

Sistem Penerjemah Percakapan Langsung Multibahasa untuk Mengatasi Masalah Kendala Bahasa

Multi-Language Live Conversation Translator System to Overcome Language Barrier Issues

Averill Saladin Atma Setiawan, Mychael Maoeretz Engel*

Program Studi Informatika, Universitas Ciputra Surabaya, Surabaya 60219, Indonesia

(*Email Korespondensi: mychael.engel@ciputra.ac.id)

Abstrak: Kendala bahasa merupakan permasalahan yang sering ditemui, terutama ketika berkomunikasi dengan orang dari luar negeri. Kendala ini dapat disebabkan oleh berbagai faktor, baik internal maupun eksternal dan dapat terjadi kapan saja. Kendala ini juga dapat ditemukan sebagai salah satu akar permasalahan lain di berbagai sektor. Bahkan dengan solusi yang ada, menggunakan penerjemah, masih saja muncul miskomunikasi jika terjemahan yang dihasilkan tidak menangkap konteks pembicaraan. Oleh karena itu, penelitian ini dilakukan dengan tujuan untuk mengembangkan aplikasi penerjemah percakapan langsung yang dapat membantu orang melakukan percakapan langsung. Pengembangan aplikasi dilakukan dengan menggunakan model *waterfall* dan memanfaatkan layanan ChatGPT sebagai layanan penerjemahan berbasis AI. Hasil pengujian yang diperoleh selama pengujian aplikasi menunjukkan kepuasan yang tinggi. Hasil yang diperoleh dapat menjadi indikator keberhasilan pendekatan yang diusulkan dalam membantu menyelesaikan masalah kendala bahasa di masyarakat.

Kata Kunci: aplikasi penerjemah, kendala bahasa, percakapan langsung, model *waterfall*

Abstract: Language barriers are a problem that is often encountered, especially when communicating with people from abroad. This problem can be caused by various factors, both internal and external and can occur at any time. This problem can also be found as one of the roots of other problems in various sectors. Even with the existing solutions, using translator, there is still miscommunication to appear if the resulting translation does not capture the context of the conversation. Therefore, this research was conducted with the aim of developing a live conversation translator application that can help people have live conversations. Application development was carried out using the Waterfall model and utilizing the ChatGPT service as an AI-based translation service. The testing results obtained during application testing showed high satisfaction. The results obtained can be an indicator of the success of the approach proposed in helping to solve the problem of language barriers in society.

Keywords: translator application, language barriers, live conversation, waterfall model

Naskah diterima 29 April 2025; direvisi 26 Mei 2025; dipublikasi 30 Mei 2025.
JUI SI is licensed under a Creative Commons Attribution-ShareAlike 4.0 International License.



1. Pendahuluan

Kendala bahasa merupakan suatu masalah umum yang dapat terjadi dan disebabkan oleh berbagai hal seperti ketidakmampuan pembicara atau lawan bicara dalam memahami komunikasi yang dilakukan, baik karena perbedaan bahasa, keterbatasan kosakata (Reale, 2023), maupun perbedaan logat atau tata krama dalam mengucapkan kata-kata (Buarqoub, 2019). Masalah ini sendiri merupakan akar dari masalah-masalah lain yang berkaitan dengan komunikasi dan dapat menimbulkan konflik yang muncul akibat kesalahpahaman dan miskomunikasi (Shamsi et al., 2020). Berdasarkan penelitian yang dilakukan oleh Buarqoub (2019), de Moissac dan Bowen (2019), diketahui bahwa

kendala bahasa dapat muncul dalam berbagai bidang komunikasi, antara lain bisnis, pelayaran, penerbangan, pendidikan, kesehatan, dan lain sebagainya. Meskipun dampak yang dapat terjadi tidak sepenting pada bidang-bidang tersebut, namun pemahaman terhadap masalah kendala bahasa tidak hanya berlaku dalam bidang profesional. Masalah ini juga dapat muncul dalam kehidupan sehari-hari masyarakat.

Wawancara dilakukan kepada beberapa masyarakat Indonesia yang berada di luar negeri, terkait kehidupan dan interaksi mereka dengan masyarakat sekitar. Mereka merasa kesulitan saat pertama kali mencoba berinteraksi dengan masyarakat setempat, apalagi jika masyarakat tersebut tidak menggunakan bahasa Inggris sebagai bahasa utama mereka. Selain itu, mereka juga merasakan adanya kendala dalam berkomunikasi jika cenderung berbicara dengan nada bicara atau logat lokal. Meskipun mereka telah mempelajari bahasa baku yang digunakan masyarakat setempat sebelum berangkat ke luar negeri menuju negara tujuan, mereka merasa bahwa apa yang mereka pelajari terkadang tidak selalu dapat diaplikasikan pada kejadian nyata yang mereka alami.

Penggunaan alat bantu seperti penerjemah terkadang dapat membantu percakapan. Akan tetapi, penggunaan alat bantu tersebut masih menjadi kendala dalam percakapan langsung karena memiliki masalah latensi penerjemahan. Hal ini membuat mereka takut jika menggunakan alat bantu penerjemahan akan membuat percakapan menjadi canggung dan tidak nyaman. Perkembangan teknologi dapat menjadi solusi untuk hal tersebut. Pemanfaatan teknologi aplikasi seluler dan teknologi *cloud* yang sering digunakan untuk menyelesaikan berbagai permasalahan, seperti sistem pengawas ujian *online* (Engel dan Agustinus, 2024), sistem integrasi data berbasis *cloud computing* untuk organisasi non-profit (Engel et al., 2023), sistem peminjaman *self-service* pada perpustakaan (Engel dan Lie, 2022) dan aplikasi pelatihan berbicara di depan umum (Lordianto dan Engel, 2023), juga dapat digunakan untuk mengatasi permasalahan kendala bahasa. Pemanfaatan teknologi AI terkini juga dapat dimanfaatkan untuk mengembangkan sistem penerjemahan. Pemanfaatan teknologi AI dari salah satu pelopornya, ChatGPT, yang dapat memahami konteks perintah yang dikirim akan membantu sistem penerjemahan untuk menerjemahkannya dengan lebih akurat.

Penelitian lain yang membahas tentang sistem penerjemahan juga telah dilakukan untuk mencari solusi terbaik dalam mengatasi permasalahan tersebut. Pada penelitian yang dilakukan oleh Alfarisi (2020), peneliti menggunakan teknologi *Optical Character Recognition* milik Tesseract untuk menerjemahkan aksara Jepang. Pendekatan ini dapat melihat hasil terjemahan dari aksara cetak yang telah dikenali oleh Tesseract OCR. Sebaliknya, penelitian yang dilakukan Lengkon et al. (2022) memiliki pendekatan yang berbeda, yaitu menggunakan teknologi *speech-to-text* dari perangkat seluler untuk mengenali suara. Pengenalan yang dihasilkan akan diterjemahkan ke dalam bahasa lain. Meskipun banyak terjadi kesalahan akibat tidak dapatnya *input* mikrofon dari MIT App Inventor, namun hasil penerjemahan dari layanan penerjemahan Yandex dapat berjalan dengan cukup baik. Selain itu, penelitian oleh Ogundokun et al. (2021) menghasilkan sistem penerjemah dengan memperhatikan emosi dari kalimat yang ingin diterjemahkan menggunakan Watson *Natural Language Comprehension*. Berdasarkan beberapa penelitian di atas, dapat disimpulkan bahwa penyelesaian masalah terkait kendala bahasa dapat dilakukan dengan berbagai pendekatan yang saling mendukung satu sama lain.

Akhirnya, penelitian ini dilakukan dengan tujuan untuk mengetahui kesesuaian pengembangan aplikasi penerjemah dengan pendekatan baru menggunakan *direct voice-based translation* yang dapat membantu pengguna melakukan percakapan secara langsung. Penelitian ini juga memanfaatkan layanan penerjemahan dari ChatGPT API sebagai layanan penerjemahan utama dan dikembangkan dengan aplikasi *mobile* berbasis iOS yang memanfaatkan teknologi dari Apple untuk mengurangi kemungkinan terjadinya kesalahan transkripsi saat menggunakan aplikasi.

2. Kajian Pustaka

2.1 Studi Terdahulu

Penelitian terkait aplikasi penerjemah sering kali dilakukan oleh peneliti internasional dan nasional. Alfarisi (2020) melakukan penelitian dengan judul “Rancang Bangun Aplikasi Terjemahan Bahasa Jepang - Indonesia Berbasis Android Menggunakan Tesseract OCR”. yang mengangkat pemanfaatan teknologi Tesseract OCR dalam identifikasi aksara dalam Bahasa Jepang. Penelitian ini bermaksud untuk melihat manfaat penggunaan teknologi

OCR dan layanan penerjemah Yandex Translation API. Hasil yang didapat dalam penelitian ini cukup baik dengan performa terjemahan Yandex Translation API memiliki kecocokan di angka 85% ketika dibandingkan dengan hasil terjemahan dari Google Translate. Selain itu performa pengenalan karakter oleh Tesseract OCR memiliki persentase 95% dalam pengenalan 20 data yang berisi kalimat dengan berbagai jenis karakter seperti Hiragana, Katakana dan Kanji.

Penelitian lain yang dilakukan oleh (Lengkong et al., 2022) yang berjudul “The Implementation of Yandex Engine on Live Translator Application for Bahasa and English Using Block Programming MIT App Inventor Mobile Based” berfokus pada pemanfaatan layanan Yandex pada pengembangan aplikasi penerjemah bahasa berbasis *mobile* menggunakan MIT App Inventor. Peneliti melihat performa penerjemah berbasis teks antara Bahasa Indonesia dan Bahasa Inggris yang dilakukan menggunakan Yandex sudah cukup baik. Proses pengembangan aplikasi menggunakan MIT App Inventor, yang sebenarnya sedang dalam tahap pengembangan juga, saat itu pun dapat men-*deliver* kebutuhan dari aplikasi seperti akses mikrofon dan lain sebagainya. Namun tidak semua hasil terjemahan suara yang dilakukan, terutama di beberapa kata dan kalimat spesifik, dapat diterjemahkan dengan baik. Terkadang ada beberapa galat yang terjadi yang disebabkan dari kurang baiknya performa mikrofon sehingga menghambat kinerja dari Yandex itu sendiri.

Penelitian yang dilakukan oleh (Ogundokun dkk., 2021) dengan judul “An Android Based Language Translator Application” diusung dengan tujuan membantu masyarakat Nigeria untuk dapat memahami 520 bahasa yang digunakan di Nigeria. Pendekatan yang dilakukan oleh peneliti berupa terjemahan berbasis teks dan pendeteksi teks pada gambar. Penelitian ini memanfaatkan layanan dari IBM Cloud Kubernetes seperti *Watson Language Converter* untuk menerjemahkan antar bahasa, *Watson Natural Language Comprehension* untuk menangkap emosi dari teks yang diterjemahkan, dan Tesseract OCR untuk rekognisi karakter pada gambar yang diunggah oleh pengguna. Peneliti melakukan pengujian pada 3 variabel, yaitu akurasi hasil terjemahan, waktu penerjemahan, dan penggunaan memori. Hasil yang didapatkan dari penelitian ini dinilai cukup baik oleh peneliti dengan nilai akurasi terjemahan 99% akurat dengan waktu penerjemahan relatif lebih cepat, yaitu terjemahan berbasis teks selesai dalam waktu 47 detik dan berbasis gambar selesai dalam 1.25 menit.

Penelitian-penelitian di atas memiliki beberapa kesamaan salah satunya adalah platform yang digunakan adalah perangkat *mobile*. Hal ini dapat menjadi faktor bahwa pemilihan teknologi aplikasi *mobile* sebagai solusi paling tepat dalam implementasi aplikasi terjemahan bahasa dibutuhkan secara instan dan cepat. Selain itu, perbedaan penggunaan metode untuk aplikasi penerjemah bahasa seperti rekognisi karakter dan *live translation* yang digunakan pada penelitian di atas membuktikan bahwa penyelesaian terhadap masalah bahasa seperti *language barrier* dapat diselesaikan dengan berbagai pendekatan.

2.2 Aplikasi Penerjemah Bahasa

Aplikasi penerjemah bahasa merupakan aplikasi yang digunakan untuk membantu penggunanya dalam menerjemahkan bahasa ke bahasa yang dituju. Menurut Pratheeksha et al. (2020) aplikasi penerjemah bahasa bekerja dengan menerjemahkan satu bahasa ke bahasa lain secara otomatis.

Penelitian terkini tentang aplikasi penerjemahan bahasa menyoroti potensinya untuk mengatasi hambatan komunikasi dalam area linguistik yang beragam. Aplikasi semacam ini memanfaatkan teknologi canggih seperti *Artificial Intelligence* (AI), *Natural Language Processing* (NLP), dan *Machine Learning* (ML) untuk menyediakan terjemahan teks lisan dan tertulis secara *real-time* (Kothari et al., 2024; Patil & Dudhe, 2024). Beberapa aplikasi berfokus pada pasangan bahasa tertentu, seperti bahasa Bangka ke bahasa Indonesia (Razsiah et al., 2023) atau bahasa global ke bahasa India (Meesala et al., 2024). Aplikasi lainnya menawarkan kemampuan multibahasa, melayani wisatawan, dan memfasilitasi komunikasi lintas budaya (Saila et al., 2023). Namun, pada akhirnya beragam aplikasi penerjemahan ini bertujuan untuk meningkatkan pengalaman pengguna, mendorong proses komunikasi yang bebas stres, dan menyederhanakan pembelajaran bahasa.

Pada penelitian ini, aplikasi yang diusulkan menggunakan metode penerjemahan antar bahasa secara langsung untuk memudahkan pengguna dalam memungkinkan komunikasi langsung antara dua pihak dengan bahasa yang berbeda.

2.3 OpenAI dan ChatGPT

OpenAI merupakan sebuah perusahaan yang memiliki fokus dalam riset dan pengembangan *Artificial Intelligence* (AI) yang memiliki tujuan untuk memastikan manfaat penggunaan AI bagi manusia (OpenAI, t.t.). OpenAI terkenal dalam peran mereka sebagai pionir dalam mengembangkan berbagai macam AI untuk memudahkan pekerjaan manusia, beberapa diantaranya adalah Point-E, Dall-E, CLIP, Whisper, dan ChatGPT. OpenAI menyediakan layanan API untuk pengembang dapat memanfaatkan layanan AI yang mereka sediakan.

ChatGPT merupakan salah satu produk unggulan dari OpenAI yang diklaim dapat membantu semua permasalahan penggunanya melalui layanan berbasis percakapan. ChatGPT dikembangkan dengan metode *Reinforcement Learning from Human Feedback* (RLHF) sehingga dapat mengikuti instruksi yang diberikan oleh pengguna dan dapat merespon secara detail serta berkelanjutan. Cara kerja ChatGPT adalah dengan memproses suatu perintah yang diberikan oleh pengguna dan akan membuat teks berdasarkan perintah tersebut. Dalam penelitian ini, ChatGPT akan digunakan sebagai penyedia layanan dalam fitur penerjemah dan ringkasan yang berbasis AI. Penggunaan layanan ChatGPT dilakukan melalui API yang disediakan oleh OpenAI.

2.4 iOS Development

iOS merupakan sebuah sistem operasi (OS) yang dibuat khusus untuk pengembangan perangkat *mobile* Apple, yaitu iPhone. iOS yang menempati peringkat ke-2 dalam *market share* menurut StatCounter (2024). Perkembangan iOS masih menjadi salah satu OS dengan teknologi terbaik dan dipilih masyarakat karena dapat memperhatikan kenyamanan dan keamanan penggunanya.

2.4.1 Xcode

XCode adalah IDE (Integrated Development Environment) khusus yang dikembangkan Apple untuk mengembangkan aplikasi iOS, macOS, WatchOS, dan tvOS. XCode sendiri hanya dapat diakses melalui Apple App Store untuk macOS. Pada penelitian ini, XCode digunakan sebagai IDE yang digunakan untuk mengembangkan aplikasi.

2.4.2 Swift

Swift merupakan bahasa pemrograman modern yang dikembangkan oleh Apple dan dirancang untuk mengembangkan aplikasi berbasis iOS. Swift sendiri dinilai dapat dijalankan dengan lebih cepat dan efisien untuk pengembangan aplikasi iOS serta didukung dengan adanya forum yang bersifat terbuka sebagai sarana diskusi (Apple Inc., t.t.-a). Pada penelitian ini, Swift digunakan sebagai bahasa pemrograman utama untuk pengembangan aplikasi iOS dengan arsitektur *Model View ViewModel* (MVVM).

2.4.3 SwiftUI

SwiftUI merupakan sebuah *framework* untuk mendukung pengembangan dari tampilan antarmuka pengguna (user interface) dalam pengembangan aplikasi iOS. SwiftUI didesain dengan mengutamakan kenyamanan dan kemudahan pengembang dalam membuat tampilan antarmuka pengguna yang dapat membuat tampilan aplikasi yang kompleks dengan lebih sedikit baris kode (Apple Inc., t.t.-b). Pada penelitian ini, SwiftUI digunakan sebagai *framework* yang menjadi pembangun dasar dari antarmuka pengguna aplikasi iOS.

2.5 Apple Speech Recognizer

Apple *Speech Recognizer* merupakan sebuah *framework* yang dikembangkan untuk dapat menerima *input* berupa audio dan mentranskripsikannya menjadi bentuk teks (Apple Developer Documentation, t.t.-a). *Framework* ini dalam penelitian ini digunakan untuk mendukung fitur percakapan dalam aplikasi yaitu menerima *input* suara dari pengguna dan mengirimkan hasil transkripsinya ke layanan penerjemah pada fitur *speech-to-text*.

2.6 Apple Speech Synthesizer

Apple *Speech Synthesizer* merupakan *framework* yang mengatur sintesis suara. *Framework* ini berfokus di 2

langkah utama yaitu mengonfigurasi pengucapan melalui kelas AVSpeechUtterance dan memproduksi pengucapan tersebut melalui AVSpeechSynthesizer (Apple Developer Documentation, t.t.-b). Pada penelitian ini, *framework* Apple *Speech Synthesizer* digunakan dalam fitur *text-to-speech* yang digunakan untuk mensintesis suara dari teks hasil terjemahan layanan penerjemah.

2.7 Firebase

Firebase merupakan sebuah layanan *backend* besutan Google yang dapat membantu *developer* untuk dapat mengatur berbagai hal yang bisa dilakukan di sisi *backend*. Sebagai layanan *backend*, Firebase memiliki berbagai jenis layanan mulai dari layanan untuk tahap *development* seperti Firestore Database, layanan otentikasi pengguna menggunakan Authentication, hingga layanan untuk memonitor dan mengatur *engagement* terhadap pengguna seperti *Messaging* dan *A/B Testing* (Firebase, t.t.-c). Firebase juga menawarkan integrasi dengan produk Google lain seperti Google Analytics untuk pengembangan berkelanjutan. Pada penelitian ini, *service* Firebase yang digunakan adalah Firebase Firestore dan Firebase Authentication.

2.7.1 Firebase Firestore

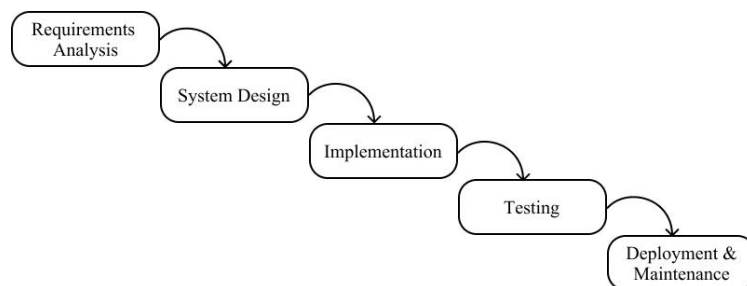
Firebase Firestore merupakan layanan penyimpanan data yang ada pada Firebase. Firestore sendiri memiliki basis data yang bersifat NoSQL atau tidak terstruktur. Keunggulan basis data yang tidak terstruktur ini lebih bersifat dinamis dan memiliki skalabilitas yang tinggi juga. Selain itu, Firestore juga menawarkan penyimpanan data yang *real-time* dan dapat diakses di berbagai platform (Firebase, t.t.-b). Pada penelitian ini, Firestore Database digunakan sebagai basis data untuk menyimpan terjemahan dan riwayat terjemahan pengguna.

2.7.2 Firebase Authentication

Firebase juga menyediakan layanan otentikasi dengan berbagai macam cara seperti penggunaan *email* dan *password*, masuk melalui akun Google, Apple, Facebook, nomor telepon, dan lain sebagainya. Layanan *Authentication* diklaim mampu untuk diimplementasikan di banyak *platform* dengan cepat dan mementingkan sekuritas dari data penggunaannya (Firebase, t.t.-a). Penggunaan *Authentication* pada penelitian ini adalah sebagai layanan otentikasi dan pembeda data terjemahan sesuai *role* pengguna.

3. Metode Penelitian

Metode penelitian yang digunakan sering kali menggunakan metode-metode tertentu sesuai dengan tingkat kesulitan dan jumlah orang yang terlibat. Metode ini sering disebut dengan *Software Development Life Cycle* (SDLC). Penggunaan SDLC sendiri bertujuan untuk dapat merancang, membangun, dan memelihara suatu sistem dengan lebih rapi dan terstruktur. Dalam penelitian ini, metode SDLC yang digunakan adalah *Waterfall*. Metode *Waterfall* sendiri mengharuskan setiap proses dipenuhi secara berurutan dan independen, yang berarti bahwa setiap tahapan harus diselesaikan sebelum memasuki tahap berikutnya (Gupta, 2021) (Pargaonkar, 2023). Langkah-langkah yang digunakan dalam metode *Waterfall* dapat dilihat pada Gambar 1 di bawah ini.



Gambar 1. Metode *waterfall*

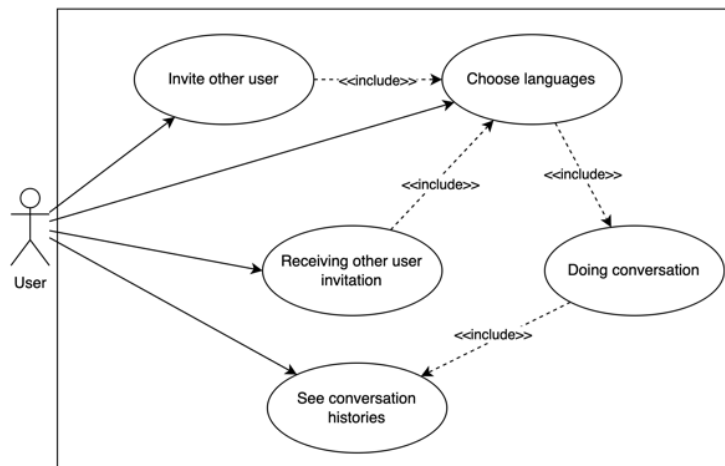
3.1 Requirement Analysis

Dalam penelitian ini, pengumpulan data terkait akar permasalahan dilakukan dengan menggunakan pendekatan kualitatif, yaitu wawancara. Wawancara dilakukan kepada masyarakat Indonesia yang tinggal di luar negeri untuk melihat bagaimana mereka berinteraksi dengan masyarakat setempat, baik dalam hal pekerjaan maupun kehidupan sehari-hari. Peneliti menanyakan kendala komunikasi apa saja yang sering mereka hadapi dan bagaimana cara mengatasinya. Hal ini dilakukan untuk mengetahui solusi yang digunakan guna melihat potensi pengembangannya.

Berdasarkan hasil wawancara yang telah dilakukan, peneliti melihat bahwa dalam percakapan antar bahasa dapat ditemukan potensi penyebab terjadinya kesalahpahaman dan miskomunikasi. Hal ini berawal dari adanya kendala bahasa yang disebabkan oleh ketidakmampuan berbicara atau memahami percakapan yang dilakukan, baik dari sumber maupun orang yang diajak bicara. Terlebih lagi, kondisi mereka saat berada di luar negeri, terutama di negara yang tidak menggunakan bahasa Inggris sebagai bahasa nasional, memaksa mereka harus berkomunikasi menggunakan bahasa setempat. Hal ini ditambah dengan penggunaan aksen dan dialek serta bahasa gaul atau bahasa gaul yang sering ditemukan dalam percakapan sehari-hari. Narasumber mengakui bahwa penggunaan alat bantu penerjemahan pada umumnya kurang mampu menangkap konteks percakapan seperti kondisi di atas. Oleh karena itu, diharapkan penelitian ini dapat membantu pengguna menerjemahkan percakapan secara langsung dan memahami konteks percakapan seperti di atas.

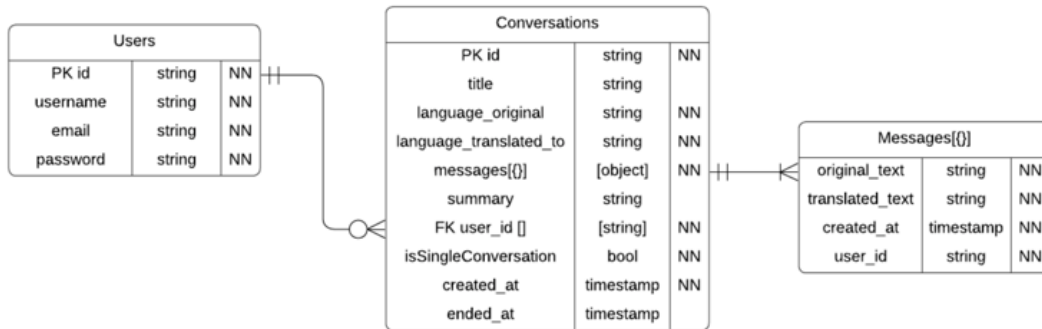
3.2 Desain Sistem

Perancangan sistem yang dibuat pada tahap ini didasarkan pada kebutuhan akan fitur-fitur yang dapat membantu pengguna dalam menyelesaikan permasalahan yang ada. Perancangan ini meliputi spesifikasi perangkat keras yang digunakan, aktivitas penggunaan sistem, arsitektur sistem, perancangan basis data, dan perancangan antarmuka pengguna. Penelitian ini menggunakan perangkat seluler berbasis iOS, sehingga memerlukan perangkat keras dengan spesifikasi minimal iOS 15. Aplikasi yang dikembangkan memungkinkan pengguna untuk melakukan percakapan antar bahasa dengan 3 fitur utama, yaitu percakapan satu perangkat, percakapan dua perangkat, dan riwayat percakapan. Berdasarkan fitur-fitur tersebut, diagram *use case* dari aplikasi yang akan dikembangkan ditunjukkan pada Gambar 2 di bawah ini.



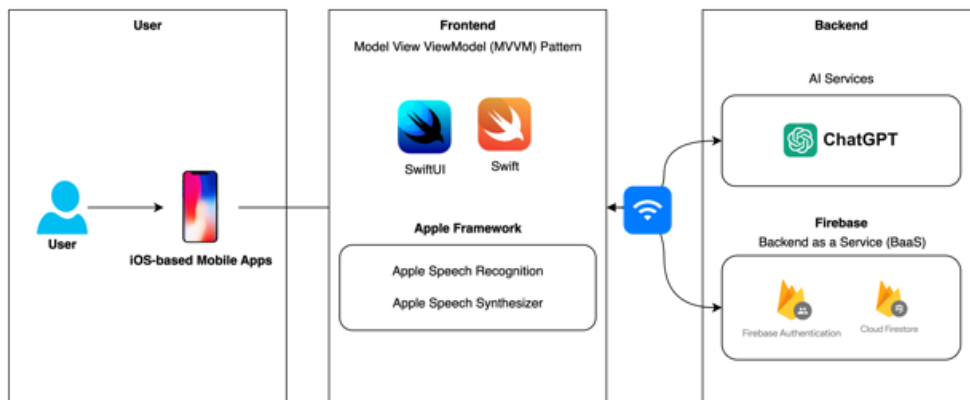
Gambar 2. Use case diagram dari sistem

Langkah selanjutnya, dibuatlah desain *database* dengan menggunakan *database* berbasis NoSQL dengan layanan dari Firebase. *Firebase Firestore* dipilih sebagai *database* utama karena keunggulannya dalam mendukung pembaruan secara *real-time*. Keunggulan ini dapat mendukung layanan penerjemahan yang lebih baik lagi. Desain *database* dapat dilihat pada Gambar 3 di bawah ini.



Gambar 3. Desain *database* dari sistem

Pengembangan aplikasi dalam penelitian ini menggunakan arsitektur *Model-View-ViewModel* (MVVM). Perancangan ini didasarkan pada arsitektur yang populer untuk pengembangan aplikasi berbasis *mobile*. Manfaat penggunaan arsitektur ini adalah dapat memecah tugas-tugas yang perlu dilakukan dalam satu *file*, sehingga tidak membebani kinerja dalam satu *file* dan memudahkan pengorganisasian kode sesuai dengan tanggung jawab masing-masing individu. Secara umum, pengembangan aplikasi akan didasarkan pada diagram arsitektur seperti yang ditunjukkan pada Gambar 4 di bawah ini.



Gambar 4. Arsitektur sistem

Dari visualisasi pada Gambar 4, pengguna akan mengakses aplikasi dari perangkat seluler mereka. Dalam sistem yang dibuat menggunakan bahasa pemrograman Swift dengan kerangka kerja SwiftUI sebagai basis antarmuka pengguna, teknologi dari Apple juga dimanfaatkan. Teknologi ini digunakan untuk fitur-fitur yang berhubungan dengan suara, seperti *Apple Speech Recognition* dan *Apple Speech Synthesizer*. Kemudian dengan menghubungkan perangkat ke internet, pengguna dapat memanfaatkan layanan penerjemahan, penyimpanan data, otentikasi, dan layanan ringkasan riwayat percakapan. Komunikasi antara sistem dan layanan di sisi backend menggunakan JSON untuk layanan ChatGPT API dan Firebase SDK untuk Firebase.

3.3 Implementasi

Tahap implementasi merupakan perwujudan dari apa yang telah dirancang pada tahap sebelumnya. Sesuai dengan desain, aplikasi diimplementasikan menggunakan bahasa pemrograman Swift yang mendukung pengembangan aplikasi *mobile* berbasis iOS. Selain itu, pada tahap ini juga akan dirancang tiga fitur utama. Ketiga fitur tersebut meliputi percakapan satu perangkat, percakapan dua perangkat, dan riwayat percakapan. Implementasi juga terpenuhi pada tampilan antarmuka pengguna. Tampilan antarmuka yang diimplementasikan juga sesuai dengan

desain dengan sedikit penyesuaian pada alur penggunaan.

3.4 Testing

Tahap pengujian ini merupakan tahap yang sangat penting, khususnya untuk pengembangan perangkat lunak. Tahap ini merupakan titik di mana sistem yang telah dikembangkan dapat diterima oleh calon pengguna atau tidak. Dalam penelitian ini, tahap pengujian dilakukan sebanyak dua kali. Pengujian pertama atau yang biasa disebut *alpha testing* dilakukan oleh tim pengembang aplikasi secara internal. Hal ini dilakukan untuk mengetahui apakah sistem yang dikembangkan memiliki *error* atau tidak. Selanjutnya, *beta testing* dilakukan kepada beberapa calon pengguna untuk mengetahui apakah sistem telah menemukan solusi dari permasalahan yang ada. Terakhir, dilakukan pula pengujian tambahan berupa *performance testing* yang dilakukan untuk melihat bagaimana hasil terjemahan yang dihasilkan oleh layanan penerjemah dari ChatGPT API.

Metode yang digunakan pada tahap *alpha testing* ini menggunakan metode *black box*. Pada tahap ini, tester melakukan uji coba untuk mengetahui apakah terdapat celah keamanan, *error*, atau *bug* pada seluruh fitur aplikasi. Mulai dari proses otentikasi pengguna seperti proses pembuatan akun hingga *logout* kembali. Keseluruhan proses yang dilakukan pada tahap ini berjumlah 21 skema pengujian.

Setelah tahap *alpha testing* berhasil, maka akan dilakukan tahap *beta testing* berikutnya. Tahap ini dilakukan untuk mendapatkan validasi terhadap solusi yang telah dirancang bagi calon pengguna. Pengguna akan diminta untuk mencoba menggunakan aplikasi guna membantu mereka melakukan percakapan antar bahasa. Kemudian akan diadakan sesi wawancara untuk mengetahui kepuasan mereka serta saran dan kritik yang dapat diberikan. Sesi wawancara dilakukan dengan metode semi terstruktur dengan panduan pertanyaan pada Tabel 1 yang berisi pertanyaan dalam tiga kategori yaitu kegunaan (*usability*), kemudahan penggunaan (*ease of use*), dan keberlanjutan penggunaan (*sustainable use*) dengan total delapan pertanyaan.

Tabel 1. Instrumen wawancara

Kegunaan	
P1	Bagaimana aplikasi dapat membantu Anda saat melakukan percakapan antarbahasa?
P2	Bagaimana fitur riwayat percakapan dan ringkasan membantu Anda memahami percakapan yang telah Anda lakukan?
Kemudahan penggunaan	
P3	Apakah aplikasi berjalan lancar tanpa bug, kesalahan, atau <i>crash</i> ?
P4	Bagaimana aplikasi merespons interaksi Anda? (misalnya: saat menekan tombol, memilih bahasa, mengisi kolom teks)
P5	Bagaimana hasil terjemahan dihasilkan oleh aplikasi?
P6	Bagaimana kecepatan terjemahan aplikasi?
P7	Seberapa mudah aplikasi digunakan secara keseluruhan dalam hal alur dan desain?
Keberlanjutan penggunaan	
P8	Apakah Anda akan menggunakan aplikasi ini saat melakukan percakapan dengan orang asing?

Hasil wawancara yang telah dilakukan akan diolah dan dianalisis untuk melihat kepuasan pengguna. Hasil ini juga dapat menunjukkan tingkat kesesuaian solusi yang ditawarkan oleh aplikasi yang telah dikembangkan terhadap permasalahan yang dialami oleh pengguna.

3.5 Deployment dan Maintenance

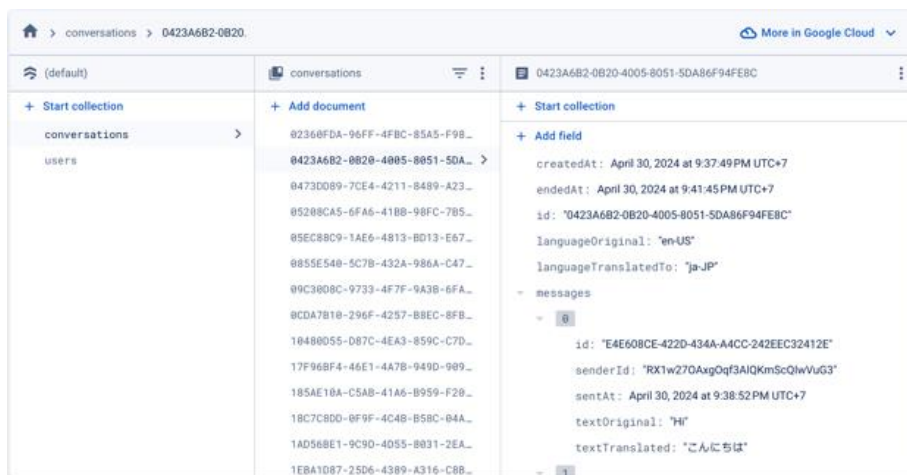
Tahap ini merupakan tahap akhir dari keseluruhan proses pengembangan perangkat lunak. Tahap ini dilakukan setelah hasil pengujian dirasa sesuai dengan yang diharapkan. Tahap ini difokuskan pada penerbitan aplikasi di pasaran agar dapat digunakan secara massal oleh pengguna.

4. Hasil dan Pembahasan

Bagian ini menjelaskan hasil implementasi yang telah dilakukan dalam mengembangkan aplikasi yang dirancang, meliputi *database* dan fitur-fiturnya. Selain itu, juga dijelaskan hasil pengujian yang telah dilakukan kepada calon pengguna.

4.1 Implementasi Database

Basis data yang digunakan dalam penelitian ini adalah basis data bertipe NoSQL, yaitu *Firebase Firestore*. Terdapat dua koleksi dokumen yang diimplementasikan dalam *database*, yaitu percakapan dan pengguna. Gambar 5 di bawah ini merupakan hasil implementasi basis data yang telah dilakukan.



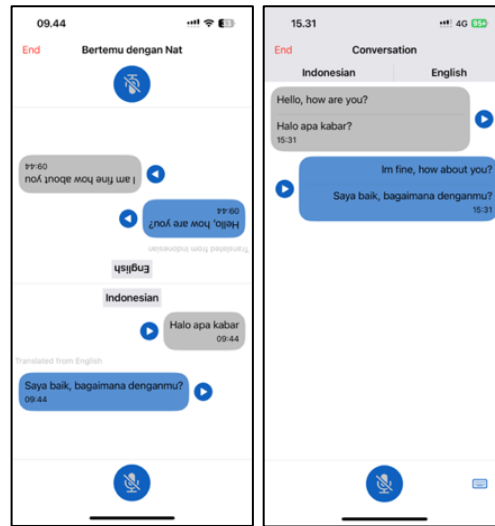
Gambar 5. Implementasi *database* pada *firebase firestore*

Gambar 5 di atas menunjukkan struktur data dalam *Firebase Firestore* yang terdiri dari dua koleksi utama: *users* dan *conversations*. Koleksi ini dirancang untuk menyimpan informasi riwayat percakapan secara *real-time* berdasarkan akun pengguna yang terotentikasi. Penggunaan struktur NoSQL memberikan fleksibilitas dalam menyimpan berbagai jenis data percakapan, termasuk metadata, waktu, dan ringkasan terjemahan. Desain ini mempermudah proses pengambilan data saat pengguna ingin meninjau kembali percakapan sebelumnya, serta mendukung performa sistem secara efisien.

4.2 Implementasi tampilan dan fitur

Antarmuka pengguna atau yang biasa dikenal dengan istilah *user interface* merupakan halaman-halaman yang berinteraksi dengan pengguna secara langsung. Implementasi antarmuka ini memanfaatkan *framework* modern untuk pengembangan aplikasi seluler berbasis iOS, *SwiftUI*. *SwiftUI* menggunakan bahasa pemrograman *Swift* sebagai basis pengembangan utamanya. Berikut ini adalah hasil implementasi fitur-fitur utama aplikasi yang dikembangkan.

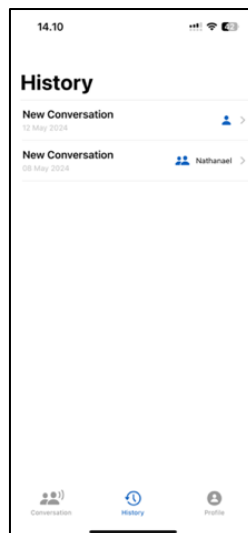
Salah satu fitur utama yang ditawarkan oleh aplikasi ini adalah fitur percakapan. Fitur percakapan sendiri memiliki dua jenis, yaitu percakapan dalam satu perangkat dan percakapan dalam dua perangkat. Fitur ini memungkinkan pengguna untuk melakukan percakapan secara langsung. Gambar 6 merupakan hasil implementasi halaman percakapan dengan bagian kiri untuk tampilan percakapan satu perangkat dan bagian kanan untuk tampilan percakapan dua perangkat.



Gambar 6. Implementasi fitur percakapan langsung

Gambar 6 di atas memperlihatkan dua skenario penggunaan aplikasi, yaitu percakapan dalam satu perangkat dan dua perangkat. Dalam percakapan satu perangkat, antarmuka dirancang agar pengguna dapat melihat *input* dan hasil terjemahan dalam satu layar, serta memfasilitasi komunikasi dengan cepat. Sementara pada mode dua perangkat, antarmuka terbagi sehingga dua pengguna yang dapat berinteraksi secara bergantian. Visualisasi ini penting untuk menunjukkan fleksibilitas aplikasi dalam beradaptasi dengan berbagai situasi percakapan nyata.

Fitur kedua yang diterapkan adalah fitur riwayat percakapan. Seperti yang dapat dilihat pada Gambar 7, terdapat daftar riwayat yang pernah dilakukan dengan indikator pembeda berdasarkan jenis percakapan yang dilakukan.

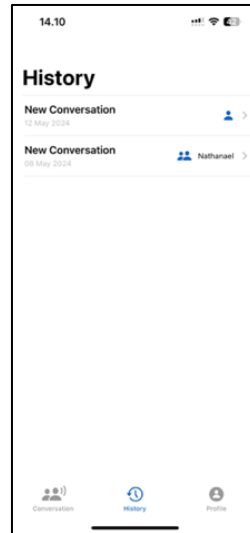


Gambar 7. Implementasi fitur riwayat percakapan

Gambar 7 di atas terlihat bahwa fitur *history* ini menampilkan semua percakapan yang pernah dilakukan oleh pengguna tersebut. Ketika pengguna memilih salah satu riwayat, maka pengguna akan diarahkan ke halaman detail

riwayat percakapan.

Fitur penting terakhir yang menjadi salah satu nilai dari penelitian ini adalah fitur ringkasan riwayat percakapan. Gambar 8 merupakan implementasi tampilan ringkasan riwayat percakapan.



Gambar 8. Implementasi fitur ringkasan percakapan

Gambar 8 di atas menampilkan fitur ringkasan percakapan yang dihasilkan oleh layanan ChatGPT API. Fitur ini memberikan inti sari dari percakapan dalam bentuk paragraf pendek yang memudahkan pengguna mengingat poin penting. Visualisasi ini memperjelas bagaimana aplikasi tidak hanya menerjemahkan, namun juga menyederhanakan informasi dalam konteks yang lebih praktis bagi pengguna. Selain itu, pengguna dapat menggunakan ringkasan ini untuk memudahkan pemahaman konteks dan meningkatkan efisiensi jika mereka ingin menemukan informasi spesifik dari percakapan yang telah mereka lakukan.

4.3 Hasil pengujian

Pengujian dilakukan dengan mengundang tiga orang penguji untuk mencoba menggunakan aplikasi secara langsung. Para penguji diminta untuk melakukan percakapan langsung. Setelah melakukan pengujian aplikasi, penguji akan diwawancarai berdasarkan beberapa kategori pertanyaan pada diskusi sebelumnya.

Berdasarkan hasil wawancara, dapat diketahui bahwa pengguna merasa aplikasi ini dapat membantu mereka untuk melakukan percakapan antar bahasa secara langsung dengan kendala yang minimal. Hasil terjemahannya pun cukup baik dengan konteks yang sesuai dengan percakapan yang dilakukan. Dari segi alur penggunaan dan desain aplikasi, pengguna merasa nyaman karena desainnya sederhana dan jelas. Namun, terdapat situasi di mana muncul kesalahan pada terjemahan bahasa Korea saat pengujian. Hal ini dikarenakan pustaka bahasa pada perangkat tidak dapat diidentifikasi yang sebenarnya dapat diatasi dengan melakukan *restart* pada perangkat pengguna.

5. Kesimpulan dan Saran

5.1 Kesimpulan

Berdasarkan studi pustaka yang telah dilakukan, ditemukan beberapa penelitian yang mendekati penelitian ini. Meskipun mengangkat permasalahan yang sama yaitu permasalahan bahasa, namun metode dan solusi yang dirancang berbeda. Penelitian yang dilakukan Alfari (2020) dan Ogundokun (2021) merupakan penelitian yang berfokus pada penyelesaian permasalahan bahasa literal dengan menggunakan teknologi deteksi karakter atau OCR. Sementara itu, Lengkong et al. (2022) menggunakan pendekatan lain yaitu berbasis audio seperti yang dilakukan

pada penelitian ini. Akan tetapi, hasil yang diperoleh pada penelitian Lengkong et al. (2022) kurang baik. Hal ini disebabkan karena penggunaan MIT App Inventor untuk aplikasi dalam mengembangkan aplikasi terutama untuk fitur berbasis audio kurang kompatibel dengan perangkat keras yang digunakan. Hal ini berbeda dengan hasil yang diperoleh pada penelitian ini. Penelitian ini berhasil mengembangkan aplikasi penerjemahan percakapan langsung berbasis iOS dengan menggunakan Swift dan SwiftUI, serta memanfaatkan teknologi Apple seperti *Apple Speech Recognition* dan *Synthesizer* untuk fitur *audio*. Aplikasi ini menggunakan layanan API dari ChatGPT untuk menerjemahkan dan meringkas riwayat percakapan, serta Firebase sebagai basis data untuk pembaruan data secara *real-time*.

Berdasarkan hasil yang diperoleh dalam pengujian dengan pengguna, aplikasi ini dinilai mampu menjadi solusi alat penerjemahan dalam kondisi percakapan langsung. Terbukti dari hasil pengujian yang menyatakan bahwa nilai kepuasan pengguna terhadap kegunaan aplikasi berada pada 86%. Nilai yang dihasilkan dari tahap pengujian ini menunjukkan bahwa pengguna merasa sangat terbantu dengan fitur-fitur yang ditawarkan dari aplikasi yang telah dihasilkan pada penelitian ini. Selain itu, nilai kemudahan penggunaan yang didapatkan dari aplikasi ini berada pada 75.2%. Hal ini mengindikasikan bahwa pengguna sudah merasa terbantu dengan alur penggunaan, tampilan, hasil terjemahan, serta kecepatan respons dari interaksi yang dilakukan oleh pengguna. Penilaian terhadap penggunaan keberlanjutan dari pengguna juga dinilai sangat baik, yaitu berada pada angka 88%. Akhirnya, dari wawancara yang telah dilakukan, diperoleh hasil yang baik dengan sedikitnya permasalahan yang timbul seperti yang telah dibahas pada bagian sebelumnya. Terlepas dari permasalahan tersebut, keseluruhan aplikasi dapat dijalankan dengan baik pada sesi pengujian pengguna. Performa yang ditawarkan juga cepat sehingga tidak mengurangi pengalaman pengguna dalam melakukan percakapan secara langsung sehingga pengguna tidak perlu takut akan kecanggungan atau munculnya kendala bahasa pada percakapan antar bahasa.

5.2 Saran

Berdasarkan masukan yang didapatkan dari pengujian yang telah dilakukan, terdapat beberapa saran yang dapat dimanfaatkan sebagai perbaikan aplikasi maupun pengembangan aplikasi penerjemah ke depannya. Dalam tahap pengujian, terdapat saran yang diajukan oleh penguji dan dapat dijadikan pertimbangan. Beberapa saran perbaikan yang diberikan ditujukan untuk dapat lebih mempermudah pengalaman pengguna seperti penambahan fitur penyimpanan *default* untuk pilihan bahasa pengguna. Hal ini dapat mencegah proses berulang yang dilakukan pengguna sehingga dapat mempercepat proses persiapan untuk melakukan percakapan. Selain itu, perbaikan juga dapat dilakukan pada respons dan penanganan *input* dari pengguna. Hal ini dapat meningkatkan kenyamanan dalam penggunaan aplikasi. Beberapa kondisi yang krusial untuk dapat ditambahkan fitur ini adalah pada deteksi suara di ruang percakapan dan status *loading* ketika melakukan interaksi dengan *backend* seperti ketika melakukan otentikasi, mengubah data percakapan, maupun meminta ringkasan percakapan.

Pengembangan untuk perluasan juga dapat dilakukan untuk meningkatkan fungsionalitas dari aplikasi. Terdapat saran dari tahap pengujian yang dapat dipertimbangkan sebagai perluasan dari aplikasi yaitu, fitur kontak untuk menyimpan akun pengguna lain yang sering diajak berbicara. Hal ini dapat mempercepat proses persiapan pengguna di beberapa kondisi tertentu. Kemudian, penguji juga menyarankan untuk penambahan fitur *quick chat* atau *quick translate*. Fitur ini diharapkan dapat membantu pengguna yang ingin mentranskripsikan audio dengan lebih cepat. Selain itu perluasan pengembangan juga dapat dilakukan dari segi platform dan layanan AI penerjemahnya. Platform yang digunakan bisa dilihat dari sisi potensi pengembangan versi *multi-platform* sehingga dapat mencakup lebih banyak pengguna. Layanan AI juga dapat menjadi pertimbangan pengembangan selanjutnya, mengingat perkembangan AI semakin pesat dan canggih di masa mendatang.

Pada akhirnya, penelitian ini memiliki beberapa keterbatasan yang perlu dicatat. Pertama, pengujian aplikasi masih terbatas pada jumlah peserta yang relatif kecil, sehingga generalisasi hasil kepuasan pengguna belum dapat mencerminkan populasi pengguna yang lebih luas. Kedua, pengujian fitur percakapan hanya dilakukan dalam kondisi lingkungan yang relatif terkendali, sehingga belum merepresentasikan berbagai situasi nyata seperti kebisingan tinggi atau koneksi internet yang tidak stabil. Ketiga, integrasi AI ChatGPT sebagai penerjemah belum mencakup personalisasi konteks pengguna secara lebih dalam.

Oleh karena itu, penelitian selanjutnya disarankan untuk melakukan pengujian dengan jumlah partisipan yang lebih besar dan bervariasi secara demografis. Selain itu, perlu dilakukan eksplorasi terhadap integrasi teknologi *noise reduction* dan *adaptive context learning* agar sistem penerjemahan menjadi lebih akurat dalam berbagai kondisi percakapan. Penelitian lanjutan juga dapat mengembangkan sistem dengan kemampuan *offline translation* untuk mengatasi kendala keterhubungan jaringan di lapangan.

Daftar Pustaka

- Alfarisi, R. E. (2020). Rancang bangun aplikasi terjemahan bahasa Jepang - Indonesia berbasis Android menggunakan Tesseract OCR. Retrieved February 1, 2024, from <http://elibrary.unikom.ac.id/id/eprint/3193>
- Al Shamsi, H., Almutairi, A. G., Al Mashrafi, S., & Al Kalbani, T. (2020). Implications of language barriers for healthcare: A systematic review. *Oman Medical Journal*, 35(2), 1–7. <https://doi.org/10.5001/OMJ.2020.40>
- Apple Developer Documentation. (t.t.-a). Speech | SFSpeechRecognizer. Diambil 16 Februari 2024, dari <https://developer.apple.com/documentation/speech/sfspeechrecognizer>
- Apple Developer Documentation. (t.t.-b). Speech synthesis. Diambil 16 Februari 2024, dari https://developer.apple.com/documentation/avfoundation/speech_synthesis
- Apple Inc. (t.t.-a). Swift | The powerful programming language that's also easy to learn. Diambil 16 Februari 2024, dari <https://developer.apple.com/swift/>
- Apple Inc. (t.t.-b). SwiftUI. Diambil 16 Februari 2024, dari <https://developer.apple.com/xcode/swiftui/>
- Buarqoub, I. A. S. (2019). Language barriers to effective communication. *Utopía y Praxis Latinoamericana*, 24(6), 64–77. Retrieved from <https://www.redalyc.org/journal/279/27962177008/html/>
- de Moissac, D., & Bowen, S. (2019). Impact of language barriers on quality of care and patient safety for official language minority Francophones in Canada. *Journal of Patient Experience*, 6(1), 24–32. <https://doi.org/10.1177/2374373518769008>
- Engel, M. M., & Agustinus, J. T. (2024). Real Time Online Exam Proctoring System in Higher Education Using WebRTC Technology. *Jurnal Teknik Informatika (Jutif)*, 4(6), 1575–1587. <https://doi.org/10.52436/1.jutif.2023.4.6.1564>
- Engel, M. M., Dinata, Y. M., & Antonio, T. (2023). Development of Cloud-Based Data Integration for Non-Profit Organization System. *International Journal of Review Management Business and Entrepreneurship (RMBE)*, 3(1), 1–10. <https://doi.org/10.37715/rmbe.v3i1.3990>
- Engel, M. M., & Lie, H. D. (2022). Library Self Service System Using NFC and 2FA Google Authenticator. *Jurnal Teknik Informatika (Jutif)*, 3(3), 753–761. <https://doi.org/10.20884/1.jutif.2022.3.3.345>
- Gupta, A. (2021). Comparative study of different SDLC models. *International Journal of Research in Applied Science and Engineering Technology*, 9(11), 73–80. <https://doi.org/10.22214/ijraset.2021.38736>
- Kothari, N., Jain, C., Soni, D., Kumar, A., Dadheech, A., & Sharma, H. (2024). Instant Language Translation App. *International Journal of Technical Research & Science*. 101-108. <https://doi.org/10.30780/specialissue-iset-2024/018>
- Lengiang, O., Mandias, G., & Tombeng, M. (2022). The implementation of Yandex engine on live translator application for Bahasa and English using block programming MIT App Inventor mobile based. *Cogito Smart Journal*, 8(1).
- Lordianto, R. L., & Engel, M. M. (2023). Rancang bangun aplikasi latihan public speaking berbasis iOS menggunakan teknologi live streaming video Agora.io. *Jurnal Informatika dan Sistem Informasi*, 9(1), 10–23. <https://doi.org/10.37715/juisi.v9i1.4325>
- Meesala, Y. V. N., Surya, V. S., Kiran, P. S., & Vardhan, R. S. (2024). Harnessing Open Innovation for Translating Global Languages into Indian Languages. *International Journal of Innovative Science and Research Technology*. 9(4), 1070-1076. <https://doi.org/10.38124/ijisrt/ijisrt24apr1907>

- Ogundokun, R. O., Awotunde, J. B., Misra, S., Segun-Owolabi, T., Adeniyi, E. A., & Jaglan, V. (2021). An android based language translator application. *Journal of Physics: Conference Series*. <https://doi.org/10.1088/1742-6596/1767/1/012032>.
- Pargaonkar, S. (2023). A comprehensive research analysis of Software Development Life Cycle (SDLC) Agile & Waterfall model advantages, disadvantages, and application suitability in software quality engineering. *International Journal of Scientific and Research Publications*, 13(8), 120–124. <https://doi.org/10.29322/ijsrp.13.08.2023.p14015>.
- Patil, A. A. & Dudhe, M. S. (2024). Language Translator App. *International Journal of Advanced Research in Science, Communication and Technology*. 4(5), 374-377. <https://doi.org/10.48175/ijarsct-17558>.
- Pratheeksha, M., Rai, P., & Vijetha, M. (2020). Language To Language Translation System. *International Journal of Scientific Research in Computer Science, Engineering and Information Technology*, 289–293. <https://doi.org/10.32628/cseit206363>.
- Razsiah F., Josi, A., & Mubaro, S. (2023). Aplikasi Penerjemah Bahasa Bangka Ke Bahasa Indonesia Menggunakan Neural Machine Translation Berbasis Website. *Jurnal Inovasi Teknologi Terapan*, 1(1), 68-76. <https://doi.org/10.33504/jitt.v1i1.67>.
- Reale, C. (2023). What are language barriers and how to overcome them. *KomunIKON*. Retrieved from <https://www.komunikon.com/en/notes/what-are-language-barriers>.
- StatCounter. (2024, Januari). Mobile Operating system market share worldwide | StatCounter Global Stats. <https://gs.statcounter.com/os-market-share/mobile/worldwide>.