

BAB III

ANALISIS DAN PERANCANGAN

3.1 Kerangka Berpikir

Kerangka berpikir menggambarkan alur konseptual penelitian yang menjelaskan setiap tahapan, dimulai dari proses identifikasi permasalahan hingga menghasilkan solusi berupa aplikasi penerjemah Bahasa Indonesia ke Bahasa Sunda berbasis Artificial Intelligeni *Hybrid AI Translation* dan teknologi *Retrieval-Based Voice Conversion* (RVC). Kerangka berpikir ini disusun berdasarkan permasalahan yang ditemukan, landasan teori yang digunakan, metode pengembangan sistem, serta implementasi teknologi yang diterapkan pada aplikasi LingoNusantara AI.

Penelitian ini diawali dengan identifikasi permasalahan mengenai semakin berkurangnya penggunaan Bahasa Sunda di kalangan masyarakat, khususnya generasi muda, serta masih terbatasnya aplikasi penerjemah yang mampu menghasilkan keluaran suara dengan karakteristik vokal pengguna. Sebagian besar aplikasi penerjemah yang tersedia hanya menghasilkan keluaran berupa teks atau menggunakan teknologi *Text-to-Speech* (TTS) konvensional sehingga suara yang dihasilkan masih terdengar sintetis dan belum mampu mempertahankan karakteristik maupun identitas vokal pengguna.

Berdasarkan permasalahan tersebut, penelitian ini melakukan studi literatur yang membahas *Artificial Intelligence* (AI), *Machine Learning*, *Deep Learning*, *Natural Language Processing* (NLP), *Machine Translation*, *Large Language Model* (LLM), *Flutter Framework*, *Retrieval-Based Voice Conversion* Framework, *Edge Text-to-Speech* (Edge TTS), *Google Gemini*, serta *Retrieval-Based Voice Conversion* (RVC). Hasil studi literatur digunakan sebagai dasar dalam merancang arsitektur sistem, menentukan metode pengembangan, serta memilih teknologi yang sesuai dengan kebutuhan penelitian.

Tahap berikutnya adalah pengumpulan data yang terdiri atas dataset Bahasa Indonesia–Bahasa Sunda sebagai basis pengetahuan *AI Translation Engine* dan dataset suara pengguna yang digunakan untuk melatih *model Retrieval-Based Voice Conversion* (RVC). Dataset suara kemudian diproses menggunakan *platform Applio* sehingga menghasilkan *model* suara berupa berkas .pth dan .index yang digunakan pada proses inferensi suara.

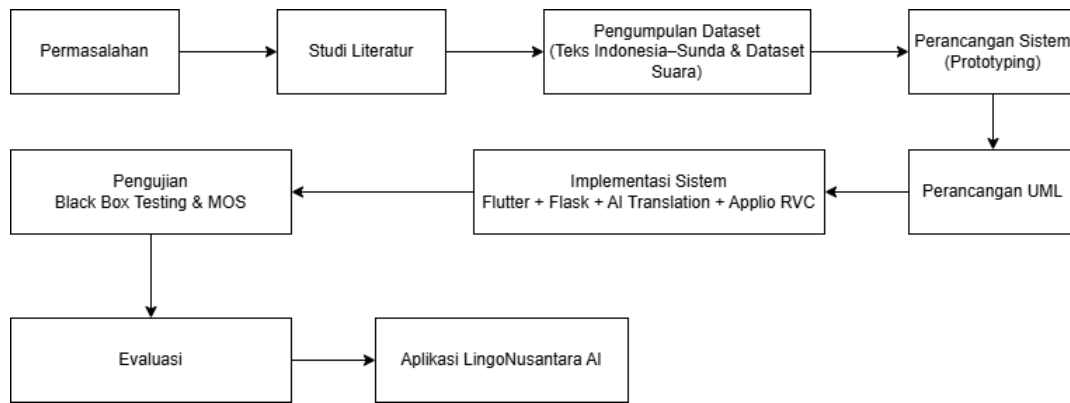
Setelah seluruh kebutuhan sistem diperoleh, dilakukan proses perancangan aplikasi menggunakan metode Prototyping. Tahapan tersebut mencakup analisis kebutuhan sistem, penyusunan arsitektur sistem, desain antarmuka pengguna (*User Interface*), penyusunan

Unified *Modeling Language* (UML), desain basis data, serta perancangan komunikasi antara aplikasi *Flutter* dengan backelask melalui REST API.

Implementasi sistem dilakukan dengan membangun aplikasi *mobile* menggunakan *Flutter Framework* sebagai *frontend*, sedangkan *Retrieval-Based Voice Conversion Framework* berbasis Python digunakan sebagai *backend*. Backend bertugas menerima permintaan dari aplikasi Android, kemudian menentukan mekanisme penerjemahan menggunakan konsep *Hybrid AI Translation*. Apabila perangkat terhubung ke internet, sistem memanfaatkan *Google Gemini* sebagai layanan *Large Language Model* (LLM) untuk menghasilkan terjemahan yang lebih alami dan kontekstual. Sebaliknya, apabila koneksi internet tidak tersedia atau layanan Gemini tidak dapat diakses, sistem secara otomatis menggunakan *AI Translation Engine* berbasis dataset lokal sehingga proses penerjemahan tetap dapat dilakukan.

Hasil terjemahan Bahasa Sunda selanjutnya diproses menggunakan *Edge Text-to-Speech* (Edge TTS) untuk menghasilkan suara dasar. Audio tersebut kemudian dikonversi menggunakan *Retrieval-Based Voice Conversion* (RVC) melalui *platform Applio* sehingga menghasilkan keluaran suara yang memiliki karakteristik menyerupai *model* suara hasil pelatihan. Selanjutnya, teks hasil terjemahan beserta file audio dikirim kembali ke aplikasi *Flutter* agar dapat ditampilkan kepada pengguna, disimpan ke dalam riwayat penerjemahan, serta dapat diputar maupun diunduh melalui aplikasi.

Tahap akhir penelitian dilakukan dengan melaksanakan pengujian sistem menggunakan metode *Black Box Testing* guna memastikan seluruh fungsi aplikasi telah berjalan sesuai kebutuhan fungsional. Selain pengujian fungsional, kualitas suara juga dievaluasi menggunakan metode *Mean Opinion Score* (MOS) untuk mengetahui tingkat kejelasan, naturalitas, kenyamanan, dan kemiripan suara hasil konversi berdasarkan penilaian responden. Hasil pengujian tersebut menjadi dasar evaluasi dalam proses penyempurnaan aplikasi sehingga dihasilkan sistem penerjemah Bahasa Indonesia ke Bahasa Sunda yang lebih akurat, adaptif, mudah digunakan, serta mampu mempertahankan karakteristik vokal pengguna.



Gambar 3. 1 Kerangka Berpikir

3.2 Metodologi Penelitian

Penelitian ini menggunakan metode Prototyping sebagai metode pengembangan perangkat lunak. Metode ini dipilih karena mampu mendukung proses pengembangan sistem secara bertahap melalui pembuatan prototype, evaluasi, serta penyempurnaan sistem berdasarkan hasil pengujian. Menurut Pressman dan Maxim (2020), metode Prototyping memungkinkan pengembang memperoleh umpan balik (*feedback*) sejak tahap awal sehingga perubahan kebutuhan sistem dapat diakomodasi tanpa harus mengulang seluruh proses pengembangan.

Metode Prototyping dipilih karena pengembangan aplikasi LingoNusantara AI dilakukan melalui beberapa tahapan penyempurnaan, mulai dari implementasi *AI Translation Engine* berbasis dataset lokal, integrasi *Google Gemini* melalui mekanisme *Hybrid AI Translation*, implementasi *Edge Text-to-Speech* (Edge TTS), hingga penerapan *Retrieval-Based Voice Conversion* (RVC) menggunakan *platform Applio*. Setiap tahapan dievaluasi secara bertahap sebelum dilanjutkan ke tahap berikutnya sehingga aplikasi yang dihasilkan sesuai dengan kebutuhan pengguna dan tujuan penelitian.

Tahapan metode Prototyping yang digunakan pada penelitian ini terdiri atas pengumpulan kebutuhan (*requirement gathering*), perancangan prototype (*quick design*), pembangunan prototype (*build prototype*), evaluasi prototype (*user evaluation*), serta implementasi dan penyempurnaan sistem (*refinement and implementation*).

Tahapan penelitian yang dilakukan adalah sebagai berikut.

1. Identifikasi Masalah

Tahap pertama dilakukan dengan mengidentifikasi berbagai permasalahan yang melatarbelakangi penelitian. Permasalahan utama yang ditemukan yaitu semakin berkurangnya penggunaan Bahasa Sunda sebagai bahasa daerah dalam kehidupan sehari-hari, terutama pada

kalangan generasi muda. Selain itu, sebagian besar aplikasi penerjemah yang tersedia saat ini hanya mampu menghasilkan keluaran berupa teks atau suara sintetis menggunakan teknologi *Text-to-Speech* (TTS) sehingga belum mampu mempertahankan karakteristik suara asli pengguna.

2. Studi Literatur

Tahap studi literatur dilakukan dengan mengumpulkan berbagai referensi ilmiah berupa buku, jurnal nasional, jurnal internasional, prosiding, dan dokumentasi resmi yang berkaitan dengan topik penelitian. Literatur yang dipelajari meliputi *Artificial Intelligence* (AI), *Machine Learning*, *Deep Learning*, *Natural Language Processing* (NLP), *Machine Translation*, *Flutter Framework*, *Retrieval-Based Voice Conversion Framework*, *Retrieval-Based Voice Conversion* (RVC), *Applio*, *Mean Opinion Score* (MOS), Black Box Testing, serta metode Prototyping. Hasil studi literatur digunakan sebagai landasan teoritis dalam proses analisis, perancangan, implementasi, dan pengujian sistem.

3. Pengumpulan Data

Tahap pengumpulan data dilakukan untuk memperoleh data yang dibutuhkan dalam proses pengembangan sistem. Data yang digunakan pada penelitian ini terdiri atas dataset penerjemahan bahasa dan dataset suara pengguna.

1. Dataset Bahasa Indonesia–Bahasa Sunda

Dataset dikumpulkan dari berbagai sumber referensi pembelajaran Bahasa Sunda, kamus dwibahasa Indonesia–Sunda, buku tata bahasa Sunda, serta contoh percakapan sehari-hari yang disesuaikan dengan kaidah kebahasaan Sunda. Dataset yang telah diperoleh kemudian melalui proses validasi dan penyusunan agar menghasilkan pasangan kalimat yang sesuai dengan kebutuhan sistem. Selain berasal dari sumber referensi kebahasaan, dataset juga dikembangkan secara bertahap melalui mekanisme AI-assisted dataset building. Pada mekanisme ini, hasil terjemahan yang dihasilkan oleh Gemini AI ketika aplikasi digunakan secara *online* dapat dijadikan sebagai referensi tambahan untuk memperkaya pasangan kalimat Bahasa Indonesia–Bahasa Sunda. Seluruh data yang akan ditambahkan ke dalam dataset tetap melalui proses validasi sehingga hanya kosakata dan struktur kalimat yang sesuai dengan kaidah Bahasa Sunda Priangan yang digunakan pada *AI Translation Engine*.

2. Dataset Suara

Dataset suara diperoleh melalui proses perekaman suara secara mandiri menggunakan mikrofon dengan format .wav. Rekaman dilakukan pada lingkungan yang minim gangguan suara sehingga menghasilkan kualitas audio yang baik sebagai data pelatihan *model Retrieval-Based Voice Conversion (RVC)*.

3. Dataset *Model RVC*

Dataset suara yang telah diperoleh selanjutnya diproses menggunakan *platform Applio* sehingga menghasilkan *model* suara berupa file .pth dan .index yang digunakan pada proses inferensi suara.

Tabel 3. 1 Dataset Penelitian

No	Jenis Dataset	Format	Fungsi
1	Dataset Bahasa Indonesia–Bahasa Sunda	JSON	Sebagai basis data penerjemahan bahasa
2	Dataset Suara Pengguna	WAV	Sebagai data pelatihan <i>model RVC</i>
3	<i>Model RVC</i>	PTH dan INDEX	Digunakan pada proses inferensi suara menggunakan <i>Applio</i>

4. Analisis Kebutuhan Sistem

Tahap analisis kebutuhan dilakukan untuk menentukan kebutuhan fungsional maupun nonfungsional sistem yang akan dikembangkan. Analisis ini bertujuan agar aplikasi yang dibangun mampu memenuhi kebutuhan pengguna dalam melakukan penerjemahan Bahasa Indonesia ke Bahasa Sunda beserta konversi suara menggunakan teknologi *Retrieval-Based Voice Conversion (RVC)*. Analisis kebutuhan meliputi kebutuhan perangkat keras, perangkat lunak, kebutuhan pengguna, kebutuhan sistem, serta komunikasi antara aplikasi *Flutter* dengan backend *Retrieval-Based Voice Conversion* melalui REST API.

5. Perancangan Sistem

Tahap perancangan dilakukan berdasarkan hasil analisis kebutuhan yang telah diperoleh sebelumnya. Perancangan sistem meliputi:

1. Perancangan arsitektur sistem.
2. Perancangan komunikasi REST API.

3. Perancangan Unified *Modeling Language* (UML).
4. Perancangan basis data sederhana.
5. Perancangan antarmuka pengguna (*User Interface*) aplikasi *Flutter*.

6. Implementasi Sistem

Tahap implementasi merupakan proses pembangunan aplikasi sesuai hasil perancangan. Sistem dikembangkan menggunakan beberapa teknologi, yaitu:

1. *Flutter* sebagai *frontend* aplikasi Android.
2. Python dan *Retrieval-Based Voice Conversion* sebagai backend REST API.
3. *Hybrid AI Translation* yang menggabungkan *AI Translation Engine* berbasis dataset lokal dengan *Google Gemini* sebagai layanan *Large Language Model* (LLM) ketika perangkat terhubung ke internet.
4. *Applio* sebagai media inferensi *Retrieval-Based Voice Conversion* (RVC).
5. Audioplayers sebagai pemutar audio hasil konversi suara pada aplikasi *Flutter*.

7. Pengujian Sistem

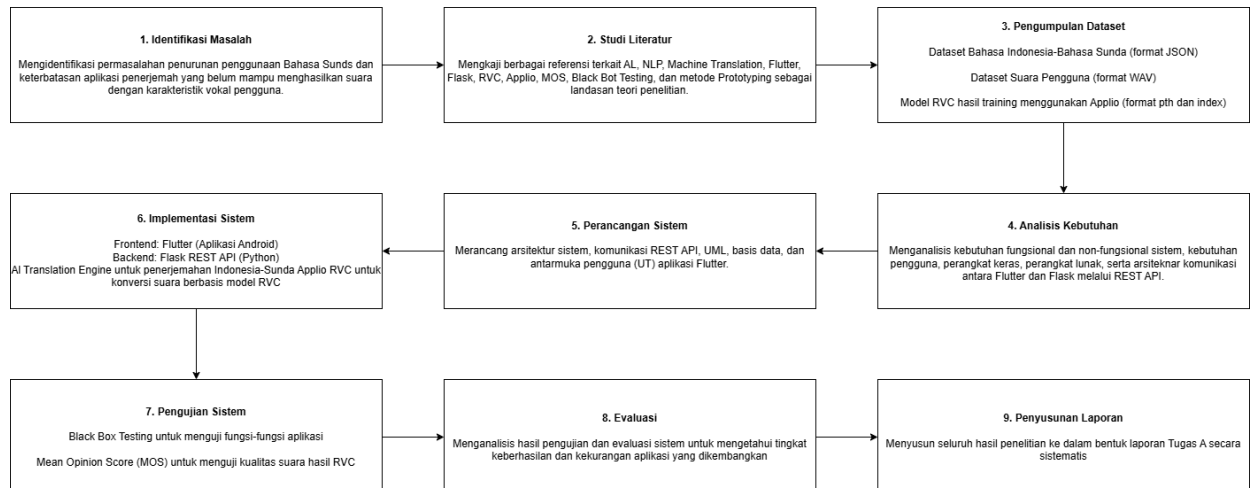
Tahap pengujian dilakukan setelah seluruh fitur berhasil diimplementasikan. Pengujian bertujuan memastikan bahwa seluruh fungsi aplikasi berjalan sesuai kebutuhan.

Pengujian dilakukan menggunakan dua metode, yaitu:

1. *Black Box Testing* untuk menguji seluruh fungsi aplikasi.
2. *Mean Opinion Score* (MOS) untuk mengevaluasi kualitas suara hasil konversi berdasarkan penilaian responden.

8. Penyusunan Laporan

Tahap terakhir adalah menyusun seluruh hasil penelitian ke dalam bentuk laporan Tugas Akhir yang terdiri atas proses analisis, perancangan, implementasi, pengujian, pembahasan, hingga kesimpulan penelitian.



Gambar 3. 2 Diagram Alur Metodologi

3.2.1 Teknik Pengumpulan Data

Teknik pengumpulan data merupakan tahapan yang dilakukan untuk memperoleh berbagai informasi dan data yang dibutuhkan dalam proses pengembangan aplikasi LingoNusantara AI. Data yang dikumpulkan meliputi referensi ilmiah sebagai landasan teori, dataset Bahasa Indonesia–Bahasa Sunda sebagai basis pengetahuan *AI Translation Engine*, dataset suara pengguna untuk pelatihan *model Retrieval-Based Voice Conversion (RVC)*, serta data hasil observasi implementasi sistem. Seluruh data tersebut digunakan sebagai dasar dalam proses analisis, perancangan, implementasi, dan pengujian aplikasi. Teknik pengumpulan data pada penelitian ini meliputi studi literatur, pengumpulan dataset bahasa, pengumpulan dataset suara, pembuatan *model RVC*, serta observasi terhadap implementasi sistem.

1. Studi Literatur

Studi literatur dilakukan dengan mengumpulkan berbagai referensi ilmiah yang berkaitan dengan *Artificial Intelligence (AI)*, *Machine Learning*, *Deep Learning*, *Natural Language Processing (NLP)*, *Machine Translation*, *Flutter Framework*, *Retrieval-Based Voice*

Conversion Framework, *Retrieval-Based Voice Conversion (RVC)*, *Applio*, *Mean Opinion Score (MOS)*, *Black Box Testing*, serta metode *Prototyping*. Referensi diperoleh dari buku, jurnal nasional, jurnal internasional, prosiding ilmiah, dan dokumentasi resmi teknologi yang digunakan. Menurut Creswell (2023), studi literatur bertujuan membangun landasan teoritis yang kuat sehingga penelitian memiliki dasar ilmiah yang jelas.

Selain mempelajari teori mengenai *Artificial Intelligence*, *Natural Language Processing (NLP)*, *Machine Translation*, *Flutter*, *Retrieval-Based Voice Conversion*, dan *Retrieval-Based Voice Conversion (RVC)*, studi literatur juga dilakukan terhadap perkembangan *Large Language Model (LLM)*, khususnya *Google Gemini*, sebagai dasar penerapan mekanisme *Hybrid AI Translation* pada aplikasi LingoNusantara AI. Kajian tersebut digunakan sebagai landasan dalam merancang integrasi antara *AI Translation Engine* berbasis dataset lokal dengan layanan penerjemahan berbasis *cloud*.

Hasil studi literatur digunakan sebagai acuan dalam menentukan arsitektur sistem, metode pengembangan, teknologi yang digunakan, serta teknik pengujian aplikasi LingoNusantara AI.

2. Pengumpulan Dataset Bahasa

Dataset bahasa digunakan sebagai basis pengetahuan *AI Translation Engine* dalam proses penerjemahan Bahasa Indonesia ke Bahasa Sunda. Dataset disusun dalam bentuk pasangan kalimat yang terdiri atas kalimat berbahasa Indonesia beserta hasil terjemahannya dalam Bahasa Sunda. Seluruh data disimpan dalam format JSON (JavaScript Object Notation) sehingga mudah diproses oleh sistem pada saat proses penerjemahan dilakukan.

Dataset dikumpulkan dari berbagai sumber, antara lain kamus dwibahasa Indonesia–Sunda, buku tata bahasa Sunda, referensi pembelajaran Bahasa Sunda, serta contoh percakapan sehari-hari yang disesuaikan dengan kaidah kebahasaan Sunda. Seluruh data yang diperoleh kemudian melalui proses validasi, normalisasi, dan penyusunan untuk menghasilkan pasangan kalimat yang konsisten serta sesuai dengan kebutuhan sistem penerjemahan.

Pada penelitian ini, dataset bahasa digunakan sebagai sumber pengetahuan utama bagi *AI Translation Engine* ketika sistem berjalan tanpa koneksi internet. Selain itu, sistem juga mengintegrasikan *Google Gemini* sebagai layanan *Large Language Model (LLM)* melalui mekanisme *Hybrid AI Translation*. Ketika perangkat terhubung ke internet, backend dapat menggunakan *Google Gemini* untuk menghasilkan terjemahan yang lebih alami dan kontekstual. Sebaliknya, apabila layanan Gemini tidak tersedia atau perangkat berada dalam kondisi *offline*, sistem secara otomatis menggunakan *AI Translation Engine* berbasis dataset lokal sehingga proses penerjemahan tetap dapat dilakukan.

Contoh struktur dataset yang digunakan ditunjukkan pada Tabel 3.2.

Tabel 3. 2 Contoh Struktur Dataset Bahasa

Bahasa Indonesia	Bahasa Sunda
Selamat pagi	Wilujeng enjing
Apa kabar?	Kumaha damang?
Terima kasih	Hatur nuhun
Sampai jumpa	Dugi ka tepang deui

Dataset ini digunakan sebagai basis pengetahuan pada *AI Translation Engine* untuk menghasilkan terjemahan secara otomatis.

3. Pengumpulan Dataset Suara (Dataset Vokal)

Dataset suara digunakan sebagai data pelatihan *model Retrieval-Based Voice Conversion* (RVC) untuk menghasilkan *model* suara yang mampu mempertahankan karakteristik vokal pengguna. Dataset diperoleh melalui proses perekaman suara secara mandiri menggunakan mikrofon dengan kualitas yang memadai. Seluruh rekaman disimpan dalam format WAV dengan spesifikasi yang sesuai untuk proses pelatihan *model* RVC.

Perekaman dilakukan pada lingkungan yang memiliki tingkat kebisingan rendah untuk meminimalkan gangguan suara (*noise*). Dataset yang telah diperoleh kemudian melalui tahap preprocessing yang meliputi pemeriksaan kualitas audio, normalisasi volume, pengurangan *noise*, pemotongan audio (*audio slicing*), serta validasi format audio sehingga seluruh data memiliki kualitas yang seragam sebelum digunakan pada proses pelatihan *model*.

Model pelatihan suara dibangun menggunakan *platform Applio* yang mengimplementasikan teknologi *Retrieval-Based Voice Conversion* (RVC). Hasil pelatihan menghasilkan *model* berupa berkas .pth dan .index yang selanjutnya digunakan pada proses inferensi untuk mengubah suara hasil *Edge Text-to-Speech* (Edge TTS) menjadi suara yang memiliki karakteristik menyerupai suara pengguna.

Tabel 3. 3 Spesifikasi Dataset Suara

Komponen	Spesifikasi
Format Audio	WAV
Channel	Mono
Sample Rate	40 kHz atau 48 kHz
Durasi Dataset	$\pm 10\text{--}30$ menit
Bahasa	Bahasa Indonesia
Lingkungan Rekaman	Minim kebisingan (noise)

Sebelum digunakan pada proses pelatihan *model*, seluruh data suara menjalani tahap audio preprocessing, meliputi proses noise reduction, normalisasi volume, dan audio slicing menjadi beberapa segmen berdurasi 3–10 detik. Tahapan ini bertujuan meningkatkan kualitas data sehingga proses ekstraksi fitur suara dapat dilakukan secara optimal.

4. Pembuatan *Model Retrieval-Based Voice Conversion (RVC)*

Dataset suara yang telah dipersiapkan kemudian digunakan untuk melatih *model Retrieval-Based Voice Conversion (RVC)* menggunakan *platform Applio*. Proses pelatihan menghasilkan *model* berupa file .pth sebagai *model* utama dan file .index sebagai indeks pencarian fitur suara. *Model* yang dihasilkan selanjutnya digunakan pada backend *Retrieval-Based Voice Conversion* untuk melakukan proses inferensi suara sehingga keluaran audio hasil terjemahan memiliki karakteristik yang menyerupai suara asli pengguna.

5. Observasi dan Pengujian Sistem

Setelah seluruh dataset dan *model* berhasil dipersiapkan, dilakukan observasi terhadap jalannya sistem secara menyeluruh. Pengujian dilakukan mulai dari proses input teks Bahasa Indonesia, pengiriman data ke backend *Retrieval-Based Voice Conversion* melalui REST API, proses penerjemahan menggunakan *AI Translation Engine*, proses inferensi menggunakan *model RVC* pada *platform Applio*, hingga sistem menghasilkan keluaran berupa teks Bahasa Sunda dan file audio yang dapat diputar melalui aplikasi *Flutter*. Data hasil pengujian digunakan sebagai dasar evaluasi untuk mengetahui apakah seluruh fungsi aplikasi telah berjalan sesuai dengan kebutuhan penelitian.

3.2.2 Metode Pengujian

Metode pengujian merupakan tahapan yang dilakukan untuk memastikan bahwa aplikasi LingoNusantara AI telah berjalan sesuai dengan kebutuhan sistem yang telah dirancang. Pengujian dilakukan terhadap seluruh fungsi utama aplikasi, mulai dari proses penerjemahan Bahasa Indonesia ke Bahasa Sunda menggunakan mekanisme *Hybrid AI Translation*, komunikasi antara aplikasi *Flutter* dengan *Retrieval-Based Voice Conversion* melalui REST API, hingga proses sintesis suara menggunakan *Edge Text-to-Speech* (Edge TTS) dan konversi suara menggunakan *Retrieval-Based Voice Conversion* (RVC). Selain menguji fungsionalitas sistem, penelitian ini juga mengevaluasi kualitas suara hasil konversi berdasarkan persepsi pengguna menggunakan metode *Mean Opinion Score* (MOS).

1. *Black Box Testing*

Black Box Testing merupakan metode pengujian perangkat lunak yang berfokus pada pengujian fungsi sistem tanpa memperhatikan struktur kode program di dalamnya. Menurut Pressman dan Maxim (2020), *Black Box Testing* bertujuan untuk memastikan bahwa setiap fungsi pada sistem menghasilkan keluaran yang sesuai dengan kebutuhan pengguna berdasarkan masukan yang diberikan. Pada penelitian ini, *Black Box Testing* dilakukan untuk menguji seluruh fitur utama aplikasi LingoNusantara AI, antara lain:

1. Input teks Bahasa Indonesia.
2. Proses penerjemahan Bahasa Indonesia ke Bahasa Sunda menggunakan *Hybrid AI Translation*.
3. Proses perekaman suara.
4. Proses pemilihan file audio.
5. Komunikasi REST API antara *Flutter* dan *Retrieval-Based Voice Conversion*.
6. Proses inferensi menggunakan *model Retrieval-Based Voice Conversion* (RVC).
7. Pemutaran audio hasil konversi.
8. Penyimpanan hasil audio pada perangkat.

Hasil pengujian dinyatakan berhasil apabila seluruh fungsi dapat berjalan sesuai dengan spesifikasi yang telah dirancang.

Tabel 3. 4 Rencana Pengujian Black Box

No	Fitur yang Diuji	Masukan	Hasil yang Diharapkan
1	Input Teks	Teks Bahasa Indonesia	Teks berhasil diterima sistem
2	Penerjemahan Bahasa	Teks Bahasa Indonesia	Sistem menampilkan hasil terjemahan Bahasa Sunda
3	Perekaman Suara	Rekam suara pengguna	File audio berhasil dibuat
4	Unggah Audio	File audio (.wav)	Audio berhasil dikirim ke backend
5	REST API	Permintaan HTTP POST	Backend menerima data dengan benar
6	<i>Hybrid AI Translation</i>	Data teks Bahasa Indonesia	Sistem berhasil menerjemahkan menggunakan <i>Google Gemini</i> saat <i>online</i> atau <i>AI Translation Engine</i> saat <i>offline</i>
7	<i>Retrieval-Based Voice Conversion</i>	Audio masukan	Audio berhasil dikonversi
8	Pemutaran Audio	File audio hasil	Audio dapat diputar melalui aplikasi
9	Unduh Audio	File hasil konversi	Audio berhasil disimpan pada perangkat

2. Mean Opinion Score (MOS)

Pengujian kualitas suara pada penelitian ini dilakukan menggunakan metode *Mean Opinion Score* (MOS). Metode ini digunakan untuk mengevaluasi kualitas suara hasil *Retrieval-Based Voice Conversion* (RVC) berdasarkan penilaian subjektif dari responden. Penilaian dilakukan dengan meminta responden mendengarkan hasil audio konversi yang dihasilkan oleh aplikasi LingoNusantara AI, kemudian memberikan penilaian terhadap beberapa aspek, yaitu kejelasan suara (speech clarity), kemiripan suara dengan suara asli (speaker similarity), kealamian suara (naturalness), dan kenyamanan saat didengarkan (listening comfort).

Nilai yang diberikan oleh seluruh responden selanjutnya dihitung menggunakan metode *Mean Opinion Score* (MOS) untuk memperoleh nilai rata-rata kualitas suara yang dihasilkan oleh sistem. Hasil pengujian MOS digunakan untuk mengetahui tingkat penerimaan pengguna terhadap kualitas suara hasil konversi serta menjadi dasar dalam mengevaluasi performa *model Retrieval-Based Voice Conversion* (RVC). Pengujian MOS dilakukan

terhadap empat aspek penilaian, yaitu kejelasan suara (speech clarity), kemiripan suara (speaker similarity), kealamian suara (naturalness), dan kenyamanan saat didengarkan (listening comfort) sehingga evaluasi kualitas suara dilakukan secara lebih komprehensif. yang diterapkan pada aplikasi.

Adapun teori mengenai metode *Mean Opinion Score* (MOS), skala penilaian, serta rumus perhitungannya telah dijelaskan pada Bab II, sedangkan hasil pengujian, proses perhitungan, dan analisis nilai MOS akan disajikan secara rinci pada Bab IV.

3. Evaluasi Hasil Pengujian

Tahap evaluasi dilakukan dengan menganalisis hasil *Black Box Testing* dan *Mean Opinion Score* (MOS). Hasil *Black Box Testing* digunakan untuk mengetahui apakah seluruh fitur aplikasi telah berjalan sesuai dengan kebutuhan sistem, sedangkan hasil MOS digunakan untuk mengukur tingkat kualitas dan naturalitas suara hasil konversi.

Apabila selama proses pengujian masih ditemukan kesalahan fungsi atau kualitas suara yang belum optimal, maka dilakukan proses perbaikan pada sistem sesuai dengan tahapan metode Prototyping hingga aplikasi memenuhi kebutuhan penelitian.

3.3 Metode Pengembangan Sistem

Metode pengembangan sistem merupakan tahapan yang digunakan sebagai pedoman dalam membangun aplikasi LingoNusantara AI. Penelitian ini menggunakan metode Prototyping karena metode tersebut memungkinkan pengembangan sistem dilakukan secara bertahap melalui proses evaluasi dan penyempurnaan berdasarkan kebutuhan pengguna. Menurut Pressman dan Maxim (2020), metode Prototyping merupakan pendekatan pengembangan perangkat lunak yang menghasilkan *model* awal (*prototype*) untuk memperoleh umpan balik (*feedback*) dari pengguna sebelum sistem dikembangkan secara menyeluruh. Pendekatan ini sangat sesuai diterapkan pada pengembangan aplikasi yang melibatkan integrasi berbagai teknologi seperti *Artificial Intelligence* (AI), *Flutter Framework*, *Retrieval-Based Voice Conversion Framework*, *Google Gemini*, *Edge Text-to-Speech* (Edge TTS), dan *Retrieval-Based Voice Conversion* (RVC).

Menurut Sommerville (2021), metode Prototyping memberikan fleksibilitas dalam proses pengembangan karena memungkinkan perubahan kebutuhan sistem dilakukan pada setiap tahapan pengembangan. Melalui proses iteratif, pengembang dapat memperbaiki fungsi

maupun antarmuka aplikasi berdasarkan hasil evaluasi sehingga sistem yang dihasilkan lebih sesuai dengan kebutuhan pengguna.

Pada penelitian ini, metode Prototyping diterapkan karena proses pengembangan aplikasi LingoNusantara AI melibatkan beberapa komponen yang saling terintegrasi, yaitu *AI Translation Engine* berbasis dataset lokal, *Google Gemini* melalui mekanisme *Hybrid AI Translation*, *Flutter Framework* sebagai *frontend*, *Flask Framework* sebagai *backend* REST API, *Edge Text-to-Speech* (Edge TTS), serta *Retrieval-Based Voice Conversion* (RVC) menggunakan *platform Applio*. Setiap komponen dikembangkan secara bertahap dan dievaluasi secara berulang hingga menghasilkan aplikasi yang mampu menerjemahkan Bahasa Indonesia ke Bahasa Sunda serta menghasilkan keluaran suara yang mempertahankan karakteristik vokal pengguna.

Tahapan metode Prototyping yang diterapkan pada penelitian ini terdiri atas beberapa tahap sebagai berikut

.

1. Pengumpulan Kebutuhan (*Requirement Gathering*)

Tahap pertama dilakukan dengan mengidentifikasi kebutuhan sistem berdasarkan hasil studi literatur, pengumpulan data, serta analisis kebutuhan pengguna. Pada tahap ini ditentukan kebutuhan fungsional dan nonfungsional aplikasi, seperti kemampuan menerjemahkan Bahasa Indonesia ke Bahasa Sunda, memilih ragam Bahasa Sunda, melakukan perekaman suara, mengunggah file audio, memproses konversi suara menggunakan *Retrieval-Based Voice Conversion* (RVC), serta menampilkan hasil terjemahan dalam bentuk teks dan audio.

2. Perancangan *Prototype* (*Quick Design*)

Setelah kebutuhan sistem diperoleh, dilakukan perancangan *prototype* sebagai gambaran awal aplikasi. Perancangan meliputi penyusunan arsitektur sistem, perancangan komunikasi antara aplikasi *Flutter* dengan backend *Retrieval-Based Voice Conversion* melalui REST API, penyusunan *Unified Modeling Language* (UML), perancangan basis data, serta perancangan antarmuka pengguna (*User Interface*). *Prototype* awal digunakan sebagai media evaluasi sebelum sistem dikembangkan secara penuh.

3. Pembuatan *Prototype* (*Build Prototype*)

Tahap berikutnya adalah membangun *prototype* menggunakan *Flutter Framework* sebagai *frontend* dan *Retrieval-Based Voice Conversion Framework* sebagai *backend*. Pada

tahap ini dikembangkan fungsi utama aplikasi, meliputi proses autentikasi pengguna, komunikasi REST API, proses penerjemahan menggunakan mekanisme *Hybrid AI Translation* yang mengintegrasikan *AI Translation Engine* berbasis dataset lokal dengan *Google Gemini*, sintesis suara menggunakan *Edge Text-to-Speech* (Edge TTS), proses konversi suara menggunakan *Retrieval-Based Voice Conversion* (RVC), serta tampilan hasil terjemahan dan audio hasil konversi.

4. Evaluasi *Prototype* (*User Evaluation*)

Prototype yang telah dibuat kemudian diuji untuk mengetahui kesesuaian fungsi dengan kebutuhan pengguna. Evaluasi dilakukan terhadap kemudahan penggunaan antarmuka, komunikasi antara *frontend* dan backend, kecepatan respons sistem, kualitas hasil penerjemahan, serta kualitas suara hasil konversi. Apabila masih ditemukan kekurangan, maka dilakukan proses perbaikan hingga prototype memenuhi kebutuhan penelitian.

5. Pengembangan Sistem (*System Development*)

Setelah prototype dinyatakan sesuai, dilakukan pengembangan sistem secara menyeluruh. Tahap ini meliputi pengembangan *AI Translation Engine* berbasis dataset lokal, integrasi *Google Gemini* melalui mekanisme *Hybrid AI Translation*, implementasi *Retrieval-Based Voice Conversion* REST API, pengembangan aplikasi menggunakan *Flutter Framework*, penerapan *Edge Text-to-Speech* (Edge TTS), serta integrasi *model Retrieval-Based Voice Conversion* (RVC) menggunakan *platform Applio*. Selain itu dilakukan optimasi komunikasi data antara *frontend* dan backend agar proses penerjemahan dan konversi suara dapat berjalan secara efisien.

6. Pengujian Sistem (*System Testing*)

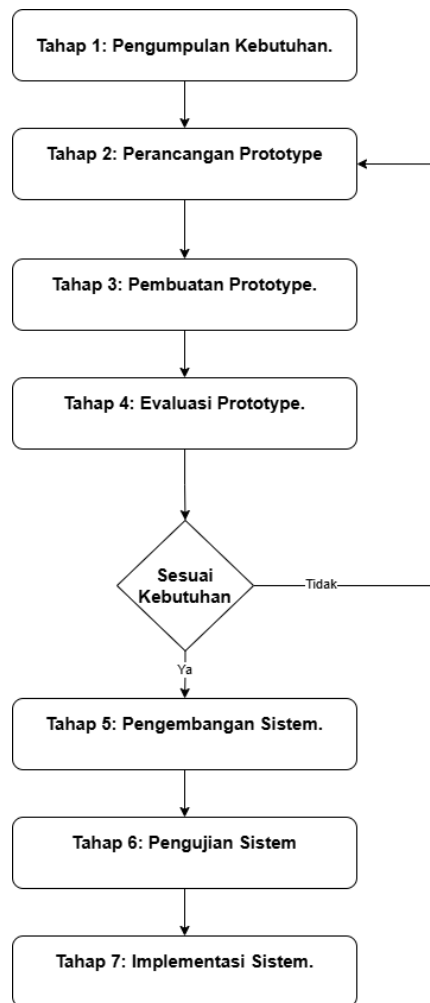
Tahap pengujian dilaksanakan guna memastikan seluruh fungsi aplikasi telah berjalan sesuai dengan kebutuhan sistem. Pengujian dilakukan menggunakan metode *Black Box Testing* untuk memastikan seluruh fungsi aplikasi bekerja dengan baik, sedangkan metode *Mean Opinion Score* (MOS) digunakan dalam mengevaluasi kualitas dan tingkat naturalitas suara hasil konversi berdasarkan. Hasil dari pengujian tersebut dijadikan dasar untuk melakukan penyempurnaan sistem.

7. Implementasi Sistem (*Implementation*)

Tahap terakhir merupakan implementasi aplikasi yang telah selesai dikembangkan. Sistem dijalankan pada perangkat Android menggunakan *Flutter Framework* sebagai *frontend*,

sedangkan proses penerjemahan dilakukan oleh backend *Retrieval-Based Voice Conversion* melalui mekanisme *Hybrid AI Translation*, yaitu menggunakan *Google Gemini* ketika perangkat terhubung ke internet dan *AI Translation Engine* berbasis dataset lokal ketika perangkat berada dalam kondisi *offline*. Hasil terjemahan selanjutnya diproses menggunakan *Edge Text-to-Speech* (Edge TTS) dan dikonversi menggunakan *Retrieval-Based Voice Conversion* (RVC) sehingga menghasilkan keluaran berupa teks Bahasa Sunda dan audio yang dapat diputar maupun diunduh melalui aplikasi.

Secara keseluruhan, penerapan metode Prototyping memberikan fleksibilitas dalam pengembangan aplikasi LingoNusantara AI karena setiap komponen sistem dapat dikembangkan dan dievaluasi secara bertahap. Pendekatan ini memungkinkan proses penyempurnaan dilakukan berdasarkan hasil pengujian sehingga aplikasi yang dihasilkan lebih sesuai dengan kebutuhan pengguna dan tujuan penelitian.



Gambar 3. 3 Tahapan Metode Prototyping pada Penelitian

Sumber: Diadaptasi dari

Pressman, R. S., & Maxim,

B. R. (2020). *Software Engineering: A Practitioner's Approach* (9th Edition). McGraw-Hill Education.

3.3.1 Dataset AI Translation

Dataset *AI Translation* terdiri atas pasangan kalimat Bahasa Indonesia dan Bahasa Sunda yang digunakan sebagai data latih (*training data*) untuk membangun *AI Translation Engine*. Dataset tersebut memuat kosakata, frasa, idiom, serta struktur kalimat yang digunakan sistem dalam menghasilkan terjemahan. Selain digunakan sebagai data latih, penelitian ini juga menggunakan data uji (*testing data*) berupa sejumlah kalimat Bahasa Indonesia yang tidak termasuk dalam data pelatihan untuk menguji kemampuan sistem dalam menghasilkan terjemahan Bahasa Sunda.

3.3.2 Dataset Retrieval-Based Voice Conversion (RVC)

Dataset suara digunakan sebagai data latih (*training data*) pada proses pelatihan model *Retrieval-Based Voice Conversion* (RVC). Dataset terdiri atas rekaman suara berbahasa Sunda yang kemudian diproses menggunakan platform Applio hingga menghasilkan model .pth dan .index. Setelah proses pelatihan selesai, model tersebut digunakan untuk melakukan proses inferensi terhadap data uji (*testing data*), yaitu audio hasil *Edge Text-to-Speech* (Edge TTS), guna menghasilkan suara dengan karakteristik yang menyerupai suara target.

3.4 Analisis Kebutuhan Sistem

Analisis kebutuhan sistem dilakukan untuk menentukan kebutuhan fungsional maupun nonfungsional yang dibutuhkan dalam proses pengembangan aplikasi LingoNusantara AI. Tahapan ini bertujuan memastikan seluruh komponen sistem dapat mendukung proses penerjemahan Bahasa Indonesia ke Bahasa Sunda, sintesis suara, konversi suara, serta penyimpanan data sesuai kebutuhan pengguna.

Berdasarkan hasil analisis yang telah dilakukan, sistem dikembangkan menggunakan *Flutter Framework* sebagai *frontend* aplikasi *mobile*, sedangkan *Retrieval-Based Voice Conversion Framework* digunakan sebagai backend yang berfungsi mengelola proses autentikasi pengguna, penerjemahan bahasa, sintesis suara menggunakan *Edge Text-to-Speech* (Edge TTS), konversi suara menggunakan *Retrieval-Based Voice Conversion* (RVC), serta komunikasi data melalui REST API.

Proses penerjemahan menggunakan mekanisme *Hybrid AI Translation*, yaitu menggabungkan *AI Translation Engine* berbasis dataset lokal dengan *Google Gemini* sebagai layanan *Large Language Model* (LLM). Ketika perangkat terhubung ke internet, sistem memanfaatkan *Google Gemini* untuk menghasilkan terjemahan yang lebih alami dan kontekstual. Sebaliknya, apabila koneksi internet tidak tersedia atau layanan Gemini tidak dapat digunakan, sistem secara otomatis menggunakan *AI Translation Engine* berbasis dataset lokal sehingga proses penerjemahan tetap dapat dilakukan.

Analisis kebutuhan sistem pada penelitian ini terdiri atas kebutuhan fungsional dan kebutuhan nonfungsional yang dijelaskan pada subbab berikut

Tabel 3. 5 Analisis Kebutuhan Fungsional

No	Kebutuhan Fungsional	Deskripsi
1	Memasukkan teks Bahasa Indonesia	Pengguna dapat memasukkan kalimat yang akan diterjemahkan ke dalam Bahasa Sunda.
2	Memilih dialek Bahasa Sunda	Pengguna dapat memilih dialek atau variasi bahasa yang tersedia pada sistem (apabila disediakan).
3	Merekam suara	Pengguna dapat merekam suara secara langsung melalui mikrofon perangkat Android.
4	Memilih file audio	Pengguna dapat memilih file audio dari penyimpanan perangkat sebagai suara masukan.
5	Mengirim data ke server	Aplikasi mengirimkan teks dan file audio ke backend <i>Retrieval-Based Voice Conversion</i> menggunakan REST API.
6	Melakukan penerjemahan	Backend menerjemahkan teks Bahasa Indonesia ke Bahasa Sunda menggunakan <i>AI Translation Engine</i> .
7	Melakukan konversi suara	Backend menjalankan proses inferensi menggunakan <i>model Retrieval-Based Voice Conversion</i> (RVC).
8	Menampilkan hasil terjemahan	Aplikasi menampilkan hasil terjemahan Bahasa Sunda kepada pengguna.
9	Memutar audio hasil konversi	Pengguna dapat mendengarkan audio hasil konversi suara melalui aplikasi.
10	Mengunduh hasil audio	Pengguna dapat menyimpan audio hasil konversi ke perangkat Android.
11	Menggunakan Gemini AI	Backend menggunakan Gemini AI untuk menerjemahkan teks Bahasa Indonesia ke Bahasa Sunda saat perangkat terhubung ke internet.
12	Fallback Translator	Backend menggunakan <i>AI Translation Engine</i> lokal apabila layanan Gemini AI tidak tersedia.
13	<i>Hybrid AI Translation</i>	Sistem melakukan proses penerjemahan menggunakan <i>Google Gemini</i> ketika perangkat terhubung ke internet dan menggunakan <i>AI Translation Engine</i> berbasis dataset lokal ketika perangkat berada dalam kondisi <i>offline</i> .

3.4.1 Analisis Kebutuhan Nonfungsional

Kebutuhan nonfungsional merupakan kebutuhan yang berkaitan dengan spesifikasi perangkat keras, perangkat lunak, performa sistem, serta lingkungan pengembangan yang

mendukung jalannya aplikasi. Menurut Dennis, Wixom, dan Tegarden (2020), kebutuhan nonfungsional mendefinisikan karakteristik sistem yang berkaitan dengan kualitas perangkat lunak, performa, keamanan, dan kemudahan penggunaan.

Dalam penelitian ini, kebutuhan nonfungsional dibagi menjadi kebutuhan perangkat keras dan perangkat lunak.

3.4.1.1 Kebutuhan Perangkat Keras

Perangkat keras yang digunakan selama proses pengembangan aplikasi LingoNusantara AI ditunjukkan pada Tabel 3.6.

Tabel 3. 6 Spesifikasi Perangkat Keras

No	Komponen	Spesifikasi
1	Laptop	Lenovo
2	Processor	Intel® Core™ i3
3	RAM	8 GB
4	Media Penyimpanan	SSD 256 GB
5	Smartphone	Android minimal versi 10
6	Mikrofon	Internal Smartphone
7	Jaringan	Wi-Fi / Hotspot Lokal
8	Koneksi Internet	Sistem dapat menggunakan <i>Google Gemini</i> ketika perangkat terhubung ke internet dan secara otomatis beralih ke <i>AI Translation Engine</i> lokal ketika koneksi internet tidak tersedia.

3.4.1.2 Kebutuhan Perangkat Lunak

Perangkat lunak yang digunakan selama proses pengembangan sistem ditunjukkan pada Tabel 3.7.

Tabel 3. 7 Kebutuhan Perangkat Lunak

No	Perangkat Lunak	Fungsi
1	Windows 11	Sistem operasi
2	Visual Studio Code	Editor kode program
3	<i>Flutter</i> SDK	Pengembangan aplikasi Android
4	Dart SDK	Bahasa pemrograman <i>Flutter</i>
5	Python 3.11	Pengembangan backend
6	<i>Retrieval-Based Voice Conversion</i> Framework	REST API Backend
7	<i>Applio</i>	Pelatihan dan inferensi <i>model</i> RVC
8	Git	Version Control
9	Android SDK	Build aplikasi Android
10	Google Chrome	Pengujian REST API dan dokumentasi
11	GPU	Intel UHD Graphics (Integrated Graphics)

3.4.1.3 Kebutuhan Pengguna

Pengguna aplikasi LingoNusantara AI merupakan masyarakat umum yang ingin menerjemahkan Bahasa Indonesia ke Bahasa Sunda dengan keluaran berupa teks dan suara. Pengguna tidak memerlukan proses registrasi maupun login sehingga aplikasi dapat langsung digunakan setelah dijalankan. Seluruh proses penerjemahan dan konversi suara dilakukan secara otomatis melalui backend *Retrieval-Based Voice Conversion* sehingga pengguna hanya perlu memasukkan teks dan merekam atau memilih file audio.

3.4.1.4 Analisis Arsitektur Sistem

Aplikasi LingoNusantara AI menggunakan arsitektur client-server. Aplikasi *Flutter* bertindak sebagai client yang menerima masukan dari pengguna, sedangkan *Retrieval-Based Voice Conversion* bertindak sebagai server yang menangani proses penerjemahan dan konversi suara. Proses kerja sistem dimulai ketika pengguna memasukkan teks Bahasa Indonesia dan merekam atau memilih file audio. Data tersebut dikirimkan ke backend *Retrieval-Based Voice Conversion* menggunakan REST API. Selanjutnya backend menjalankan proses penerjemahan menggunakan *AI Translation Engine* dan melakukan inferensi suara menggunakan *model Retrieval-Based Voice Conversion* (RVC) melalui *platform Applio*. Setelah proses selesai, backend mengirimkan hasil berupa teks Bahasa Sunda dan file audio kepada aplikasi *Flutter* untuk ditampilkan kepada pengguna.

3.5 Perancangan Sistem

Perancangan sistem merupakan tahapan yang dilakukan untuk menggambarkan rancangan aplikasi LingoNusantara AI sebelum proses implementasi. Tahap ini bertujuan untuk memberikan gambaran mengenai arsitektur sistem, interaksi antaraktor dengan sistem, alur proses aplikasi, struktur kelas, rancangan basis data, serta rancangan antarmuka pengguna. Perancangan sistem dilakukan menggunakan Unified Modeling Language (UML) yang meliputi Use Case Diagram, Activity Diagram, Sequence Diagram, dan Class Diagram. Selain itu, dilakukan pula perancangan basis data (Entity Relationship Diagram/ERD) serta perancangan antarmuka aplikasi sebagai acuan dalam proses implementasi menggunakan *Flutter Framework* dan *Retrieval-Based Voice Conversion Framework*.

Arsitektur sistem menerapkan mekanisme *Hybrid AI Translation*, yaitu mengintegrasikan *AI Translation Engine* berbasis dataset lokal dengan *Google Gemini* sebagai layanan *Large Language Model* (LLM). Hasil terjemahan kemudian diproses menggunakan *Edge Text-to-Speech* (Edge TTS) dan dikonversi menggunakan *Retrieval-Based Voice Conversion* (RVC) melalui *platform Applio* sehingga menghasilkan keluaran berupa teks dan audio yang dapat digunakan oleh pengguna.

Perancangan sistem pada penelitian ini meliputi beberapa bagian, yaitu:

1. Perancangan Arsitektur Sistem.
2. Use Case Diagram.
3. Activity Diagram.
4. Sequence Diagram.
5. Class Diagram.
6. Entity Relationship Diagram (ERD).
7. Perancangan Antarmuka (*User Interface*).

3.5.1 Perancangan Arsitektur Sistem

Berdasarkan hasil analisis kebutuhan sistem yang telah dilakukan pada tahap sebelumnya, arsitektur aplikasi LingoNusantara AI dirancang menggunakan pola *client-server* untuk memisahkan proses antarmuka pengguna dengan proses pengolahan data. Pendekatan ini dipilih karena proses penerjemahan menggunakan *Hybrid AI Translation*, sintesis suara menggunakan *Edge Text-to-Speech* (Edge TTS), serta konversi suara menggunakan *Retrieval-Based Voice Conversion* (RVC) memerlukan proses komputasi yang lebih kompleks sehingga

dijalankan pada sisi backend. Dengan rancangan tersebut, aplikasi Flutter berperan sebagai antarmuka pengguna, sedangkan backend bertanggung jawab dalam memproses penerjemahan, sintesis suara, konversi suara, serta pengelolaan data yang digunakan oleh sistem.

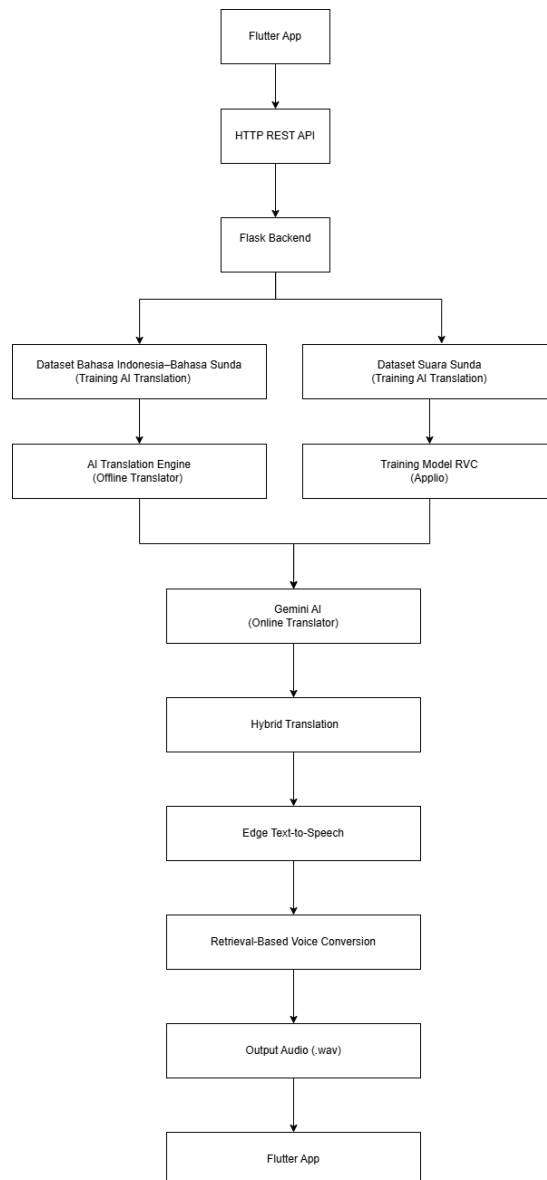
Pada penelitian ini, aplikasi dikembangkan menggunakan arsitektur client-server. Aplikasi Android dibangun menggunakan *Flutter Framework* sebagai *frontend*, sedangkan backend dikembangkan menggunakan *Retrieval-Based Voice Conversion Framework* berbasis bahasa pemrograman Python. Seluruh proses penerjemahan, pengolahan teks, sintesis suara, serta konversi suara dijalankan pada sisi backend sehingga aplikasi Android hanya berfungsi sebagai antarmuka pengguna dan media komunikasi dengan server.

Berbeda dengan penelitian sebelumnya, sistem pada penelitian ini menerapkan konsep *Hybrid AI Translation*, yaitu menggabungkan *AI Translation Engine* berbasis dataset lokal dengan layanan Gemini AI. *AI Translation Engine* digunakan sebagai penerjemah utama ketika aplikasi berjalan secara *offline*, sedangkan Gemini AI dimanfaatkan ketika perangkat terhubung ke internet untuk menghasilkan terjemahan yang lebih alami serta memperkaya kosakata Bahasa Sunda. Dengan pendekatan tersebut, aplikasi tetap dapat digunakan tanpa koneksi internet, namun mampu meningkatkan kualitas hasil terjemahan ketika layanan Gemini AI tersedia.

Setelah proses penerjemahan selesai, hasil teks Bahasa Sunda diteruskan ke modul *Edge Text-to-Speech* (Edge TTS) untuk menghasilkan suara dasar dalam bentuk file audio. Selanjutnya file audio tersebut diproses menggunakan teknologi *Retrieval-Based Voice Conversion* (RVC) melalui aplikasi *Applio* sehingga suara yang dihasilkan memiliki karakteristik sesuai *model* suara yang telah dilatih sebelumnya. File audio hasil konversi kemudian disimpan pada server dan dikirimkan kembali ke aplikasi Android untuk diputar, diunduh, maupun disimpan ke dalam riwayat penerjemahan.

Arsitektur ini memungkinkan proses penerjemahan, sintesis suara, dan konversi suara berjalan secara terintegrasi sehingga pengguna memperoleh hasil terjemahan Bahasa Sunda beserta keluaran audio yang terdengar lebih natural dan sesuai dengan karakter suara yang diinginkan.

Secara keseluruhan, arsitektur sistem dibangun dari lima komponen utama, yaitu:



Gambar 3. 4 Arsitektur Sistem LingoNusantara AI

Dataset AI Translation terdiri atas pasangan kalimat Bahasa Indonesia dan Bahasa Sunda yang digunakan sebagai data latih (*training data*) untuk membangun AI Translation Engine. Dataset tersebut memuat kosakata, frasa, idiom, serta struktur kalimat yang digunakan sistem dalam menghasilkan terjemahan.

3.5.1.2 Perancangan Arsitektur Sistem

AI Translation Engine pada penelitian ini terlebih dahulu melakukan pencarian terhadap dataset lokal yang terdiri atas Dictionary Engine, Phrase Engine, Sentence Engine, Grammar Engine, Context Engine, dan Idiom Engine. Apabila kata, frasa, atau kalimat yang dimasukkan pengguna ditemukan pada dataset lokal, sistem akan menghasilkan terjemahan menggunakan AI Translation Engine.

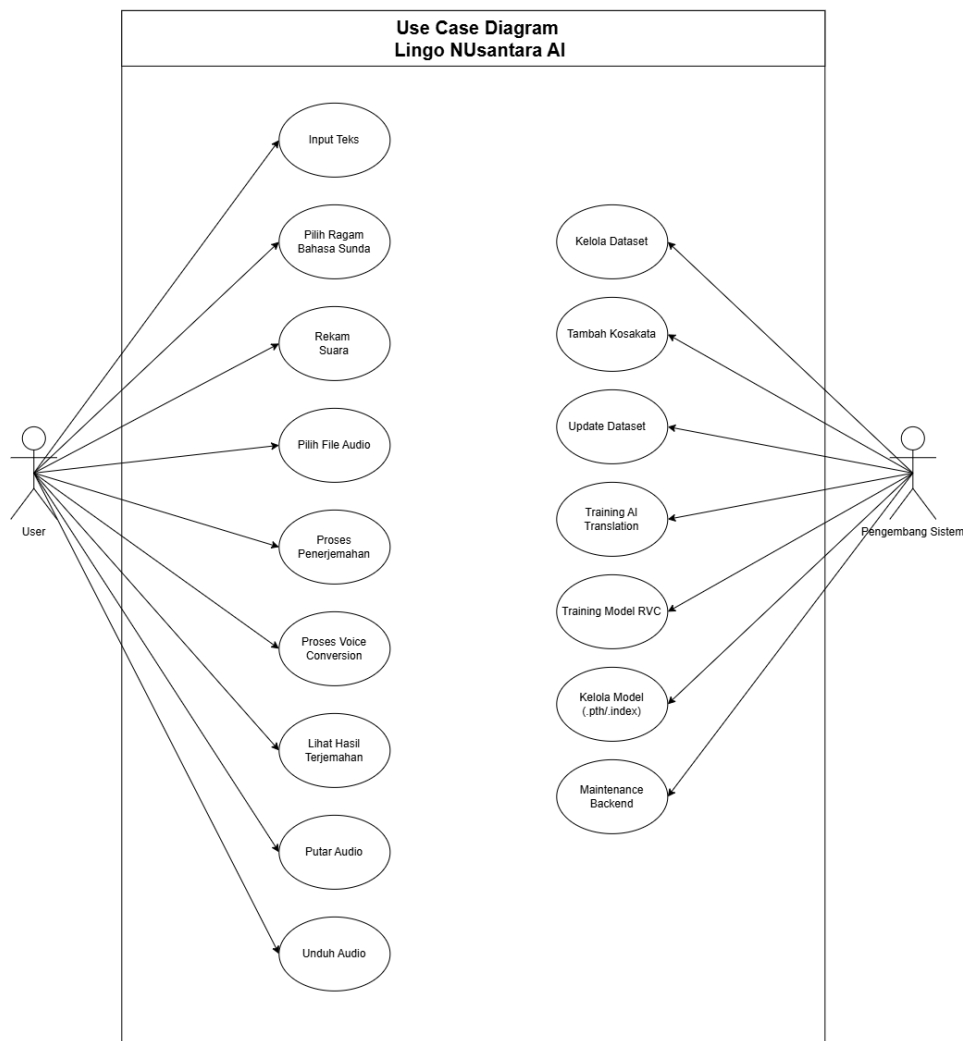
Sebaliknya, apabila kata atau kalimat tidak ditemukan pada dataset lokal, sistem akan melakukan pemeriksaan terhadap koneksi internet. Jika koneksi internet tersedia, proses penerjemahan dialihkan ke Google Gemini melalui mekanisme Hybrid AI Translation sehingga sistem tetap dapat menghasilkan terjemahan yang kontekstual. Namun apabila koneksi internet tidak tersedia, sistem hanya menerjemahkan kosakata yang tersedia pada dataset lokal, sedangkan kosakata yang tidak ditemukan akan dipertahankan dalam bahasa asal atau ditampilkan sebagai kata yang belum tersedia pada dataset.

3.5.2 Use Case Diagram

Berdasarkan hasil analisis kebutuhan fungsional, aplikasi LingoNusantara AI dirancang untuk mendukung proses penerjemahan Bahasa Indonesia ke Bahasa Sunda serta konversi suara menggunakan teknologi *Retrieval-Based Voice Conversion* (RVC). Analisis tersebut menghasilkan dua aktor yang terlibat dalam sistem, yaitu Pengguna sebagai pengguna utama aplikasi dan Pengembang Sistem sebagai pihak yang bertanggung jawab terhadap pengembangan sistem. Pengguna berinteraksi secara langsung dengan aplikasi untuk memanfaatkan seluruh fitur yang tersedia, sedangkan Pengembang Sistem berperan dalam pengelolaan dataset, penambahan kosakata dan pasangan kalimat baru, pembaruan dataset, serta pelatihan ulang model AI dan model RVC sebagai bagian dari pengembangan sistem pada versi berikutnya.

Berdasarkan hasil analisis tersebut, aktor Pengguna memiliki hak akses untuk melakukan registrasi akun, login, memasukkan teks Bahasa Indonesia, memilih ragam Bahasa Sunda, melakukan proses penerjemahan, melakukan konversi suara menggunakan teknologi *Retrieval-Based Voice Conversion* (RVC), memutar dan mengunduh hasil audio, melihat riwayat penerjemahan, mengelola profil, mengubah pengaturan aplikasi, serta melihat informasi aplikasi. Sementara itu, aktor Pengembang Sistem bertanggung jawab terhadap pengelolaan dan pengembangan sistem, meliputi pengelolaan dataset bahasa, penambahan kosakata dan pasangan kalimat baru, pembaruan dataset, pelatihan ulang model AI Translation

dan model RVC, pengelolaan model yang digunakan, serta pemeliharaan backend agar sistem tetap berjalan dengan baik.



Gambar 3. 5 Use Case Diagram LingoNusantara AI

Deskripsi Use Case

1. Input Teks

Pengguna memasukkan kalimat dalam Bahasa Indonesia melalui halaman utama aplikasi. Data teks tersebut akan dikirimkan ke backend *Retrieval-Based Voice Conversion* untuk diproses

menggunakan *AI Translation Engine* sehingga menghasilkan terjemahan ke dalam Bahasa Sunda.

2. Pilih Ragam Bahasa Sunda

Pengguna memilih ragam Bahasa Sunda yang akan digunakan dalam proses penerjemahan, yaitu Sunda Lemes, Sunda Loma, atau Sunda Kasar. Pilihan ini akan dikirimkan ke backend sebagai parameter sehingga hasil terjemahan menyesuaikan ragam bahasa yang dipilih.

3. Rekam Suara

Pengguna melakukan perekaman suara secara langsung menggunakan mikrofon pada perangkat Android. Rekaman suara digunakan sebagai masukan dalam proses konversi suara menggunakan *model Retrieval-Based Voice Conversion (RVC)*.

4. Pilih File Audio

Selain melakukan perekaman secara langsung, pengguna juga dapat memilih file audio yang tersimpan pada perangkat Android sebagai data masukan untuk proses konversi suara.

5. Proses Penerjemahan

Backend *Retrieval-Based Voice Conversion* menerima data dari aplikasi *Flutter* kemudian menjalankan *AI Translation Engine* untuk menerjemahkan teks Bahasa Indonesia menjadi Bahasa Sunda berdasarkan ragam bahasa yang dipilih oleh pengguna.

6. Proses Voice Conversion

Setelah proses penerjemahan selesai, backend menjalankan proses inferensi menggunakan *model Retrieval-Based Voice Conversion (RVC)* melalui *platform Applio* sehingga menghasilkan suara yang memiliki karakteristik menyerupai suara asli pengguna.

7. Lihat Hasil Terjemahan

Aplikasi menampilkan hasil terjemahan Bahasa Sunda kepada pengguna setelah seluruh proses pemrosesan selesai dilakukan.

8. Putar Audio

Pengguna dapat memutar audio hasil konversi menggunakan fitur pemutar audio yang tersedia pada aplikasi untuk mendengarkan hasil suara yang telah dihasilkan.

9. Unduh Audio

Pengguna dapat mengunduh file audio hasil konversi ke penyimpanan perangkat Android sehingga dapat diputar kembali maupun digunakan pada kebutuhan lain.

10. Kelola Dataset

Pengembang Sistem mengelola dataset Bahasa Indonesia dan Bahasa Sunda yang digunakan oleh AI Translation Engine. Proses ini meliputi penambahan, pengubahan, maupun penghapusan data agar kualitas hasil penerjemahan terus meningkat.

11. Tambah Kosakata

Pengembang Sistem menambahkan kosakata maupun pasangan kalimat baru ke dalam dataset sebagai proses pengembangan sistem sehingga cakupan penerjemahan menjadi lebih luas.

12. Training AI Translation

Pengembang Sistem melakukan pelatihan ulang model AI Translation menggunakan dataset terbaru agar akurasi penerjemahan meningkat.

13. Training Model RVC

Pengembang Sistem melakukan pelatihan model Retrieval-Based *Voice* Conversion (RVC) menggunakan dataset suara sehingga menghasilkan model suara (.pth dan .index) yang digunakan oleh aplikasi.

14. Maintenance Backend

Pengembang Sistem melakukan pemeliharaan backend untuk memastikan seluruh layanan penerjemahan, Text-to-Speech, dan Retrieval-Based *Voice* Conversion berjalan dengan baik.

Tabel 3. 8 Daftar Use Case

No	Use Case	Deskripsi
----	----------	-----------

1	Input Teks	Memasukkan kalimat Bahasa Indonesia yang akan diterjemahkan.
2	Pilih Ragam Bahasa Sunda	Memilih ragam Bahasa Sunda (Lemes, Loma, atau Kasar) sebagai parameter penerjemahan.
3	Rekam Suara	Merekam suara menggunakan mikrofon perangkat Android.
4	Pilih File Audio	Memilih file audio dari penyimpanan perangkat.
5	Proses Penerjemahan	Menerjemahkan teks Bahasa Indonesia menjadi Bahasa Sunda menggunakan AI Translation Engine.
6	Proses <i>Voice</i> Conversion	Menjalankan konversi suara menggunakan model Retrieval-Based <i>Voice</i> Conversion (RVC) melalui Applio.
7	Lihat Hasil Terjemahan	Menampilkan hasil terjemahan Bahasa Sunda.
8	Putar Audio	Memutar audio hasil konversi suara.
9	Unduh Audio	Menyimpan audio hasil konversi ke perangkat Android.
10	Kelola Dataset	Mengelola dataset bahasa yang digunakan sistem.
11	Tambah Kosakata	Menambahkan kosakata dan pasangan kalimat baru.
12	Training AI Translation	Melatih ulang model AI Translation.
13	Training Model RVC	Melatih ulang model Retrieval-Based <i>Voice</i> Conversion (RVC).
14	Maintenance Backend	Melakukan pemeliharaan backend.

3.5.3 Activity Diagram

Berdasarkan hasil analisis proses bisnis aplikasi, Activity Diagram disusun untuk menggambarkan alur kerja sistem mulai dari pengguna memberikan masukan hingga sistem menghasilkan keluaran berupa teks terjemahan dan audio hasil konversi suara. Diagram ini menunjukkan urutan aktivitas yang terjadi pada setiap komponen sistem sehingga proses penerjemahan, sintesis suara, dan konversi suara dapat dipahami secara menyeluruh.

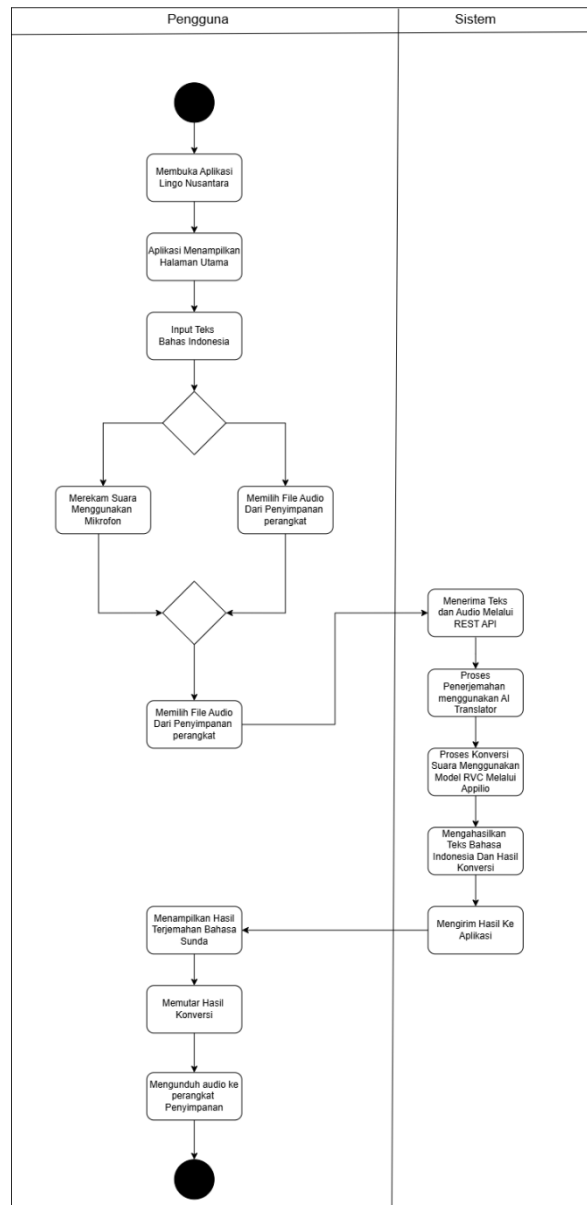
Activity Diagram pada penelitian ini menggambarkan alur aktivitas yang terjadi selama proses penerjemahan Bahasa Indonesia ke Bahasa Sunda pada aplikasi LingoNusantara AI. Proses diawali ketika pengguna memasukkan teks dan memilih ragam Bahasa Sunda, kemudian sistem mengirimkan data ke backend melalui REST API. Backend selanjutnya menentukan mekanisme penerjemahan menggunakan konsep *Hybrid AI Translation*, yaitu memanfaatkan Google Gemini ketika koneksi internet tersedia atau menggunakan AI Translation Engine berbasis dataset lokal ketika sistem berjalan secara offline. Setelah proses penerjemahan selesai, hasil teks diproses menggunakan *Edge Text-to-Speech* (Edge TTS)

untuk menghasilkan audio dasar, kemudian dilakukan konversi suara menggunakan model *Retrieval-Based Voice Conversion* (RVC). Hasil akhir berupa teks terjemahan dan file audio dikirim kembali ke aplikasi Flutter untuk ditampilkan, diputar, disimpan dalam riwayat, maupun diunduh oleh pengguna.

Alur proses dimulai ketika pengguna memasukkan teks atau suara ke dalam aplikasi kemudian memilih ragam Bahasa Sunda yang diinginkan, yaitu Loma, Lemes, atau Kasar. Selanjutnya aplikasi mengirimkan permintaan ke backend untuk diproses menggunakan *Hybrid AI Translation*, yaitu *AI Translation Engine* lokal maupun Gemini AI apabila koneksi internet tersedia. Setelah memperoleh hasil terjemahan, backend menghasilkan audio menggunakan *Edge Text-to-Speech* dan meneruskannya ke modul RVC untuk melakukan konversi suara berdasarkan *model* yang telah dilatih sebelumnya. Hasil akhir berupa teks terjemahan dan file audio kemudian dikirim kembali ke aplikasi sehingga dapat diputar, diunduh, maupun disimpan ke dalam riwayat penerjemahan.

Alur Activity Diagram

1. Pengguna membuka aplikasi LingoNusantara AI.
2. Sistem menampilkan halaman utama aplikasi.
3. Pengguna memasukkan teks Bahasa Indonesia.
4. Pengguna merekam suara atau memilih file audio.
5. Aplikasi mengirimkan teks dan audio ke backend *Retrieval-Based Voice Conversion* melalui REST API.
6. Backend menjalankan proses penerjemahan menggunakan *AI Translation Engine*.
7. Backend menjalankan inferensi menggunakan *model Retrieval-Based Voice Conversion* (RVC) melalui *Applio*.
8. Backend menghasilkan teks terjemahan dan file audio hasil konversi.
9. Aplikasi menampilkan hasil terjemahan kepada pengguna.
10. Pengguna dapat memutar audio hasil konversi.
11. Pengguna dapat mengunduh hasil audio ke perangkat Android.
12. Proses selesai.



Gambar 3. 6 Activity Diagram

Keterangan Activity Diagram

Berdasarkan Gambar 3.5, proses dimulai ketika pengguna membuka halaman penerjemahan dan memasukkan teks atau memilih masukan berupa suara. Pengguna kemudian menentukan ragam Bahasa Sunda yang akan digunakan sebelum menjalankan proses penerjemahan.

Aplikasi *Flutter* mengirimkan data ke backend *Retrieval-Based Voice Conversion* melalui REST API. Backend selanjutnya melakukan proses penerjemahan menggunakan *AI Translation Engine* lokal atau Gemini AI sesuai kondisi sistem. Setelah hasil terjemahan diperoleh, backend menghasilkan audio menggunakan *Edge Text-to-Speech* yang kemudian

diproses kembali menggunakan *Retrieval-Based Voice Conversion* (RVC) agar menghasilkan karakter suara sesuai *model* yang telah dilatih. Setelah seluruh proses selesai, backend mengirimkan hasil terjemahan beserta file audio kepada aplikasi *Flutter*. Pengguna kemudian dapat melihat hasil terjemahan, memutar audio, mengunduh hasil konversi suara, serta menyimpan riwayat penerjemahan ke dalam basis data.

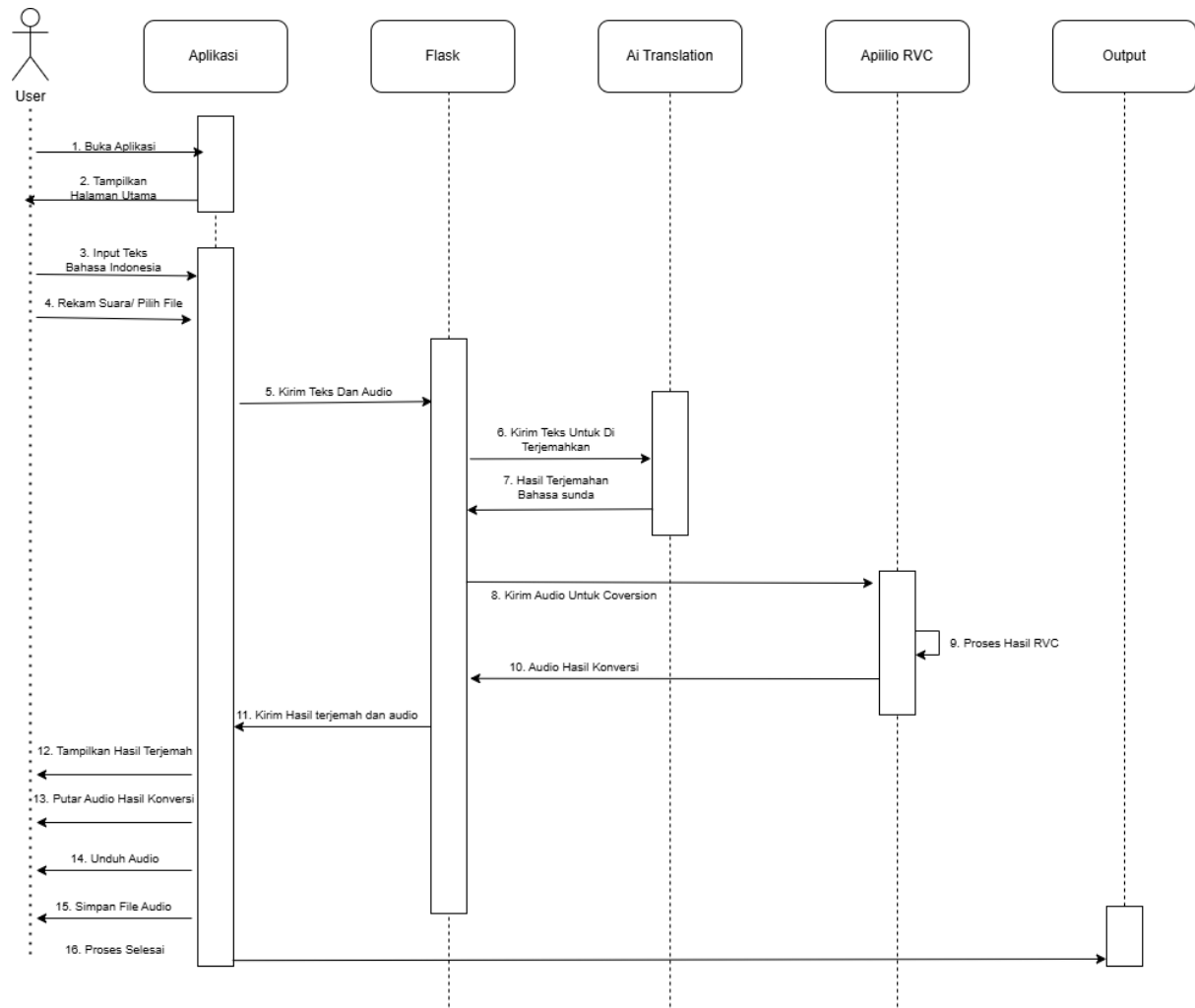
3.5.4 Sequence Diagram

Berdasarkan hasil analisis komunikasi antar komponen sistem, Sequence Diagram dirancang untuk menggambarkan urutan pertukaran data dan proses yang terjadi mulai dari pengguna memberikan masukan hingga sistem menghasilkan keluaran berupa teks terjemahan dan audio hasil konversi suara. Diagram ini memperlihatkan bagaimana setiap komponen saling berkomunikasi melalui mekanisme *request* dan *response* sehingga seluruh proses dapat berlangsung secara terintegrasi.

Sequence Diagram pada penelitian ini menggambarkan urutan komunikasi antara pengguna, aplikasi Flutter, backend, AI Translation Engine, Google Gemini, Edge Text-to-Speech (Edge TTS), modul Retrieval-Based *Voice Conversion* (RVC), serta basis data selama proses penerjemahan berlangsung. Proses dimulai ketika pengguna mengirimkan teks dan data suara melalui aplikasi Flutter. Selanjutnya backend menerima permintaan tersebut dan menentukan mekanisme penerjemahan menggunakan konsep *Hybrid AI Translation*. Apabila perangkat terhubung ke internet, backend memanfaatkan Google Gemini untuk menghasilkan terjemahan yang lebih alami. Sebaliknya, apabila koneksi internet tidak tersedia, sistem menggunakan AI Translation Engine berbasis dataset lokal. Hasil terjemahan kemudian diproses menggunakan Edge TTS untuk menghasilkan audio dasar sebelum dilakukan konversi suara menggunakan model RVC melalui platform Applio. Setelah seluruh proses selesai, backend menyimpan informasi yang diperlukan ke dalam basis data dan mengirimkan hasil terjemahan beserta file audio kepada aplikasi Flutter untuk ditampilkan kepada pengguna.

Proses dimulai ketika pengguna memasukkan teks atau suara ke dalam aplikasi kemudian memilih ragam Bahasa Sunda yang diinginkan. Aplikasi *Flutter* mengirimkan data ke backend *Retrieval-Based Voice Conversion* melalui REST API. Backend selanjutnya meneruskan permintaan ke *Hybrid AI Translator* yang akan menentukan proses penerjemahan menggunakan *AI Translation Engine* lokal maupun Gemini AI apabila koneksi internet tersedia. Setelah hasil terjemahan diperoleh, backend menghasilkan audio menggunakan Edge *Text-to-Speech* kemudian melakukan konversi suara melalui modul RVC. Seluruh hasil

penerjemahan dan informasi audio disimpan ke dalam database sebelum dikirim kembali ke aplikasi *Flutter* untuk ditampilkan kepada pengguna.



Gambar 3. 7 Sequence Diagram

Penjelasan Sequence Diagram

Berdasarkan Gambar 3.6, proses diawali ketika User memasukkan teks Bahasa Indonesia atau memilih masukan berupa suara pada aplikasi *Flutter*. Selanjutnya aplikasi mengirimkan permintaan penerjemahan ke backend *Retrieval-Based Voice Conversion* melalui REST API.

Backend kemudian menjalankan proses penerjemahan menggunakan *Hybrid AI Translator*. Pada tahap ini sistem dapat menggunakan *AI Translation Engine* berbasis dataset lokal maupun Gemini AI apabila perangkat terhubung ke internet. Setelah memperoleh hasil

terjemahan Bahasa Sunda, backend menghasilkan audio menggunakan Edge *Text-to-Speech* sebagai suara dasar.

Audio hasil sintesis kemudian diteruskan ke modul *Retrieval-Based Voice Conversion* (RVC) yang dijalankan menggunakan *Applio* untuk menghasilkan karakter suara sesuai *model* yang telah dilatih sebelumnya. Setelah proses konversi selesai, backend menyimpan informasi riwayat penerjemahan dan lokasi file audio ke dalam database *SQLite*.

Tahap terakhir, backend mengirimkan hasil terjemahan beserta file audio ke aplikasi *Flutter* sehingga pengguna dapat melihat hasil terjemahan, memutar audio, mengunduh hasil konversi suara, serta menyimpan riwayat penggunaan aplikasi.

Alur Sequence Diagram

1. Pengguna membuka aplikasi LingoNusantara AI.
2. Aplikasi *Flutter* menampilkan halaman utama.
3. Pengguna memasukkan teks Bahasa Indonesia.
4. Pengguna merekam suara atau memilih file audio.
5. *Flutter* mengirimkan teks dan audio ke *Retrieval-Based Voice Conversion* REST API.
6. *Retrieval-Based Voice Conversion* mengirimkan teks ke *AI Translation Engine*.
7. *AI Translation Engine* mengembalikan hasil terjemahan Bahasa Sunda.
8. *Retrieval-Based Voice Conversion* mengirimkan audio ke *Applio* RVC.
9. *Applio* menjalankan proses inferensi menggunakan *model Retrieval-Based Voice Conversion* (RVC).
10. *Applio* mengembalikan audio hasil konversi ke *Retrieval-Based Voice Conversion*.
11. *Retrieval-Based Voice Conversion* mengirimkan teks terjemahan dan audio hasil konversi ke aplikasi *Flutter*.
12. *Flutter* menampilkan hasil terjemahan kepada pengguna.
13. Pengguna memutar audio hasil konversi.
14. Pengguna mengunduh audio hasil konversi ke perangkat Android.
15. Proses selesai.

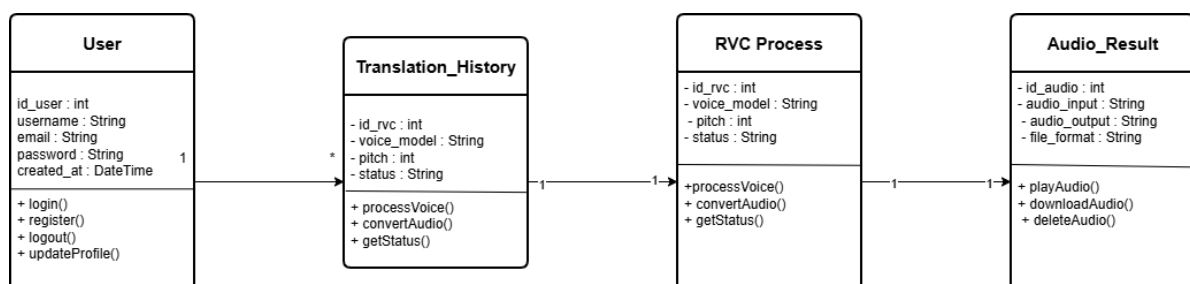
3.5.5 Class Diagram

Berdasarkan hasil analisis struktur sistem, aplikasi LingoNusantara AI dirancang menggunakan beberapa kelas yang saling berhubungan untuk mendukung proses penerjemahan Bahasa Indonesia ke Bahasa Sunda, pengelolaan data pengguna, serta konversi

suara menggunakan teknologi *Retrieval-Based Voice Conversion* (RVC). Pembagian kelas dilakukan agar setiap komponen memiliki tanggung jawab yang jelas sehingga proses pengembangan, pemeliharaan, dan pengembangan fitur pada aplikasi dapat dilakukan secara lebih terstruktur.

Class Diagram pada penelitian ini menggambarkan hubungan antara kelas antarmuka (*page*), kelas layanan (*service*), dan kelas model (*model*) yang digunakan dalam pengembangan aplikasi LingoNusantara AI. Kelas antarmuka berfungsi menangani interaksi pengguna dengan aplikasi, kelas layanan bertanggung jawab mengelola komunikasi antara aplikasi Flutter dan backend melalui REST API, sedangkan kelas model digunakan untuk merepresentasikan data yang diproses selama proses penerjemahan dan konversi suara. Pembagian struktur kelas tersebut mendukung penerapan arsitektur yang bersifat modular sehingga setiap komponen dapat dikembangkan maupun dipelihara secara mandiri tanpa memengaruhi komponen lainnya.

Berdasarkan hasil analisis implementasi sistem, struktur kelas dibagi ke dalam beberapa kelompok utama, yaitu kelas antarmuka yang terdiri atas halaman aplikasi seperti *SplashPage*, *DashboardPage*, *TranslatePage*, *HistoryPage*, dan *ProfilePage*; kelas layanan yang meliputi *ApiService*, *AuthService*, dan *HistoryService*; serta kelas model yang digunakan untuk merepresentasikan data pengguna, hasil penerjemahan, proses *Retrieval-Based Voice Conversion* (RVC), dan data audio. Pembagian tersebut bertujuan agar setiap kelas memiliki tanggung jawab yang spesifik sehingga alur komunikasi antar komponen menjadi lebih terorganisasi. Penjelasan mengenai fungsi masing-masing kelas dijabarkan pada bagian penjelasan Class Diagram setelah gambar.



Gambar 3.7 Class Diagram LingoNusantara AI

Penjelasan Class Diagram

Berdasarkan Gambar 3.7, Main merupakan titik awal (*entry point*) aplikasi yang bertugas menjalankan aplikasi *Flutter* dan mengarahkan pengguna menuju *SplashPage* saat

aplikasi pertama kali dibuka. Setelah proses *splash screen* selesai, sistem menampilkan LoginPage untuk melakukan autentikasi pengguna. Apabila proses autentikasi berhasil, pengguna akan diarahkan menuju DashboardPage sebagai halaman utama aplikasi.

DashboardPage berfungsi sebagai pusat navigasi menuju seluruh fitur utama aplikasi, yaitu TranslatePage, RVCPage, HistoryPage, ProfilePage, dan SettingsPage. Melalui halaman ini pengguna dapat mengakses seluruh layanan yang tersedia pada aplikasi LingoNusantara AI. TranslatePage digunakan untuk memasukkan teks Bahasa Indonesia yang akan diterjemahkan ke dalam Bahasa Sunda. Selanjutnya proses penerjemahan dilakukan melalui ApiService yang mengirimkan permintaan (*request*) ke backend *Retrieval-Based Voice Conversion* dan menerima hasil terjemahan beserta informasi tingkat kepercayaan (*confidence*) dari *model Artificial Intelligence*.

Setelah proses penerjemahan selesai, RVCPage bertugas melakukan konversi suara menggunakan teknologi *Retrieval-Based Voice Conversion* (RVC) sehingga hasil terjemahan dapat diubah menjadi suara dengan karakter suara yang telah dilatih sebelumnya. Seluruh proses komunikasi dengan backend dilakukan melalui ApiService.

HistoryPage digunakan untuk menampilkan riwayat hasil penerjemahan yang sebelumnya telah disimpan ke dalam basis data. Pengelolaan data riwayat dilakukan oleh HistoryService, yang bertanggung jawab mengambil, menyimpan, dan menghapus data riwayat penerjemahan.

Proses autentikasi pengguna ditangani oleh AuthService, yang bertugas melakukan proses login, validasi pengguna, serta mengelola sesi pengguna selama aplikasi digunakan. Pada lapisan *model (model layer)*, *UserModel* digunakan untuk merepresentasikan data pengguna, *TranslationModel* menyimpan informasi hasil penerjemahan, *RVCModel* menyimpan informasi proses konversi suara menggunakan RVC, sedangkan *AudioModel* merepresentasikan data file audio hasil konversi suara.

Hubungan antar kelas pada aplikasi bersifat association, yaitu hubungan kerja sama antar kelas dalam menjalankan fungsi sistem. DashboardPage memiliki hubungan dengan halaman-halaman fitur sebagai media navigasi, sedangkan TranslatePage, RVCPage, HistoryPage, dan AuthService berinteraksi dengan berbagai layanan (*service*) untuk melakukan komunikasi dengan backend. Selanjutnya, setiap layanan memanfaatkan *model* data sebagai representasi informasi yang diproses dan disimpan selama aplikasi berjalan.

3.5.6 Perancangan Basis Data

Berdasarkan hasil analisis kebutuhan data, aplikasi LingoNusantara AI memerlukan mekanisme penyimpanan yang mampu mengelola informasi pengguna, riwayat penerjemahan, proses konversi suara, serta metadata file audio secara terstruktur. Perancangan basis data dilakukan untuk mendukung proses penyimpanan, pengambilan, dan pengelolaan data yang dihasilkan selama aplikasi digunakan sehingga setiap informasi dapat diakses kembali secara efisien pada proses penerjemahan maupun penelusuran Riwayat. Berdasarkan hasil analisis kebutuhan data, aplikasi LingoNusantara AI memerlukan mekanisme penyimpanan yang mampu mengelola informasi pengguna, riwayat penerjemahan, proses konversi suara, serta metadata file audio secara terstruktur. Perancangan basis data dilakukan untuk mendukung proses penyimpanan, pengambilan, dan pengelolaan data yang dihasilkan selama aplikasi digunakan sehingga setiap informasi dapat diakses kembali secara efisien pada proses penerjemahan maupun penelusuran riwayat.

Pemisahan antara penyimpanan metadata dan berkas audio dilakukan untuk mengurangi beban penyimpanan pada database serta meningkatkan performa backend *Retrieval-Based Voice Conversion* ketika melakukan proses inferensi menggunakan *model Retrieval-Based Voice Conversion* (RVC) melalui *platform Applio*. Dengan pendekatan tersebut, sistem dapat mengelola data secara lebih optimal tanpa mengurangi kecepatan proses penerjemahan maupun konversi suara. Perancangan basis data pada penelitian ini terdiri atas dua bagian utama, yaitu Entity Relationship Diagram (ERD) dan struktur tabel basis data yang digunakan selama proses implementasi sistem.

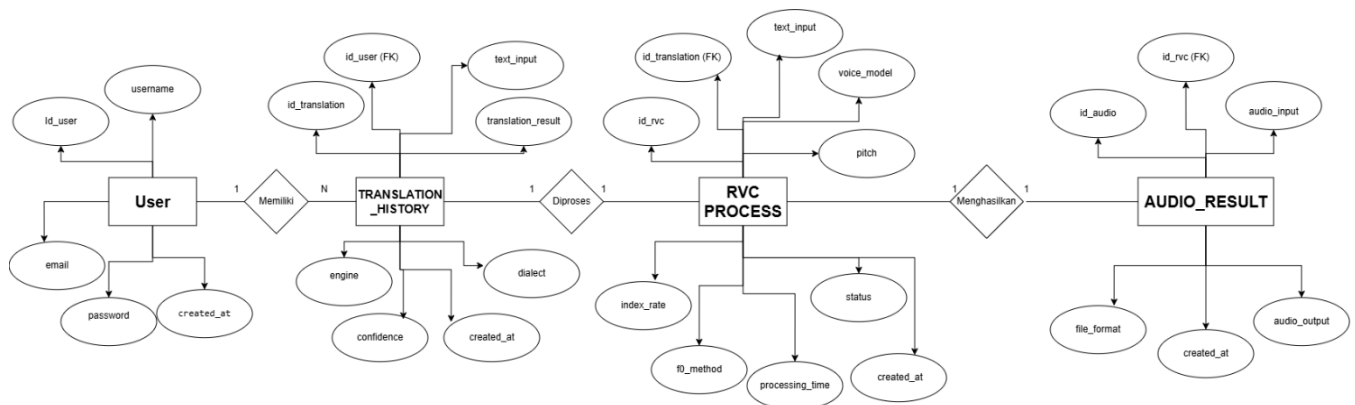
Berdasarkan hasil analisis hubungan antar data, struktur basis data disusun agar setiap proses yang dilakukan pengguna dapat direkam secara sistematis. Hubungan antar entitas dirancang untuk menjaga konsistensi data, memudahkan proses pencarian riwayat penerjemahan, serta mendukung integrasi antara hasil penerjemahan, proses konversi suara, dan file audio yang dihasilkan oleh sistem. Dengan rancangan tersebut, proses pengelolaan data menjadi lebih terstruktur dan mudah dikembangkan pada implementasi berikutnya.

3.5.6.1 Entity Relationship Diagram (ERD)

Berdasarkan hasil analisis kebutuhan data, hubungan antar entitas pada aplikasi LingoNusantara AI dirancang untuk mendukung proses autentikasi pengguna, penyimpanan riwayat penerjemahan, pengelolaan hasil konversi suara, serta penyimpanan informasi file audio yang dihasilkan oleh sistem. Hubungan antar entitas tersebut disusun agar setiap data

yang saling berkaitan dapat dikelola secara konsisten dan memudahkan proses pengambilan informasi pada saat aplikasi dijalankan.

Entity Relationship Diagram pada penelitian ini menggambarkan keterkaitan antara entitas pengguna, riwayat penerjemahan, hasil konversi suara, serta data pendukung lainnya yang digunakan selama proses penerjemahan berlangsung. Setiap entitas memiliki atribut yang disesuaikan dengan kebutuhan sistem sehingga mampu mendukung proses penyimpanan data secara terintegrasi. Hubungan antar entitas dirancang menggunakan relasi yang sesuai dengan kebutuhan aplikasi sehingga integritas data tetap terjaga ketika proses penyimpanan, pembaruan, maupun penghapusan data dilakukan.



Gambar 3. 8 Entity Relationship Diagram

Berdasarkan rancangan tersebut, setiap entitas memiliki fungsi yang berbeda namun saling terhubung dalam mendukung proses operasional aplikasi. Entitas User digunakan untuk menyimpan informasi akun pengguna, entitas History menyimpan riwayat hasil penerjemahan, sedangkan entitas yang berkaitan dengan proses *Retrieval-Based Voice Conversion* (RVC) menyimpan informasi mengenai hasil konversi suara beserta lokasi file audio yang dihasilkan. Hubungan antar entitas tersebut memungkinkan sistem mengelola data secara terstruktur sehingga informasi dapat ditampilkan kembali ketika pengguna mengakses riwayat penerjemahan.

Penjelasan Entity Relationship Diagram

Berdasarkan Gambar 3.8, hubungan antar entitas dapat dijelaskan sebagai berikut.

1. User memiliki hubungan One-to-Many (1:N) dengan Translation_History, sehingga satu pengguna dapat melakukan banyak proses penerjemahan, sedangkan setiap riwayat penerjemahan hanya dimiliki oleh satu pengguna.
2. Translation_History menyimpan hasil proses penerjemahan yang dihasilkan oleh *AI Translation Engine (Hybrid AI Translation)*. Entitas ini menyimpan informasi seperti *text_input*, *translation_result*, *dialect*, *engine*, *confidence*, dan *created_at* sebagai riwayat hasil penerjemahan sebelum dilakukan proses konversi suara.
3. Translation_History memiliki hubungan One-to-One (1:1) dengan RVC_Process, karena setiap hasil penerjemahan akan diproses satu kali oleh modul *Retrieval-Based Voice Conversion (RVC)*. Entitas RVC_Process menyimpan informasi proses konversi suara seperti *Voice_model*, *pitch*, *index_rate*, *f0_method*, *processing_time*, *status*, dan *created_at*.
4. RVC_Process memiliki hubungan One-to-One (1:1) dengan Audio_Result, karena setiap proses konversi suara menghasilkan satu file audio. Entitas Audio_Result menyimpan informasi *audio_input*, *audio_output*, *file_format*, serta waktu pembuatan (*created_at*) sehingga file audio dapat diputar maupun diunduh kembali melalui aplikasi *Flutter*.

3.5.6.2 Struktur Tabel Basis Data

Berdasarkan hasil analisis kebutuhan penyimpanan data, setiap tabel pada basis data dirancang untuk mendukung fungsi tertentu dalam aplikasi LingoNusantara AI. Struktur tabel disusun dengan menentukan atribut, tipe data, panjang karakter, serta fungsi masing-masing atribut sehingga proses penyimpanan, pencarian, dan pengelolaan data dapat dilakukan secara efisien. Selain itu, setiap tabel dihubungkan melalui *primary key* dan *foreign key* sesuai kebutuhan relasi antar data yang telah dirancang pada Entity Relationship Diagram (ERD).

Tabel 3. 9 Struktur Tabel User

Nama Kolom	Tipe Data	Keterangan
id_user	INT (PK)	Primary Key identitas pengguna
username	VARCHAR(100)	Nama pengguna
email	VARCHAR(100)	Alamat email pengguna
password	VARCHAR(255)	Password yang telah dienkripsi
created_at	TIMESTAMP	Waktu pembuatan akun

Tabel 3. 10 Struktur Tabel Translation_History

Nama Kolom	Tipe Data	Keterangan
id_translation	INT (PK)	Primary Key riwayat penerjemahan
id_user	INT (FK)	Foreign Key menuju tabel User
text_input	TEXT	Teks Bahasa Indonesia
translation_result	TEXT	Hasil terjemahan Bahasa Sunda
engine	VARCHAR(50)	Engine AI yang digunakan
confidence	INT	Tingkat kepercayaan hasil terjemahan
dialect	VARCHAR(30)	Jenis dialek Bahasa Sunda
created_at	TIMESTAMP	Waktu proses penerjemahan

Tabel 3. 11 Struktur Tabel RVC_Process

Nama Kolom	Tipe Data	Keterangan
id_rvc	INT (PK)	Primary Key proses RVC
id_translation	INT (FK)	Foreign Key menuju tabel Translation_History
<i>Voice_model</i>	VARCHAR(100)	<i>Model</i> suara RVC yang digunakan
pitch	INT	Nilai pitch suara
index_rate	FLOAT	Nilai index rate <i>model</i>
f0_method	VARCHAR(30)	Metode ekstraksi F0
processing_time	FLOAT	Lama proses konversi suara
status	VARCHAR(20)	Status proses RVC
created_at	TIMESTAMP	Waktu proses konversi

Tabel 3. 12 Struktur Tabel Audio_Result

Nama Kolom	Tipe Data	Keterangan
id_audio	INT (PK)	Primary Key hasil audio
id_rvc	INT (FK)	Foreign Key menuju tabel RVC_Process
audio_input	VARCHAR(255)	Lokasi file audio masukan
audio_output	VARCHAR(255)	Lokasi file audio hasil konversi
file_format	VARCHAR(10)	Format file audio
created_at	TIMESTAMP	Waktu pembuatan file audio

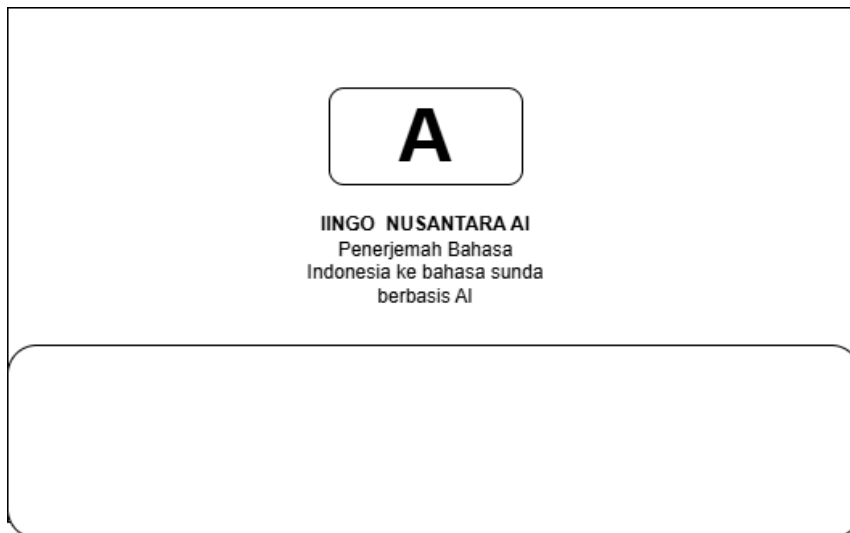
Berdasarkan struktur tabel yang telah dirancang, setiap data yang dibutuhkan aplikasi memiliki tempat penyimpanan yang terorganisasi sesuai dengan fungsinya masing-masing. Rancangan tersebut mendukung proses autentikasi pengguna, penerjemahan, penyimpanan riwayat, serta pengelolaan hasil konversi suara sehingga implementasi basis data dapat berjalan secara konsisten dan mudah dikembangkan pada tahap implementasi sistem.

3.5.7 Perancangan Antarmuka

Berdasarkan hasil analisis kebutuhan pengguna, perancangan antarmuka pada aplikasi LingoNusantara AI dilakukan untuk memberikan kemudahan dalam mengakses seluruh fitur yang tersedia. Antarmuka dirancang sesuai dengan alur penggunaan aplikasi, mulai dari proses autentikasi, penerjemahan Bahasa Indonesia ke Bahasa Sunda, konversi suara menggunakan *Retrieval-Based Voice Conversion* (RVC), hingga pengelolaan riwayat dan profil pengguna. Rancangan ini menjadi acuan dalam proses implementasi antarmuka aplikasi pada tahap pengembangan.

3.5.7.1 Halaman Splash Screen

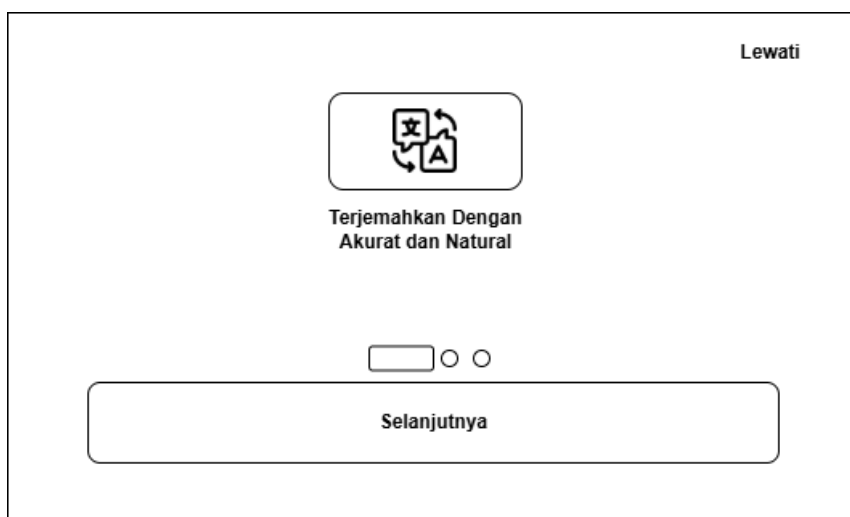
Halaman Splash Screen merupakan tampilan pertama yang muncul ketika aplikasi dijalankan. Halaman ini menampilkan logo dan identitas aplikasi LingoNusantara AI sebagai proses pembuka sebelum pengguna memasuki aplikasi. Selain sebagai identitas visual, halaman ini juga digunakan untuk melakukan proses inisialisasi sistem sehingga seluruh komponen aplikasi dapat dipersiapkan sebelum digunakan.



Gambar 3. 9 Rancangan Halaman Splash Screen

3.5.7.2 Halaman Onboarding

Halaman Onboarding berfungsi untuk memperkenalkan fitur utama aplikasi kepada pengguna sebelum melakukan proses login. Pada halaman ini ditampilkan informasi singkat mengenai kemampuan aplikasi dalam menerjemahkan Bahasa Indonesia ke Bahasa Sunda menggunakan *Artificial Intelligence* serta melakukan konversi suara menggunakan teknologi *Retrieval-Based Voice Conversion* (RVC). Pengguna dapat melanjutkan ke proses login ataupun melewati halaman ini.



Gambar 3. 10 Rancangan Halaman Onboarding

3.5.7.3 Halaman Login

Halaman Login digunakan sebagai proses autentikasi pengguna sebelum mengakses seluruh fitur aplikasi. Pengguna diminta memasukkan alamat email dan kata sandi yang telah terdaftar. Setelah proses autentikasi berhasil dilakukan, sistem akan mengarahkan pengguna menuju halaman Dashboard sebagai halaman utama aplikasi. Selain login menggunakan email, aplikasi juga menyediakan opsi autentikasi menggunakan akun Google untuk mempermudah proses masuk ke dalam sistem.

Selamat Datang

Email

Password

[Lupa Password?](#)

Masuk

Atau Masuk Dengan

Google Email

[Belum Punya Akun? Daftar di sini](#)

Gambar 3. 11 Rancangan Halaman Login

3.5.7.4 Halaman Register

Halaman Register digunakan untuk proses pendaftaran akun baru. Pengguna diminta mengisi informasi berupa nama lengkap, alamat email, kata sandi, serta konfirmasi kata sandi. Setelah seluruh data berhasil divalidasi, sistem akan menyimpan data pengguna sehingga akun dapat digunakan untuk proses login pada aplikasi.

Buat Akun Baru

Nama Lengkap

Email

Password

Konfirmasi Password

Daftar

Gambar 3. 12 Rancangan Halaman Register

3.5.7.5 Halaman Dashboard

Dashboard merupakan halaman utama yang ditampilkan setelah pengguna berhasil masuk (login) ke dalam aplikasi. Halaman ini menyediakan akses cepat menuju fitur-fitur utama aplikasi, yaitu Translate, *Voice Conversion*, dan History. Selain itu, Dashboard juga menampilkan informasi mengenai teknologi *Artificial Intelligence* yang digunakan serta riwayat aktivitas pengguna sehingga memudahkan pengguna dalam mengakses proses penerjemahan yang pernah dilakukan sebelumnya. Pada bagian bawah halaman tersedia Bottom Navigation yang digunakan untuk berpindah antarhalaman utama aplikasi, yaitu Beranda, History, *Voice Conversion*, dan Profile.

Nama User

Lingo Nusantara AI

Menu Utama

Translet

Voice Conversion

History

Seting

Gambar 3. 13 Rancangan Halaman Dashboard

3.5.7.6 Halaman Translate

Halaman Translate merupakan halaman utama yang digunakan pengguna untuk melakukan proses penerjemahan. Pengguna dapat memasukkan teks Bahasa Indonesia ke dalam area input, memilih dialek Bahasa Sunda yang diinginkan, serta memilih metode masukan suara melalui perekaman langsung ataupun mengunggah berkas audio. Setelah seluruh data dimasukkan, sistem akan memproses penerjemahan menggunakan *AI Translation Engine*.

Translate

Bahasa Indonesia

Dialek Sunda

Kecepatan Suara

Rekam Suara

Upload

Gambar 3. 14 Rancangan Halaman Translate

3.5.7.7 Halaman Hasil Terjemahan

Halaman Hasil Terjemahan digunakan untuk menampilkan hasil proses penerjemahan yang telah dilakukan sistem. Halaman ini menampilkan teks Bahasa Indonesia, hasil terjemahan Bahasa Sunda, serta menyediakan fitur untuk memutar audio hasil *Text-to-Speech*, menyalin teks hasil terjemahan, membagikan hasil terjemahan, dan memilih *model* suara yang akan digunakan pada proses *Voice Conversion*. Halaman ini menjadi penghubung antara proses penerjemahan teks dengan proses konversi suara menggunakan teknologi *Retrieval-Based Voice Conversion* (RVC).

The wireframe shows a 'Translate' interface. It has a title 'Translate' at the top. Below it, there are two text input fields. The first is labeled 'Bahasa Indonesia' and contains the text 'Saya mau makan'. The second is labeled 'Bahasa Sunda' and contains the text 'Abdi hoyong tuang'. Below these fields, there are two buttons: 'Dictionary Engine' and 'Loma'. Below these buttons, there are two more buttons: 'Berhenti' and 'Salin'. At the bottom, there is a large button labeled 'Bagikan'.

Gambar 3. 15 Rancangan Halaman Hasil Terjemahan

3.5.7.8 Halaman *Voice Conversion*

Halaman *Voice Conversion* digunakan untuk melakukan proses konversi suara menggunakan teknologi *Retrieval-Based Voice Conversion* (RVC). Pengguna dapat merekam suara secara langsung ataupun mengunggah berkas audio sebagai masukan. Selanjutnya pengguna memilih *model* suara yang akan digunakan sebelum sistem menghasilkan audio baru dengan karakteristik suara yang telah dipilih.

RVC Voice

Voice Transalator AI

Mulai Record

Process RVC

Play Result

Gambar 3. 16 Rancangan Halaman Voice Conversion

3.5.7.9 Halaman History

Halaman History digunakan untuk menampilkan seluruh riwayat proses penerjemahan yang pernah dilakukan oleh pengguna. Setiap riwayat memuat informasi berupa teks hasil penerjemahan, waktu proses, serta fitur pencarian sehingga pengguna dapat menemukan riwayat tertentu dengan lebih mudah.

Riwayat Terjemah

Saya mau makan
Abdi hoyong tuang

Copy Hapus

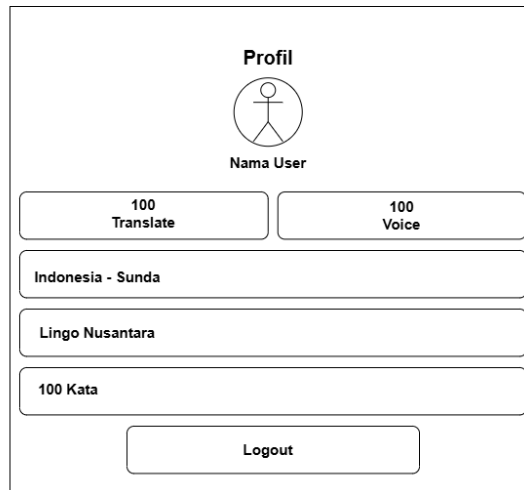
Saya mau makan
Abdi hoyong tuang

Copy Hapus

Gambar 3. 17 Rancangan Halaman History

3.5.7.10 Halaman Profile

Halaman Profile digunakan untuk menampilkan informasi akun pengguna yang sedang aktif. Halaman ini juga menyediakan menu untuk mengubah informasi profil, mengatur preferensi bahasa, melakukan pengaturan suara, serta keluar (*logout*) dari aplikasi.

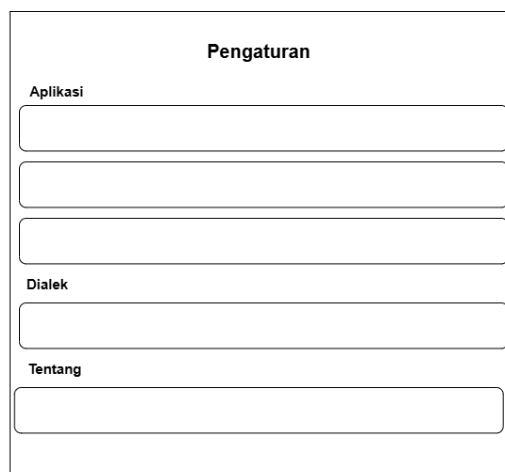


The wireframe for the Profile page is titled "Profil" and features a stick figure icon representing a user. Below the icon is a label "Nama User". The page contains several interactive elements: two buttons at the top labeled "100 Translate" and "100 Voice"; a language selection section with "Indonesia - Sunda" and "Lingo Nusantara" options; a "100 Kata" button; and a "Logout" button at the bottom.

Gambar 3. 18 Rancangan Halaman Profile

3.5.7.11 Halaman Settings

Halaman **Settings** digunakan untuk mengatur konfigurasi aplikasi sesuai dengan kebutuhan pengguna. Pengaturan yang tersedia meliputi pemilihan bahasa aplikasi, dialek Bahasa Sunda, *model* suara bawaan, kecepatan audio, pengaturan notifikasi, mode gelap (*dark mode*), serta informasi versi aplikasi.

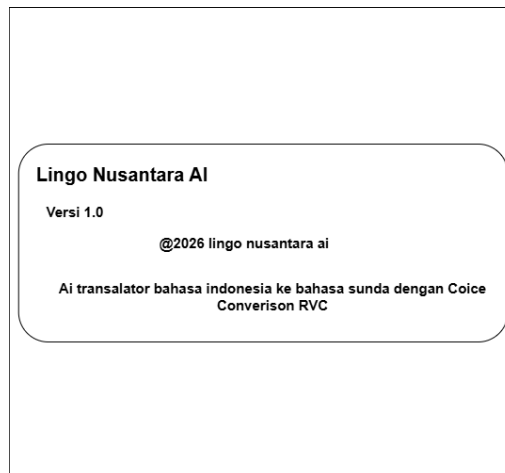


The wireframe for the Settings page is titled "Pengaturan". It includes a section labeled "Aplikasi" with three empty input fields. Below this is a "Dialek" section with one input field. At the bottom is a "Tentang" section with one input field.

Gambar 3. 19 Rancangan Halaman Settings

3.5.7.12 Halaman About

Halaman About berisi informasi umum mengenai aplikasi LingoNusantara AI, meliputi nama aplikasi, versi aplikasi, tujuan pengembangan, teknologi yang digunakan, serta identitas pengembang. Halaman ini bertujuan memberikan informasi kepada pengguna mengenai aplikasi yang digunakan serta ruang lingkup pengembangannya.



Gambar 3. 20 Rancangan Halaman About

Berdasarkan perancangan antarmuka yang telah dibuat, setiap halaman pada aplikasi LingoNusantara AI dirancang untuk mendukung proses penerjemahan Bahasa Indonesia ke Bahasa Sunda, konversi suara menggunakan teknologi *Retrieval-Based Voice Conversion* (RVC), serta pengelolaan data pengguna secara mudah dan terintegrasi. Rancangan antarmuka ini selanjutnya menjadi acuan dalam proses implementasi aplikasi yang dibahas pada Bab IV.

3.5.8 Evaluasi *Prototype*

Prototype yang telah dibangun kemudian dievaluasi untuk memastikan bahwa rancangan dan fungsi sistem telah sesuai dengan kebutuhan pengguna serta tujuan penelitian. Proses evaluasi dilakukan melalui peninjauan terhadap antarmuka aplikasi, alur kerja sistem, komunikasi antara aplikasi *Flutter* dengan backend *Retrieval-Based Voice Conversion*, serta implementasi fitur utama yang meliputi proses autentikasi, *Hybrid AI Translation*, dan *Retrieval-Based Voice Conversion* (RVC).

Evaluasi dilakukan dengan membandingkan prototype yang telah dikembangkan terhadap kebutuhan fungsional yang telah ditentukan pada tahap sebelumnya. Hasil evaluasi digunakan sebagai dasar untuk melakukan penyempurnaan pada struktur sistem, Class

Diagram, Entity Relationship Diagram (ERD), rancangan antarmuka pengguna (*User Interface*), serta alur proses penerjemahan dan konversi suara.

Apabila masih ditemukan kekurangan atau ketidaksesuaian pada prototype, maka dilakukan proses revisi dan penyempurnaan hingga prototype dinyatakan telah memenuhi kebutuhan pengguna dan tujuan penelitian. Setelah proses evaluasi dan perbaikan selesai dilakukan, pengembangan sistem dilanjutkan ke tahap implementasi secara menyeluruh.