

[**2.8** **Structured Query Language (SQL)** 15](#_Toc165974901)

[**2.8.1** **MySQL** 15](#_Toc165974902)

[**2.9** **Google Cloud Platform (GCP)** 16](#_Toc165974903)

[**2.9.1** **App Engine** 16](#_Toc165974904)

[**2.9.2** **Cloud Storage** 17](#_Toc165974905)

[**2.9.3** **Cloud SQL** 17](#_Toc165974906)

[**2.10** **Tinjauan Pustaka (Literature Review)** 18](#_Toc165974907)

[**BAB III METODOLOGI PENELITIAN** 24](#_Toc165974908)

[**3.1** **Objek Penelitian** 24](#_Toc165974909)

[**3.2** **Metode Pengumpulan Data** 24](#_Toc165974910)

[**3.2.1** **Studi Pustaka** 24](#_Toc165974911)

[**3.2.2** **Observasi** 24](#_Toc165974912)

[**3.2.3** **Wawancara** 24](#_Toc165974913)

[**3.3** **Metode Pengembangan** 25](#_Toc165974914)

[**3.3.1** ***Analysis*** 25](#_Toc165974915)

[**3.3.2** ***Design*** 25](#_Toc165974916)

[**3.3.3** ***Implementation*** 26](#_Toc165974917)

[**3.3.4** ***Testing*** 26](#_Toc165974918)

[**3.3.5** ***Maintenance*** 26](#_Toc165974919)

[**3.4** **Alur Penelitian** 26](#_Toc165974920)

[**BAB IV ANALISIS, DESAIN, IMPLEMENTASI, DAN PENGUJIAN SISTEM** 28](#_Toc165974921)

[**DAFTAR PUSTAKA** 37](#_Toc165974922)

# **DAFTAR GAMBAR**

[Gambar 2.1 Arsitektur Mircoservice 11](#_Toc165274825)

[Gambar 2.2 Cara Kerja API 13](#_Toc165274826)

[Gambar 2.3 Logo JavaScript 14](#_Toc165274827)

[Gambar 2.4 Logo NodeJS 14](#_Toc165274828)

[Gambar 2.5 Logo MySQL 15](#_Toc165274829)

[Gambar 2.6 Logo Google Cloud 16](#_Toc165274830)

[Gambar 4.1 *Use Case Diagram* 31](#_Toc165274831)

# **DAFTAR TABEL**

[*Tabel 2.1 Tinjauan Pustaka* 20](#_Toc156248561)

# **BAB I PENDAHULUAN**

## **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 (Rinaldy Malik & Nur Ashar, 2019).

Belajar bahasa merupakan hal yang penting terlebih lagi bahasa asing. Dalam era global 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 (Subekti, 2019).

Sebagai salah satu contohnya bahasa Jepang. Jepang sendiri membutuhkan pekerja imigram dikarenakan tingkat kelahiran rendah yang mengakibat rendahnya populasi dengan rentang usia muda (Jack, 2016) dan berdasarkan Education First (EF*) English Proficiency Index* Jepang berada diurutan 87 dari 113 negara (EF, 2023). 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 (Gullberg et al., 2010) oleh karena itu belajar bahasa asing 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 (Gulnaz Beybitzhanovna, 2023)

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

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 *mircoservices*.

*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 (Newman, 2015).

Implementasi arsitektur mikroservis tidak cocok dilakukan secara on-premise karena memerlukan infrastruktur yang kompleks dan biaya operasional yang tinggi. Pembangunan dan pemeliharaan data center sendiri dapat menjadi beban yang besar, terutama untuk organisasi yang tidak memiliki sumber daya dan keahlian yang cukup dalam pengelolaan infrastruktur IT. Selain itu, skalabilitas dan elastisitas yang diperlukan untuk arsitektur mikroservis sulit dicapai secara on-premise tanpa investasi yang signifikan dalam perangkat keras dan perangkat lunak (Kratzke, 2018). Oleh karena itu, implementasi aplikasi *website* akan dilakukan kedalam lingkungan cloud dengan layanan yang sediakan dengan Google Cloud.

Implementasi arsitektur *microservices* dalam lingkungan cloud cocok untuk karena skalabilitas, fleksibilitas, elastisitas, ketersediaan(*availability*), dan keamanan yang diperlukan oleh aplikasi website. Dengan menggunakan layanan Google Cloud, pengembang dapat dengan mudah menyebarkan, mengelola, dan memantau layanan-layanan *microservices* secara efisien, sambil memanfaatkan fitur otomatisasi dan keamanan yang disediakan oleh Google Cloud. Hal ini memastikan bahwa aplikasi dapat berjalan dengan handal dan aman, serta dapat menanggapi fluktuasi lalu lintas dengan cepat tanpa memerlukan investasi besar dalam infrastruktur fisik(Astya et al., 2017).

Dalam penelitian ini penulis akan menggunakan metode pengembangan perangkat lunak waterfall. Metode sangat sesuai dengan penelitian kerena kebutuhan yang telah ditentukan secara jelas dan sedikit perubahan (Aroral, 2021). Dalam konteks perancangan aplikasi *website* edukasi, di mana kurikulum dan materi pembelajaran sudah terdefinisi dengan baik sebelumnya, serta sedikit perubahan yang diharapkan dalam proses pengembangan, Dengan pendekatan waterfall, tahapan pengembangan seperti analysis, design, implementation, testing dan maintenance dapat dilakukan secara berurutan dan terdokumentasi dengan baik, yang sesuai dengan sifat proyek yang terstruktur dan terperinci. Ini memungkinkan penulis untuk menyusun skripsi dengan rapi dan terperinci, menyoroti setiap tahapan secara sistematis dalam proses pengembangan aplikasi *website*.

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.

## **Rumusan Masalah**

Berdasarkan latar belakang yang telah dipaparkan sebelumnya, maka rumusan masalah adalah bagaimana penerapan adalah implementasikan arsitektur *microservices* untuk *aplikasi* Website Edukasi dalam lingkungan *cloud*.

## **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:

### **Metode**

1. Metode pengumpulan data yang digunakan adalah observasi, studi pustaka, dan wawancara.
2. Metode pengembangan perangkat lunak menggunakan metode Waterfall.
3. Testing fungsional aplikasi website menggunakan Black Box

### **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.

### **Proses**

1. Penelitian ini akan berfokus pada implementasi arsitektur *microservices* pada aplikasi berbasis website.
2. Katakana dan Hiragana akan menjadi fokus isian konten dalam aplikasi website
3. Website Edukasi berfokus dalam penulisan Bahasa jepang

## **Tujuan Penelitian**

Tujuan dari diadakannya penelitian ini adalah untuk pengimplementasikan arsitektur *microservices* untuk Website Edukasi dalam lingkungan *cloud*.

## **Manfaat Penelitian**

Adapun manfaat yang didapatkan dari penelitian ini adalah sebagai berikut :

### **Bagi Penulis**

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

### **Bagi Universitas**

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

### **Bagi Masyarakat (spesifik)**

*Aplikasi* website edukasi ini menjadi salah satu alternatif media belajar Bahasa jepang untuk anak usia dinil.

## **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:

### **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.

### **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 (Aroral, 2021). 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.

## **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**

## **Bahasa Jepang**

Bahasa Jepang atau Nihongo (日本語) adalah bahasa resmi yang digunakan oleh masyarakat Jepang. Bahasa dan huruf yang digunakan memiliki keterikatan dengan bangsa-bangsa disekitarnya. Bahasa Jepang ini memiliki tiga macam huruf atau kana yaitu Hiragana, Katakana, serta Kanji. Masing-masing dari huruf tersebut memiliki fungsi tersendiri. Hiragana digunakan pada bahasa Jepang asli dan tata bahasa, sedangkan Katakana digunakan pada bahasa serapan dan terutama penulisan nama. Kanji atau Hanzi sendiri merupakan serapan dari bangsa Cina ketika dibawah kekuasaan Dinasti Han yang umumnya digunakan pada kata benda (Sunarti et al., 2016).

## **Client**

Sebagai komponen dalam arsitektur jaringan, seorang client adalah entitas yang meminta atau mengakses layanan, informasi, atau sumber daya dari server. Sebuah client dapat berupa perangkat keras seperti komputer atau smartphone, atau perangkat lunak seperti browser web, aplikasi email, atau aplikasi permainan. Fungsi utamanya adalah untuk mengirim permintaan kepada server dan menerima respons atau layanan yang diberikan oleh server sesuai dengan permintaannya. Dengan demikian, client berperan sebagai pengguna akhir yang memanfaatkan layanan yang disediakan oleh server dalam sebuah jaringan computer (Harsono, 2022).

## **Server**

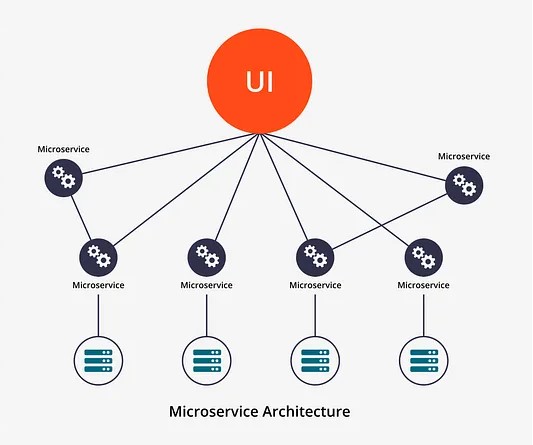
Server, dalam bahasa Indonesia bisa diartikan dengan peladen yaitu suatu sistem komputer yang memiliki layanan khusus berupa penyimpanan data. Server memiliki peran penting dalam menyediakan layanan akses agar lebih cepat untuk mengirim atau menerima data maupun informasi yang tersedia pada server. Dan server ini memiliki bentuk fisik. Server dalam bentuk fisiknya, berwujud jaringan komputer dan memiliki ukuran yang sangat besar dengan beberapa komponen pendukung prosesor dan RAM yang berkapasitas besar (Harsono, 2022).

## **Aplikasi *Website***

Aplikasi web adalah perangkat lunak yang berjalan di browser web. 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 (Sturm et al., 2017).

## **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) (Newman, 2015).



Gambar 2.1 Arsitektur Mircoservice

Berdasarkan buku yang ditulis oleh Newman (Newman, 2015) 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

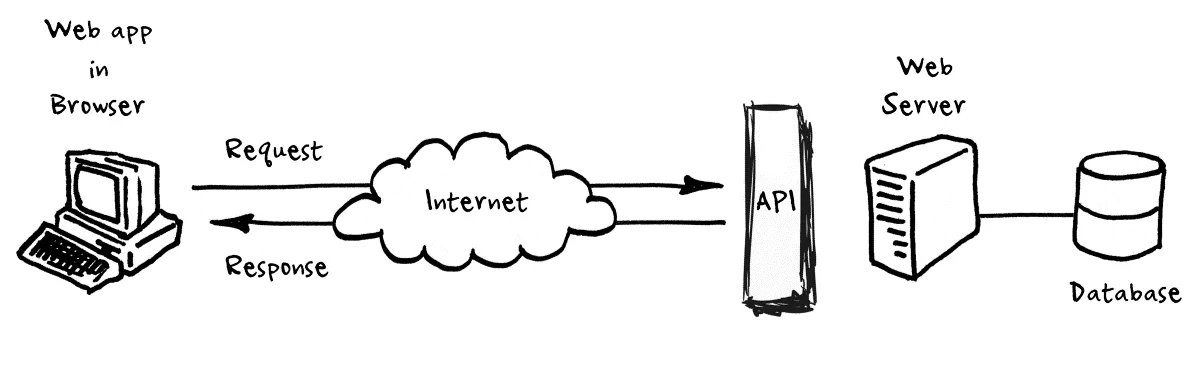
Dengan *microservices,* komposisi layanan dapat dilakukan dengan mudah, karena proses pemilihan dan penyusunan setiap layanan hanya perlu untuk memperhatikan interface antar layanan.

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.

## ***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* (Jeong et al., 2009) 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 (Robillard, 2009)



Gambar 2.2 Cara Kerja API

### ***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 (Ahmad et al., 2021).

## **JavaScript**

JavaScript adalah bahasa pemrograman berorientasi objek yang menggunakan pewarisan objek prototipe (Johannes et al., 2019). 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 (Sohail Malik et al., 2019).



Gambar 2.3 Logo JavaScript

### **NodeJS**

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



Gambar 2.4 Logo NodeJS

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 (Koishybayev & Kapravelos, 2020)

## **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 (Jamison, 2003)

### **MySQL**

MySQL adalah perangkat lunak database *open-source* yang paling banyak digunakan di dunia, dengan lebih dari 100 juta pengguna di seluruh dunia.



Gambar 2.5 Logo MySQL

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 (Rawat & Purnama, 2021).

## **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.



Gambar 2. Logo Google Cloud

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 (Challita et al., 2018).

### **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".

### **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.

### **Cloud SQL**

Cloud SQL adalah layanan database SQL yang ditawarkan oleh Google Cloud dan dikelola sepenuhnya oleh google. Cloud SQL juga kompatibel dengan layanan Google Cloud seperti Compute Engine, App Engine, Cloud Run, dan lain-lain. Selain itu Cloud SQL juga mendukung sistem manajemen basis data relasional seperti MySQL, PostgreSQL, dan SQL Server.

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

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

*Tabel 2.1 Tinjauan Pustaka*

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| No | Judul | Penulis | Desain Arsitektur | Protokol Komunikasi | Server | Hasil |
| 1 | IMPLEMENTASI ARSITEKTUR PEMROGRAMAN MICROSERVICES UNTUK MIGRASI SISTEM INFORMASI PT. DWIDAYA WORLD WIDE (DWIDAYA TOUR AND TRAVEL) DENGAN METODOLOGI SDLC RAPID APPLICATION DEVELOPMENT | Wisnu Sukma Maulana (2022) | Microservices | RESTful API | On Premise | Penelitian ini melibatkan implementasi arsitektur microservices dari aplikasi yang terpisah dengan teknologi berbeda baik dari sisi level front-end maupun back-end ke dalam suatu sistem baru secara terpusat. |
| 2 | IMPLEMENTASI ARSITEKTUR MICROSERVICES PADA RANCANG BANGUN APLIKASI MARKETPLACE BERBASIS WEB | Alessandro Sinambela, Ernawati, Funny Farady Coastera (2021) | Microservices | RESTful API | On Premise | Penelitian ini menghasilkan aplikasi marketplace menggunakan arsitektur microservices berbasis web, arsitektur microservices terdiri lima servis meliputi api-gateway, produk servis, keranjang servis, order servis dan pengguna servis. Aplikasi berhasil lulus dengan hasil persentase pengujian black box 100% dari 25 aktivitas berhasil dan hasil persentase pengujian endpoint api 100% dari 29 aktivitas berhasil. Dan tidak hanya itu peniliti juga melakukan uji kelayakan menggunakan kuisioner kepada pedagang Pasar Tradisional Bengkulu (PTM) didapat nilai varibel tampilan 4,54 dengan kategori sangat baik, variabel kemudahan pengguna 4,63 dengan kategori sangat baik, dan variabel kinerja sistem 4,22 dengan kategori baik. |
| 3 | IMPLEMENTASI ARSITEKTUR MICROSERVICE PADA BACK ENDSISTEM INFORMASI ATLANTAS BERBASIS WEBSITE | Calvin Seviro Bima Sakti, Indra Hermawan (2020) | Microservices | RESTful API | On Premise | Penelitian ini peneliti berhasil membuat sistem back end pada sistem informasi atlantas berhasil dibuat dengan konsep arsitektur microservices yang memberikan informasi kasus pelanggaran dan kebutuhan setiap unit akan saling terintegrasi. Penelitian ini tergolong memuaskan karena dalam pengujian blackbox berhasil mendapatkan nilai 100% yang berarti bahwa sistem backend menggunakan arsitektur microservices sudah berjalan sesuai fungsionalitasnya. |
| 4 | IMPLEMENTASI ARSITEKTUR MICROSERVICE PADA PEMBUATAN SURAT UNIT KEGIATAN MAHASISWA INFORMATIKA DAN KOMPUTER MENGGUNAKAN NODE.JS | Y.Yohakim Marwanta, Badiyanto (2019) | Microservices | RESTful API | On Premise | Penelitian ini menghasilkan sistem aplikasi Surat Unit Kegiatan menggunakan Nodejs sebagai bahasa pemrograman. penerapan arsitektur microservice disistem aplikasi memberikan kemudaan pada saat pengembangan aplikasi kedepannya jika diperlukan penambahan fitur yang mampu meningkatkan performa aplikasi. Hal ini terjadi karena pengembang hanya perlu berfokus pada satu struktur sistem yang kecil tanpa harus memperhatikan keterkaitannya dengan fitur lainnya |
| 5 | ANALISIS PADA ARSITEKTUR MICROSERVICE UNTUK LAYANAN BISNIS TOKO ONLINE | Liqoo Mumbahiz Alchuluq, Fahrul Nurzaman (2020) | Microservices | RESTful API | On Premise | Penelitian menghasilkan dua buah aplikasi sederhana yang sama, menggunakan dua desain arsitektur yang berbeda yaitu monolith dan microservices. Dipenelitian peneliti menguji performance test dari dua aplikasi tersebut hasilnya aplikasi yang menggunakan arsitektur microservices berjalan lebih cepat. |
| 6 | IMPLEMENTASI ARSITEKTUR MICROSERVICES DALAM PERANCANGAN APLIKASI WEBSITE EDUKASI BAHASA JEPANG UNTUK ANAK DIUSIA DINI | Yahya Pratama | Microservices | RESTful API | Cloud | Penelitian ini akan menghasilkan aplikasi website Edukasi Bahasa Jepang untuk Anak Diusia Dini dengan arsitektur microservices. Aplikasi akan menggunakan RESTful API sebagai protokol komunikasi antara layanan. selain itu dengan penerapan layanan cloud yang ditawarkan oleh Google Cloud Platform (GCP), aplikasi dapat diimplementasikan dan dikelola secara efektif oleh infrastruktur yang disediakan oleh GCP. Dalam peluncurannya aplikasi menjadi lebih efisien karena App Engine secara otomatis menangani aspek skalabilitas sesuai dengan jumlah request. Aplikasi ini juga akan menggunakan HTML, CSS sebagai frontend dan NodeJs sebagai backend.  Perbedaan  - availability dari aplikasi website ini akan terjamin karena infrastruktur it akan dikelola oleh google cloud - update fitur dan perbaikkan bug menjadi lebih mudah karena app engine menyediakan fitur splitting traffic |

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

## **Objek Penelitian**

Objek dalam penelitian ini melakukan Perancangan Aplikasi Website Edukasi Bahasa Jepang Untuk Anak Diusia Dini menggunakan arsitektur *Microservices*

## **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.

### **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.

### **Observasi**

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

### **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.

## **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.

### ***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.

### ***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.

### ***Implementation***

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

### ***Testing***

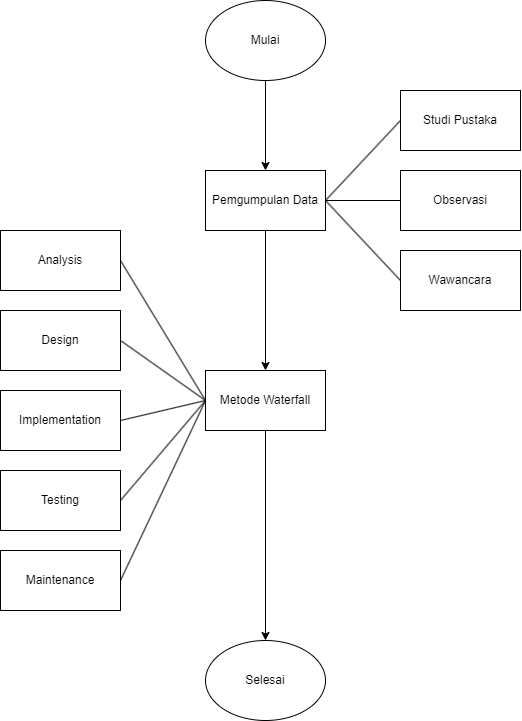
Dalam tahap ini, semua komponen yang telah diimplementasikan akan dilakukan proses pengujian dengan tujuan untuk membuktikan apakah keseluruhan sistem beroperasi secara baik dan sesuai dengan desain yang telah ditetapkan.

### ***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.

## **Alur Penelitian**

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

******

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

* 1. **Analysis**

Pada tahap ini penulis melakukan analisis kebutuhan aplikasi website, ada tiga tahapan yaitu Identifikasi Masalah, dan Identifikasi Sistem Usulan.

* + 1. **Identifikasi Masalah**

Dari wawancara dan observasi yang penulis telah lakukan Saat ini website seperti nhk.or.jp dan duolingo.com ini masih kurang dalam hal interaktif belajar menulis huruf hiragana dan katakana di dalam websitenya. Dan dari wawancara yang telah saya lakukan belajar bahasa jepang itu lebih baik dimulai dengan menulis huruf hiragana dan katakana.

* + 1. **Identifikasi Sistem Usulan**

Aplikasi website yang ingin penulis buat adalah aplikasi website edukasi bahasa jepang dengan Pemetaan Berbasis situs web menggunakan HTML, CSS, dan JavaScript untuk frontendnya dan untuk backendnya penulis akan menggunakan Nodejs dan Express.

Aplikasi website juga akan berfokus dalam hal belajar interaktif menulis huruf hiragana dan katakana yang diharapkan dapat membantu menyelesaikan masalah yang ada.

* 1. **Design**

Pada tahap ini penulis melakukan desain aplikasi website menggunakan UML, dan juga pada tahap ini penulis membuat desain database, dan user interface

* + 1. **Desain UML**

Berdasarkan analisa yang sudah diuraikan sebelumnya, di sub bab ini penulis akan menguraikan mengenai perancangan aplikasi website menggunakan UML (Unified Modeling Language) yang terdiri atas Use Case Diagram, Activity diagram, Class Diagram, dan Sequence Diagram.

* + - 1. **Use Case Diagram**

Use Case Diagram mendeskripsikan interaksi antar aktor/stakeholder yang terlibat dengan sistem ini dan juga peran masing-masing stakholder pada aplikasi website ini. Berikut ini adalah pendefinisian aktor dan deskripsi kegiatan masing-masing aktor pada aplikasi website :

|  |  |  |
| --- | --- | --- |
| **No** | **Aktor** | **Deskripsi** |
| 1 | User | Aktor ini memiliki akses untuk membaca newsfeed/artikel dan belajar menulis. |

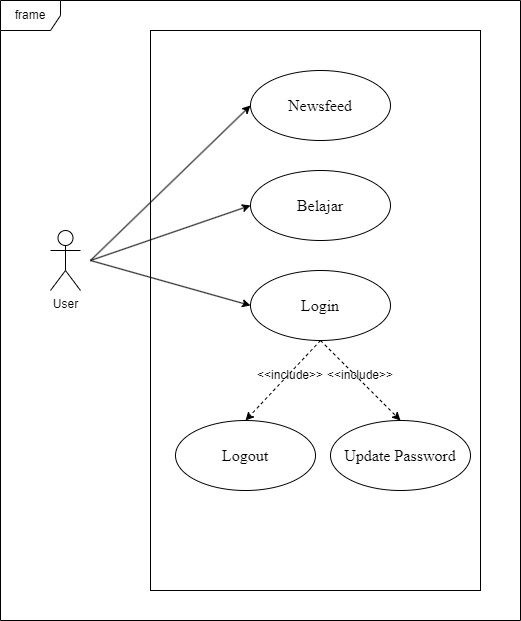
* + - * 1. **Identifikasi Use Case**

Berikut ini adalah pendefinisian use case pada aplikasi website :

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Use Case** | **Deskripsi** | **Aktor** |
| 1 | Login | Use case ini menggambarkan kegiatan untuk memasukkan username dan password agar dapat mengakses website | User |
| 2 | Logout | Use case ini menggambarkan kegiatan untuk mengakhiri interaksi user dengan website | User |
| 3 | Update Password | Use case ini menggambarkan kegiatan untuk mengubah password | User |
| 4 | Belajar | Use case ini menggambarkan kegiatan untuk belajar menulis hirgana atau katakana | User |
| 5 | Newsfeed | Use case ini menggambarkan kegiatan untuk membaca artikel diwebsite | User |

* + - * 1. **Use Case Diagram Aplikasi Website Edukasi Bahasa Jepang**

Setelah penulis mengidentifikasi aktor dan use case dari tiap-tiap aktor yang ada di aplikasi website, hasil identifikasi tersebut dapat digambarkan sebagai berikut:



Gambar 4. *Use Case Diagram*

* + - * 1. **Use Case Scenario**
      1. *Scenario use case login*

|  |  |  |
| --- | --- | --- |
| **No** | *Use Case 01* | |
| **Nama** | *Login* | |
| ***Description*** | Use case ini menggambarkan kegiatan untuk memasukkan username dan password agar dapat mengakses website | |
| ***Actor*** | *User* | |
| **Narasi Utama** | | |
| ***Precondition*** | *-* | |
| ***Typical Course of Events*** | *Actor Actions* | *System Response* |
| 1. Memilih menu *login* | 2. Menampilkan *form* *login* |
| 3. *Input username* dan *password* | 4. Validasi *username* dan *password* |
|  | 5. Menampilkan halaman utama |
| ***Alternate Courses*** | Jika data yang di-*input* oleh *user* tidak *valid* maka website akan menampilkan pesan kesalahan | |
| ***Post Condition*** | Aktor berhasil *login* dan menampilan halaman utama. | |

* + - 1. *Scenario use case logout*

|  |  |  |
| --- | --- | --- |
| **No** | *Use Case 02* | |
| **Nama** | *Logout* | |
| ***Description*** | Use case ini menggambarkan kegiatan untuk melakukan logout | |
| ***Actor*** | *User* | |
| **Narasi Utama** | | |
| ***Precondition*** | *Aktor sudah login* | |
| ***Typical Course of Events*** | *Actor Actions* | *System Response* |
| 1. Memilih tombol *logout* | 2. Menghapus *token jwt* |
|  | 3. Menampilkan halaman *login* |
| ***Alternate Courses*** | - | |
| ***Post Condition*** | Aktor berhasil *logout* dan menampilan halaman login. | |

* + - 1. *Scenario use case update password*

|  |  |  |
| --- | --- | --- |
| **No** | *Use Case 03* | |
| **Nama** | *Update Password* | |
| ***Description*** | Use case ini menggambarkan kegiatan untuk melakukan perubahan password | |
| ***Actor*** | *User* | |
| **Narasi Utama** | | |
| ***Precondition*** | *Aktor sudah login* | |
| ***Typical Course of Events*** | *Actor Actions* | *System Response* |
| 1. Memilih tombol *change password* | 2. Menampilkan halaman mengganti password |
| 3. isi *password* lama dan *password* baru, pilih *submit* | 4. Validasi data |
|  |  | 5. Menyimpan data *password* baru |
|  |  | 6. Menampilkan halaman *home* |
| ***Alternate Courses*** | Jika data yang diinput *Aktor* salah maka website akan menampilkan kesalahannya | |
| ***Post Condition*** | Aktor berhasil mengubah *password* | |

* + - 1. *Scenario use case belajar*

|  |  |  |
| --- | --- | --- |
| **No** | *Use Case 04* | |
| **Nama** | *Belajar* | |
| ***Description*** | Use case ini menggambarkan kegiatan untuk melakukan belajar menulis huruf hiragana atau katakana | |
| ***Actor*** | *User* | |
| **Narasi Utama** | | |
| ***Precondition*** | *Aktor sudah login* | |
| ***Typical Course of Events*** | *Actor Actions* | *System Response* |
| 1. Memilih tombol *change password* | 2. Menampilkan halaman mengganti password |
| 3. isi *password* lama dan *password* baru, pilih *submit* | 4. Validasi data |
|  |  | 5. Menyimpan data *password* baru |
|  |  | 6. Menampilkan halaman *home* |
| ***Alternate Courses*** | Jika data yang diinput *Aktor* salah maka website akan menampilkan kesalahannya | |
| ***Post Condition*** | Aktor berhasil mengubah *password* | |

* + - 1. *Scenario use case newsfeed*
      2. **Activity Diagram**
      3. **Class Diagram**
      4. **Sequence Diagram**
    1. **Desain Database**
    2. **Desain User Interface**
  1. **Implementation**
  2. **Testing**
  3. **Maintenance**

# **DAFTAR PUSTAKA**

Ahmad, I., Suwarni, E., Borman, R. I., Asmawati, Rossi, F., & Jusman, Y. (2021). Implementation of RESTful API Web Services Architecture in Takeaway Application Development. *2021 1st International Conference on Electronic and Electrical Engineering and Intelligent System, ICE3IS 2021*, 132–137. https://doi.org/10.1109/ICE3IS54102.2021.9649679

Aroral, H. K. (2021). Waterfall Process Operations in the Fast-paced World: Project Management Exploratory Analysis. *International Journal of Applied Business and Management Studies*, *6*(1).

Astya, P. N., Galgotias University. School of Computing Science and Engineering, Institute of Electrical and Electronics Engineers. Uttar Pradesh Section, & Institute of Electrical and Electronics Engineers. (2017). *IEEE International Conference on Computing, Communication and Automation (ICCCA 2017) : proceeding : on 5th-6th May, 2017*.

Challita, S., Zalila, F., Gourdin, C., & Merle, P. (2018). A precise model for Google cloud platform. *Proceedings - 2018 IEEE International Conference on Cloud Engineering, IC2E 2018*, 177–183. https://doi.org/10.1109/IC2E.2018.00041

EF. (2023). *EF EPI EF English Profi ciency Index A Ranking of 113 Countries and Regions by English Skills*. www.efset.orgwww.ef.com/epi

Gullberg, M., Roberts, L., Dimroth, C., Veroude, K., & Indefrey, P. (2010). *Adult Language Learning After Minimal Exposure to an Unknown Natural Language*.

Gulnaz Beybitzhanovna, B. (2023). *The importance of effective foreign language teaching methods*. https://creativecommons.org/licenses/by-sa/4.0/

Harsono, H. (2022). FAKTOR-FAKTOR YANG MEMPENGARUHI SISTEM INFORMASI BERBASIS KOMPUTER: SISTEM OPERASI, SERVER, DAN PROGRAMMER (LITERATURE REVIEW EXECUTIVE SUPPORT SISTEM FOR BUSINESS). *Jurnal Manajemen Pendidikan Dan Ilmu Sosial*, *3*(2). https://doi.org/10.38035/jmpis.v3i2

Jack, D. (2016). *Chicago Unbound The Issue of Japan’ s Aging Population*. http://www.stat.go.jp/english/data/nenkan/pdf/yhyou02.pdf

Jamison, D. C. (2003). *Structured Query Language (SQL) Fundamentals*.

Jeong, S. Y., Xie, Y., Beaton, J., Myers, B. A., Stylos, J., Ehret, R., Karstens, J., Efeoglu, A., & Busse, D. K. (2009). Improving Documentation for eSOA APIs through User Studies. In *LNCS* (Vol. 5435). http://www.w3.org/TR/wsdl

Johannes, D., Khomh, F., & Antoniol, G. (2019). A large-scale empirical study of code smells in JavaScript projects. *Software Quality Journal*, *27*(3), 1271–1314. https://doi.org/10.1007/s11219-019-09442-9

Koishybayev, I., & Kapravelos, A. (2020). *Mininode: Reducing the Attack Surface of Node.js Applications*. https://kapravelos.com/projects/mininode

Kratzke, N. (2018). A brief history of cloud application architectures. In *Applied Sciences (Switzerland)* (Vol. 8, Issue 8). MDPI AG. https://doi.org/10.3390/app8081368

Newman, S. (2015). *Building Microservices*. http://safaribooksonline.com

Rawat, B., & Purnama, S. (2021). MySQL Database Management System (DBMS) On FTP Site LAPAN Bandung. *International Journal of Cyber and IT Service Management (IJCITSM)*, *1*(2), 173–179. https://doi.org/10.34306/ijcitsm.v1i1.47

Rinaldy Malik, A., & Nur Ashar Asnur Pendidikan Bahasa Jerman, M. (2019). *BAHTERA: Jurnal Pendidikan Bahasa dan Sastra*. *18*. http://journal.unj.ac.id/unj/index.php/bahtera/

Robillard, M. (2009). What Makes APIs Hard to Learn? The Answers of Developers. *IEEE Software*. https://doi.org/10.1109/ms.2009.116

Sohail Malik, R., Patra, J., & Pradel, M. (2019). *NL2Type: Inferring JavaScript Function Types from Natural Language Information*.

Sturm, R., Pollard, C., & Craig, J. (2017). Managing Web-Based Applications. In *Application Performance Management (APM) in the Digital Enterprise* (pp. 83–93). Elsevier. https://doi.org/10.1016/b978-0-12-804018-8.00007-3

Subekti, N. A. (2019). *KEBERADAAN BAHASA INDONESIA DAN BAHASA INGGRIS DALAM MENUNJANG ILMU PENGETAHUAN DI ERA GLOBAL*.

Sunarti, Rangga R.Y, & Dedi Damhudi. (2016). Aplikasi Pembelajaran Huruf Hiragana dan Katakana Dilengkapi dengan Suara Berbasis Android. *Jurnal Sains, Teknologi Dan Industri*, *14*, 9–16.

**LAMPIRAN**

**Lampiran I. Wawancara Ahli**

Tujuan : Mendapatkan informasi mengenai topik belajar yang cocok untuk anak-anak dan aspek apa yang perlu diprioritaskan dalam pengembangan aplikasi website.

Tempat : Dilakukan melalui aplikasi video conference ZOOM Meeting.

Waktu : Jam 12.00 WIB, Senin, 26 Februari 2024

Durasi : 10 menit

Narasumber : Ibu Annisya Pramesti Husniadhy S.Pd (Guru Bahasa Jepang)

1. Penanya : Halo selamat pagi Bu Annisya.

Narasumber : Halo selamat pagi suara saya kedengaran ?

1. Penanya : Iya kedengaran.

Narasumber : Oke baik.

1. Penanya : Oke sebelumnya saya ingin berterima kasih kepada ibu karena telah bersedia untuk diwawancara hari ini, oh iya bu untuk sesi wawancara akan direkam apakah ibu keberatan ?

Narasumber : Engga kok silakan aja

1. Penanya : Oke ini saya langsung mulai ke pertanyaan yang pertama ya bu

Narasumber : Apa saya harus perkenalan diri dulu atau engga usah

1. Penanya : Oh boleh-boleh silakan jika ibu berkenan

Narasumber : Jadi perkenalkan nama saya Annisya Pramesti background saya yaitu guru Bahasa jepang saya lulusan dari universitas negeri Jakarta jurusan Pendidikan Bahasa jepang nah saya dibidang ini udah dari sebelum saya lulus sampai saat ini mungkin kurang lebih 3 tahun ya seperti itu dan murid yang saya ajar macem-macem ya saya pernah mengajar murid sd kelas 1, kelas 3, kelas 5, SMP, SMA sampai Karyawan pun pernah. Mohon bantunnya untuk hari ini salam kenal.

1. Penanya : Sungguh banyak pengalamannya juga ini ya ibu Annisya ternyata

Narasumber : Ya lumayan lah.

1. Penanya : Oke ini saya langsung mulai ke pertanyaan yang pertama ya bu.

Narasumber : Ya boleh

1. Penanya : Dalam pengalaman Ibu sebagai guru bahasa Jepang, ibu bilang punya pengalaman mengajar kepada anak-anak ya bu itu tantangan utamanya apa ya bu ?

Narasumber : Tantangan utamanya sebenarnya ini karena mereka ini engga suka belajar, kaya belajar yang monoton kalo misalnya SMP, SMA, bahkan Mahasiswa atau Karyawan itu kan kalo belajar yang memang apa ya kalo dibilang cepet bosen juga eh apasih bosen tapi mereka melakukan belajarnya dengan baik tapi kalo misal anak SD ini yang umurnya misalkan kelas 1 tuh umur berapa saya lupa, Sekarang saya lagi ngajar anak kelas 3 SD umur 9 tahun dia itu memang gampang bosen dalam belajar Bahasa jepang itu maksudnya misalnya kita kan kalo disekolah dengerin guru atau dosen yang ngejelasin gitu dan itu memang harus didengarkan kalo mereka itu kaya gampang bosen gitu gampang kaya “gimana kalo kita main aja” kaya gitu jadi harus di apa ya. Jadi harus dibarengi sama bermainnya mereka jadi selama ini saya ngajar sebenarnya bermain sambil belajar sih kaya misalnya untuk Bahasa jepang sendiri yang pertama perlu banget dipelajari itu hurufnya ya huruf jepang ada tiga hiragana, katakana dan, kanji itu harus dipelajari dulu untuk orang yang mau belajar Bahasa jepang jadi pertama hiragana ya hiragana saya waktu itu ngajarin dengan cara main jadi saya kenal dulu huruf-hurufnya jadi sehari misal Cuma 5 atau 10 tergantung dari siswa bisa menerima seberapa banyak nah itu dengan cara bermain jadi ini huruf A ini huruf I U E O kaya gitu dan kemudian dipake kartu kita tulis hurufnya satu-satu gitu terus kita acak dan kita tanya kemereka coba tunjuk yang A yang mana gitu yang I yang mana gitu terus nanti kita acak kartunya mereka tunjuk urutan A I U E O yang mana gitu mereka itu lebih seneng kaya gitu belajarnya dari pada kita ngejelasin doang atau suruh di baca buku aja jadi lebih menarik bagi mereka itu sambil main jadi tantangannya karena mereka lebih suka main dari pada belajar yah jadi harus dibarengi dengan belajarnya juga.

1. Penanya : Jadi menarik juga ya ?

Narasumber : Iya bener harus menarik.

1. Penanya : Nah disini saya ada pertanyaan lagi bu, Saya ingin bertanya pendapat ibu nih tentang penggunaan teknologi dalam membantu pembelajaran Bahasa jepang bagi anak-anak itu gimana bu ? kaya misalnya website belajar atau game gitu bu menurut ibu gimana ?

Narasumber : Kalo website belajar kayanya lebih cocok untuk anak yang memang sudah biasa menggunakan internet dan mencari tahu sendiri tentang apa yang pengen mereka cari tahu cuma kalo selama pengalaman saya itu tuh lebih cocok kegame sih ya jadi mungkin ada game. Kan kalo sekarang tuh layar itu bisa ditouchscreen misalkan cara nulis mereka bisa pakai jarinya nulis huruf A sesuai urutan abis itu kalo misalkan benar bisa ditampilkan “selamat kamu berhasil” gitu atau emot senyum atau ada efek sparkle-sparkle yang bikin mereka seneng gitu karena sesuatu yang eye-catching tuh mereka suka nah terus sangat membantu sih kalo menurut saya untuk teknologi apa lagi kalo bisa mengajak anak itu bermain sambil belajar gitu kaya waktu itu saya ngajarin hurufkan lewat flash card ya dan itu masih manual itu kurang apa ya sebenarnya seru juga sih cuma mungkin anak jaman sekarang tuh udh mengenal gadget mungkin akan lebih seru lagi kalo ada gamenya gitu ada suara atau ada lebih desain lebih menarik dan warna-warna lebih menarik mungkin itu sangat membantu anak dalam belajar apa lagi kalo cuma diklik-klik aja kan kaya coba tebak yang mana huruf A misalnya gitu mereka tinggal klik gitu nanti muncul apa reward atau pujian-pujiannya

1. Penanya : Berarti Bu Annisya ini melihat positif penggunaan teknologi terhadap motivasi dan minat belajar anak-anak terhadap Bahasa Jepang?

Narasumber : Iya hmm, saya soalnya selain manual waktu itu juga pernah pengan kelas untuk anak SD yang online jadi karena online kan engga mungkin manual bawa-bawa kartu dan sebagainya saya waktu itu pakai powerpoint sih ya cuma kelemahan powerpoint itu kita sebagai orang yang punya powerpoint yang apa sih yang nanya ini mereka “yang mana nih yang huruf A ?” terus mereka jawab “itu bu yang sebelah kiri” kaya apa ya kurang enak gitu lebih enak kalo mereka milih sendiri atau nunjuk sendiri kaya gitu sih mungkin bisa dikembangan lewat aplikasi atau ada website yang bisa seperti itu akan sangat membantu.

1. Penanya : Oke bu, berarti disini saya ingin bertanya pertanyaan terakhir bu, kan disini penelitian saya ini saya akan pengembangan aplikasi website nih bu menurut ibu aspek yang paling prioritaskan dari pengembangan aplikasi ini apa bu dari pengalaman ibu sendiri ?

Narasumber : Kaya kita ngomongin hal dasar yang perlu dipelajar sama pembelajar bahasa jepang dulu kali ya yaitu tentang huruf hiragana kemudian katana dan baru kekanji mungkin yang perlu ada di website itu seperti cara menulis kemudian latihan tulisnya kemudian mungkin ada flashcard yang tadi saya jelaskan misalnya nanti pilih yang huruf A yang mana terus nanti huruf A I U E O diacak dan mereka tinggal klik-klik gitu sesuai urutan yang tetap itu gimana jadi buat lebih menarik aja sih soalnya kalo belajar huruf kadang kaya bosen ya kapan nih mulai belajar bahasa jepang pada lah belajar huruf itu sangat penting bagi semua pembelajar bahasa jepang gitu engga terkecuali umur sekian-sekian dan ini kali ya warna mereka ini suka warna yang eye-catching ya engga yang monoton atau yang monochrome misalnya hitam putih jadinya warna yang eye-catching bikin mereka itu seneng juga ngeliatin websitenya jadi engga bosen gitu.

1. Penanya : Oke sih bu itu pertanyaan terakhir. Sebelumnya saya yang berterima kasih nih bu untuk waktu dan wawasan sangat berharga sekali bu.

Narasumber : Oke baik terima kasih kembali ya.

1. Penanya : Sebelumnya diakhir bu ini wawancara, ibu tidak keberatan kan wawancara ini akan dijadikan sebagai data dalam penelitaan saya.

Narasumber : Oke silakan.

1. Penanya : Kalo begitu sekali lagi terima kasih bu sangat membantu sekali wawasannya terlebih lagi dari ibu nih yang pengalamannya banyak banget

Narasumber : Terima kasih kembali ya.

1. Penanya : Terima kasih yang bu saya izin matikan recordnya