

Implementasi Prototipe Pencocokan Kosakata Bahasa Indonesia dengan Berkas Audio Modul Taluk dan Minang Berbasis Python

R.A Rizka Qori Yuliani Putri^{1*}, Ika Mayla Sari¹, Siti Nurjanah¹, Ery Syafrianti¹, Dila Marta Putri¹ dan Linna Oktaviana Sari²

¹Program Studi S1 Teknik Elektro, Fakultas Teknik, Universitas Riau, Pekanbaru, Indonesia

²Program Studi S1 Teknik Informatika, Fakultas Teknik, Universitas Riau, Pekanbaru, Indonesia

*Penulis korespondensi, E-mail: rizkaqoriyuliani Putri@lecturer.unri.ac.id

Abstrak

Keragaman bahasa daerah mendorong pengembangan alat bantu digital untuk mengenalkan kosakata dan audio bahasa daerah. Penelitian ini menggunakan pendekatan implementatif-deskriptif untuk menganalisis prototipe Python yang mencocokkan kosakata bahasa Indonesia dengan berkas audio pada modul yang dalam dokumentasi dinamai Bahasa Taluk dan Bahasa Minang. Objek analisis mencakup empat modul dan delapan konsep kata atau frasa. Tahapan penelitian meliputi inventarisasi kode dan berkas audio, identifikasi jalur masukan teks atau suara, analisis pencocokan string, pemutaran dan visualisasi WAV, serta identifikasi keterbatasan teknis dan linguistik. Pada jalur suara, ucapan dikenali melalui `speech_recognition` dengan `recognize_google` berkonfigurasi id-ID, kemudian transkripsi dicocokkan secara eksak melalui `if-elif`. Hasil menunjukkan bahwa prototipe dapat menjalankan alur masukan–pemetaan–audio–visualisasi, tetapi sensitif terhadap variasi ortografi seperti "sama sama"/"sama-sama" dan "dimana"/"di mana". Akar permasalahan terletak pada ketergantungan terhadap string eksak dan belum adanya validasi linguistik pasangan kosakata serta pelafalan. Literatur menempatkan Taluk Kuantan sebagai subdialek dalam Bahasa Melayu Riau Dialek Kuantan, sehingga perbandingan dengan bahasa Indonesia dan Minangkabau memerlukan validasi penutur atau ahli bahasa. Prototipe ini bermanfaat sebagai demonstrasi integrasi dan dasar pengembangan sumber belajar kosakata, bukan sebagai ukuran performa penerjemahan.

Kata kunci: bahasa daerah, pencocokan kosakata, pengenalan ucapan, Python, audio prerekaman

Abstract

Regional-language diversity motivates the development of digital aids for introducing regional vocabulary and audio. This study applies an implementation-oriented descriptive approach to analyze a Python prototype that matches Indonesian vocabulary with audio files in modules labeled "Bahasa Taluk" and "Bahasa Minang" in the documentation. The analysis covers four modules and eight word or phrase concepts. The procedure includes code and audio-file inventory, identification of text- and speech-input paths, analysis of exact string matching, WAV playback and visualization, and identification of technical and linguistic limitations. For speech input, `speech_recognition` with `recognize_google` configured for Indonesian (id-ID) converts speech to text before exact `if-elif` matching. The documented results show that the prototype can execute the input–mapping–audio–visualization flow, but it is sensitive to orthographic variants such as "sama sama"/"sama-sama" and "dimana"/"di mana". The core problem is the dependence on exact strings and the absence of linguistic validation of vocabulary pairs and pronunciation. Literature places Taluk Kuantan as a subdialect within the Riau Malay Kuantan dialect; therefore, comparisons with Indonesian and Minangkabau require validation by native speakers or language experts. The prototype is positioned as an integration demonstration and a baseline for vocabulary-learning resources rather than a translation-performance measure.

Keywords: regional language, vocabulary matching, speech recognition, Python, prerecorded audio

1. Pendahuluan

Bahasa daerah merupakan bagian dari keragaman linguistik Indonesia dan dapat didukung melalui sumber daya digital untuk membantu dokumentasi serta pengenalan kosakata. Handoko [1] membahas pengembangan korpus bahasa Minangkabau beserta tantangan dan arah pengembangannya, sedangkan Sujaini dan Putra [2] menunjukkan pemanfaatan algoritme komputasi untuk identifikasi bahasa daerah Indonesia.

Kedua kajian tersebut menunjukkan bahwa pengembangan teknologi bahasa daerah memerlukan dokumentasi, representasi bahasa, dan mekanisme pemrosesan yang sesuai dengan tujuan sistem.

Perkembangan pemrosesan ucapan semakin banyak memanfaatkan pembelajaran mendalam. Mehrish dkk. [3] mengulas pemanfaatan jaringan saraf, transformer, serta berbagai representasi pembelajaran dalam pengenalan dan

sintesis ucapan. Pada konteks bahasa Indonesia dan bahasa daerah, Arisaputra dkk. [4] mengembangkan sistem *automatic speech recognition* (ASR) berbasis XLS-R untuk bahasa Indonesia, Jawa, dan Sunda. Pratama dan Amrullah [5] melakukan *fine-tuning* model Whisper pada bahasa Jawa dan mengevaluasinya menggunakan *word error rate* (WER). Selain pemrosesan ucapan, pendekatan berbasis jaringan saraf juga telah diterapkan pada penerjemahan bahasa daerah. Sulistyio dkk. [6] mengembangkan penerjemahan Madura-Indonesia menggunakan *long short-term memory* (LSTM), sedangkan Ahda dkk. [7] membandingkan optimisasi Adam dan RMSprop pada penerjemahan dua arah Minangkabau-Indonesia berbasis *neural machine translation*.

Penelitian mengenai ASR bahasa daerah Indonesia juga berkembang melalui pemanfaatan model prelatih dan pembelajaran mandiri. Sakti dan Titalim [8] mengevaluasi model *self-supervised* pada kumpulan data bahasa etnik Indonesia untuk tugas ASR dengan sumber daya terbatas. Cryssiover dan Zahra [9] menggunakan Wav2Vec 2.0 untuk pengenalan bahasa Sunda. Sementara itu, Jarin dkk. [10] mengembangkan sistem ASR untuk pendiktean medis berbahasa Indonesia dengan korpus dan proses evaluasi yang terstruktur. Penelitian yang lebih baru menunjukkan penggunaan augmentasi data pada Speech-Transformer untuk pengenalan dialek Indonesia [11] serta pemanfaatan arsitektur multimodal Transformer-LSTM untuk penyelarasan leksikal lintas bahasa daerah Indonesia [12].

Pada skala multibahasa, model prelatih telah memperluas cakupan bahasa yang dapat diproses oleh sistem ucapan modern. Radford dkk. [13] mengembangkan Whisper melalui pelatihan berskala besar menggunakan data ucapan multibahasa. Pratap dkk. [14] mengembangkan Massively Multilingual Speech yang mencakup lebih dari seribu bahasa untuk beberapa tugas pemrosesan ucapan. Pengembangan lebih lanjut melalui SEAMLESSM4T mengintegrasikan pengenalan ucapan dan penerjemahan teks maupun suara dalam satu kerangka multibahasa [15]. Pendekatan-pendekatan tersebut menunjukkan arah perkembangan teknologi menuju model berbasis korpus besar, pelatihan model, dan evaluasi menggunakan metrik kuantitatif.

Berbeda dengan pendekatan berbasis model, implementasi pada penelitian ini tidak membangun atau melatih model pengenalan ucapan, sintesis ucapan, maupun penerjemahan mesin. Kode menggunakan pencocokan *string* terhadap kondisi yang ditentukan dalam percabangan if-elif. Masukan suara terlebih dahulu diubah menjadi teks menggunakan *recognize_google* dari pustaka *speech_recognition* dengan konfigurasi bahasa Indonesia (id-ID), kemudian hasilnya digunakan untuk memilih berkas audio WAV yang telah tersedia.

Dalam literatur yang ditinjau pada artikel ini, pembahasan didominasi pendekatan berbasis korpus dan pembelajaran dengan evaluasi kuantitatif, sedangkan prototipe

pencocokan kosakata tetap yang mengintegrasikan masukan teks atau suara, pemutaran audio prerekaman, dan visualisasi sinyal tidak menjadi fokus utama. Dengan demikian, gap yang dibahas pada artikel ini bersifat implementatif. Kontribusi dan kebaruan yang diklaim dibatasi pada analisis integrasi komponen tersebut dalam satu prototipe Python, bukan pada pengembangan model pengenalan ucapan, sintesis ucapan, atau penerjemahan mesin baru. Pembatasan klaim ini diperlukan agar hasil dibahas secara proporsional sesuai data yang terdokumentasi.

Akar permasalahan pada prototipe ini bukan hanya bagaimana memutar audio setelah pengguna memberikan masukan, tetapi bagaimana menjembatani bentuk masukan bahasa Indonesia yang relatif terstandar dengan variasi leksikal, ortografis, dan pelafalan pada bahasa daerah. Jalur suara menggunakan pengenalan berkonfigurasi id-ID, sedangkan keputusan keluaran ditentukan oleh kecocokan string yang ditulis secara manual. Konsekuensinya, kesalahan transkripsi, perbedaan spasi atau tanda hubung, serta perbedaan bentuk kosakata daerah dapat menghasilkan kegagalan keluaran meskipun maksud pengguna sebenarnya sama. Kondisi ini menjadi alasan mengapa hasil perlu dibahas lebih jauh pada dua lapis, yaitu lapis teknis berupa reliabilitas alur pemrosesan dan lapis linguistik berupa ketepatan pasangan kosakata serta pelafalan.

Konteks linguistik juga perlu diperjelas agar istilah pada modul tidak langsung dianggap sebagai klasifikasi bahasa yang telah tervalidasi. Penelitian tentang Melayu Riau di Kuantan Singingi menyebut Taluk Kuantan sebagai salah satu subdialek dalam Bahasa Melayu Riau Dialek Kuantan [16]. Sementara itu, kajian tata bahasa Minangkabau menunjukkan kedekatan Minangkabau dengan bahasa Indonesia pada unsur kosakata, morfem, dan sintaksis [17]. Kedekatan tersebut memungkinkan munculnya bentuk-bentuk yang serupa, tetapi tidak berarti setiap kata mempunyai bentuk, pelafalan, fungsi sosial, atau makna pragmatis yang identik. Karena itu, prototipe yang memuat konsep seperti "saya", "kami", dan "kamu" memerlukan perhatian khusus terhadap variasi pronomina dan konteks penggunaannya.

Penelitian ini bertujuan mengimplementasikan dan menganalisis prototipe pencocokan kosakata bahasa Indonesia dengan keluaran audio pada modul yang dalam laporan sumber dinamai Bahasa Taluk dan Bahasa Minang. Analisis difokuskan pada arsitektur program, mekanisme pencocokan string, jalur masukan teks dan suara, pemutaran serta visualisasi audio, akar permasalahan implementasi, dan aspek linguistik yang masih memerlukan validasi. Ruang lingkup artikel dibatasi pada arah masukan bahasa Indonesia menuju keluaran audio pada modul bahasa daerah karena kondisi input pada kode ditulis dalam bahasa Indonesia dan jalur pengenalan suara menggunakan konfigurasi id-ID. Arah sebaliknya tidak ditunjukkan oleh implementasi yang terdokumentasi.

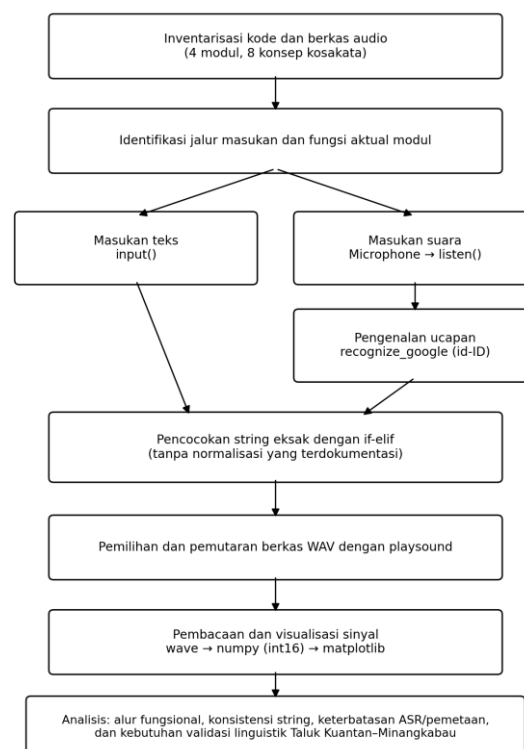
Secara praktis, penelitian ini diharapkan memberi manfaat sebagai bukti konsep alat bantu pengenalan kosakata dan audio daerah; secara teknis, sebagai dasar untuk merancang normalisasi masukan, struktur pemetaan yang lebih terpusat, dan protokol pengujian; serta secara ilmiah, sebagai penunjuk aspek kebahasaan Taluk Kuantan dan Minangkabau yang perlu dikaji lebih lanjut sebelum sistem digunakan sebagai sumber belajar atau penerjemahan.

2. Metode Penelitian

Penelitian ini menggunakan metode implementatif-deskriptif. Objek penelitian adalah empat modul Python yang pada dokumentasi diberi nama "Text to Speech Bahasa Taluk", "Text to Speech Bahasa Minang", "Speech to Speech Bahasa Taluk", dan "Speech to Speech Bahasa Minang", beserta berkas audio WAV yang dipanggil oleh kode. Unit analisis terdiri atas delapan konsep kata atau frasa, yaitu "saya", "kami", "kamu", "terima kasih", "sama-sama", "dari mana", "di mana", dan "sejak kapan". Implementasi ditelaah berdasarkan fungsi aktual kode, bukan hanya nama modul. Lingkungan pengembangan menggunakan Python pada Visual Studio Code dengan pustaka wave, playsound, speech_recognition, numpy, dan matplotlib. Karena dokumentasi tidak memuat jumlah penutur, pengulangan percobaan, transkrip referensi, atau validasi ahli bahasa, metode penelitian difokuskan pada analisis alur fungsional dan karakteristik implementasi, bukan eksperimen performa kuantitatif.

2.1. Rancangan dan Tahapan Penelitian

Tahapan penelitian disusun sebagai berikut. Pertama, dilakukan inventarisasi empat modul program, delapan konsep masukan, serta kelompok berkas WAV yang dipanggil. Kedua, setiap modul ditelusuri untuk mengidentifikasi jenis masukan aktual, yaitu input teks melalui `input()` atau input suara melalui `Microphone` dan `listen()`. Ketiga, alur pemrosesan ditelusuri sampai tahap transkripsi `recognize_google`, pencocokan `if-elif`, pemilihan berkas audio, pemutaran `playsound`, pembacaan WAV, konversi `numpy`, dan visualisasi `matplotlib`. Keempat, keluaran yang terdokumentasi dianalisis untuk memisahkan keberhasilan pengenalan ucapan, keberhasilan pemetaan string, dan keberhasilan pemrosesan audio. Kelima, keterbatasan teknis dan kebutuhan validasi linguistik diidentifikasi dengan membandingkan perilaku kode dengan literatur terkait bahasa daerah dan ASR. Diagram blok metode penelitian dan alur pemrosesan prototype adalah seperti yang ditunjukkan pada Gambar 1



Gambar 1. Diagram blok metode penelitian dan alur pemrosesan prototype.

2.2. Arsitektur dan Jalur Masukan

Pada jalur masukan teks yang terdokumentasi pada modul "Text to Speech Bahasa Taluk", pengguna memasukkan kata atau frasa melalui fungsi `input()`. Nilai masukan dibandingkan langsung dengan kondisi pada percabangan `if-elif`. Jika terdapat kecocokan, program memilih berkas WAV yang dikaitkan dengan kondisi tersebut, memutarnya menggunakan `playsound`, kemudian membaca berkas untuk keperluan visualisasi.

Pada jalur masukan suara, `speech_recognition` digunakan untuk menginisialisasi `Recognizer` dan `Microphone`. Ucapan direkam melalui `listen()` dengan parameter `phrase_time_limit=2`, lalu dikenali menggunakan `recognize_google` dengan parameter `language='id-ID'`. Teks hasil pengenalan selanjutnya dibandingkan dengan kondisi `if-elif` untuk menentukan berkas audio yang diputar. Secara metodologis, tahap pengenalan ucapan dan tahap pencocokan dipisahkan karena kegagalan pada tahap pertama tidak identik dengan kegagalan pada tahap kedua.

Salah satu modul diberi label "Text to Speech Bahasa Minang", tetapi kode pada modul tersebut menggunakan mikrofon dan pengenalan ucapan, bukan input teks. Oleh karena itu, pengelompokan analisis didasarkan pada fungsi aktual yang dijalankan. Pendekatan ini mencegah nama modul digunakan untuk menyimpulkan fungsi `text-to-speech`, `speech-to-speech`, atau arah penerjemahan yang tidak benar-benar ditunjukkan oleh kode.

2.3. Pencocokan Kosakata dan Audio

Secara konseptual program memuat delapan kata atau frasa, yaitu "saya", "kami", "kamu", "terima kasih", "sama-sama", "dari mana", "di mana", dan "sejak kapan". Pencocokan dilakukan secara eksak melalui if-elif. Pada jalur teks, dua kondisi ditulis sebagai "sama sama" dan "dimana", sedangkan pada jalur suara ditulis sebagai "sama-sama" dan "di mana". Perbedaan bentuk *string* ini dipertahankan sebagai karakteristik implementasi dan dibahas pada hasil.

Kode tidak mendokumentasikan tahap normalisasi seperti perubahan huruf besar-kecil, penghilangan spasi berlebih, atau penyamaan tanda hubung sebelum pencocokan. Akibatnya, masukan yang secara makna sama tetapi berbeda bentuk penulisan berpotensi tidak memenuhi kondisi yang sama. Sistem juga tidak membentuk keluaran baru untuk kosakata yang tidak tersedia; penambahan entri memerlukan penambahan kondisi dan berkas audio.

2.4. Pengenalan dan Pemrosesan Audio

Pada masukan suara, program merekam ucapan melalui mikrofon dan mengubahnya menjadi teks menggunakan `recognize_google`. Konfigurasi id-ID menunjukkan bahwa jalur pengenalan diarahkan pada ucapan bahasa Indonesia. Kode juga memuat blok try-except untuk menangani kondisi ketika proses pengenalan tidak menghasilkan teks yang dapat digunakan.

Jika teks hasil pengenalan sama dengan salah satu kondisi yang tersedia, berkas audio yang terkait diputar menggunakan `playsound`. Dengan demikian, jalur suara terdiri atas dua tahap yang secara logis terpisah: pengenalan ucapan menjadi teks dan pencocokan teks terhadap berkas audio. Pemisahan ini penting karena kesalahan pada pengenalan ucapan tidak sama dengan kesalahan pada pemetaan kosakata.

Setelah berkas audio ditentukan, pustaka `wave` digunakan untuk membuka WAV dan memperoleh frekuensi sampel melalui `getframerate()`, jumlah frame melalui `getnframes()`, jumlah kanal melalui `getnchannels()`, serta data audio melalui `readframes()`.

Data audio dikonversi menjadi larik `int16` menggunakan `numpy.frombuffer()`. Dokumentasi kode kemudian membentuk dua deret sampel melalui `signal_array[0:2]` dan `signal_array[1:2]`, membuat vektor waktu dari jumlah frame dan frekuensi sampel, serta memplot amplitudo terhadap waktu menggunakan `matplotlib`. Nilai frekuensi sampel, jumlah kanal aktual, dan parameter kualitas rekaman tidak dilaporkan sebagai hasil penelitian.

Implementasi menggunakan Python dengan pustaka `wave`, `playsound`, `speech_recognition`, `numpy`, dan `matplotlib`. Penelitian yang terdokumentasi belum memuat protokol pengujian berulang, jumlah penutur, jumlah percobaan, nilai akurasi, word error rate (WER), waktu respons, atau prosedur validasi linguistik. Oleh karena itu, pembahasan

dibatasi pada demonstrasi fungsi yang dapat ditelusuri dari kode dan contoh keluaran yang tersedia.

2.5. Kriteria Analisis

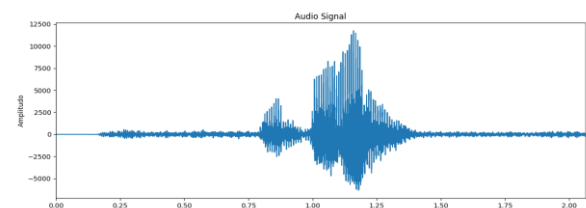
Analisis dilakukan menggunakan empat kriteria. Pertama, keterlacakan alur fungsional dari masukan sampai keluaran audio dan visualisasi. Kedua, konsistensi pemetaan, terutama kesesuaian bentuk *string* antarjalur dan keberadaan normalisasi masukan. Ketiga, pemisahan sumber kegagalan antara pengenalan ucapan dan pemetaan kosakata. Keempat, kecukupan informasi linguistik, yaitu apakah pasangan kosakata, pelafalan, dan penamaan varietas bahasa telah didukung sumber atau validasi penutur. Kriteria tersebut dipilih agar manfaat prototipe dapat dijelaskan tanpa melebihi kemampuan sistem yang belum diuji secara kuantitatif.

3. Hasil dan Analisis

3.1. Implementasi Jalur Teks dan Suara

Pada jalur teks, fungsi `input()` menerima kata atau frasa yang kemudian dibandingkan secara eksak dengan kondisi if-elif. Jika kondisi terpenuhi, `playsound` memutar berkas audio yang dipetakan dan kode selanjutnya membaca WAV untuk membentuk visualisasi sinyal.

Hasil yang terdokumentasi memperlihatkan contoh keluaran berupa audio dan grafik amplitudo terhadap waktu setelah masukan teks sesuai dengan salah satu kondisi program. Gambar 2 menampilkan bentuk gelombang berkas audio yang dipanggil. Contoh ini menunjukkan bahwa rangkaian input teks, pemilihan berkas, pembacaan audio, dan visualisasi dapat dijalankan pada implementasi yang ditampilkan, tetapi tidak merupakan pengujian performa terstruktur.



Gambar 2. Visualisasi amplitudo terhadap waktu pada keluaran berbasis masukan teks.

Gambar 2 menunjukkan bahwa berkas WAV yang dipanggil dapat dibaca dan direpresentasikan sebagai sinyal domain waktu. Namun, grafik amplitudo tidak mengukur kesesuaian makna, ketepatan pelafalan, maupun akurasi pemetaan. Karena itu, grafik digunakan sebagai bukti keluaran fungsional pemrosesan audio, bukan sebagai metrik keberhasilan penerjemahan.

Pada jalur teks ditemukan ketergantungan terhadap bentuk *string*. Kondisi untuk konsep "sama-sama" ditulis sebagai "sama sama", sedangkan konsep "di mana" ditulis sebagai "dimana". Karena pencocokan dilakukan secara langsung tanpa normalisasi yang terdokumentasi, variasi tanda hubung atau spasi dapat memengaruhi terpenuhinya kondisi. Temuan ini merupakan keterbatasan logika

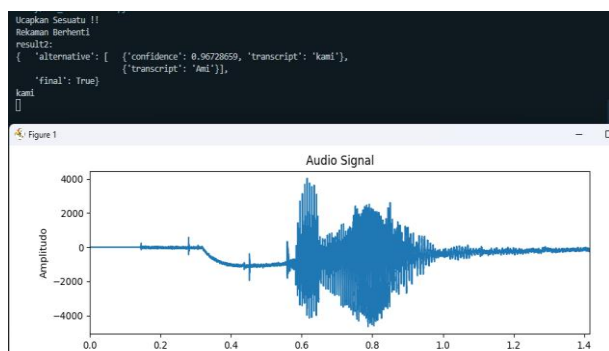
pencocokan dan tidak dapat dinilai tingkat kejadiannya karena pengujian berulang tidak tersedia.

Walaupun dokumentasi awal menggunakan istilah *text-to-speech*, kode tidak membangkitkan suara baru dari teks melalui model akustik atau *vocoder* seperti pada TTS modern [3]. Program memilih dan memutar rekaman yang telah tersedia. Secara fungsional, mekanisme pada jalur ini lebih tepat disebut pencocokan teks-ke-audio prerekaman.

Pada jalur masukan suara, program menerima ucapan melalui mikrofon dengan batas `phrase_time_limit=2`. Fungsi `recognize_google` berkonfigurasi `id-ID` menghasilkan representasi teks yang kemudian digunakan sebagai masukan ke struktur *if-elif*.

Setelah transkripsi tersedia, program membandingkannya dengan kosakata yang telah ditentukan. Jika teks sesuai, berkas audio terkait diputar. Dengan mekanisme ini, keberhasilan keluaran akhir bergantung secara berurutan pada keberhasilan pengenalan ucapan dan kesesuaian hasil transkripsi dengan *string* pada kondisi program.

Dokumentasi hasil menampilkan satu contoh keluaran jalur suara berupa teks hasil pengenalan pada konsol dan grafik amplitudo audio. Pada contoh Gambar 3, konsol menampilkan hasil pengenalan "kami" sebelum visualisasi sinyal ditampilkan. Contoh tersebut menunjukkan alur fungsional dari masukan suara menuju transkripsi, pemetaan, dan keluaran audio, tetapi tidak mewakili ukuran akurasi karena jumlah pengulangan dan variasi penutur tidak dilaporkan.



Gambar 3. Tampilan hasil pengenalan masukan suara dan visualisasi sinyal audio keluaran.

Gambar 3 menegaskan adanya dua tahap berbeda. Tahap pertama adalah pengenalan ucapan bahasa Indonesia menjadi teks, sedangkan tahap kedua adalah pencocokan teks dengan berkas audio. Jika transkripsi berbeda dari seluruh kondisi yang tersedia, berkas target tidak dapat dipilih melalui struktur *if-elif*. Pada implementasi ini tidak terdapat data yang memungkinkan probabilitas kegagalan pada masing-masing tahap dihitung secara terpisah.

Karena pengenalan suara memanfaatkan `recognize_google`, penelitian ini tidak mengembangkan model ASR baru. Hal tersebut berbeda dengan Arisaputra dkk. [4], Pratama dan Amrullah [5], Sakti dan Titalim [8], serta Cryssiover dan Zahra [9] yang mengevaluasi model ASR pada data tertentu. Literatur tersebut digunakan

sebagai pembandingan metodologis; nilai performanya tidak dibandingkan langsung dengan prototipe ini karena data pengujian kuantitatif tidak tersedia.

3.2. Pemetaan Kosakata dan Analisis Sinyal

Delapan konsep masukan dihubungkan dengan berkas audio tertentu melalui percabangan *if-elif*. Tabel 1 merangkum empat modul sebagaimana dinamai pada laporan sumber serta jenis masukan yang tampak dari kode. Nama modul dipertahankan sesuai dokumentasi, tetapi fungsi aktual ditentukan berdasarkan perintah yang dijalankan.

Tabel 1. Struktur empat modul dan berkas audio yang terdokumentasi pada program.

Modul pada laporan	Masukan menurut kode	Berkas audio yang dipanggil
Text to Speech Bahasa Taluk	Teks	Kelompok generik: saya.wav, kami.wav, ... , sejjakkapan.wav
Text to Speech Bahasa Minang	Suara	Kelompok generik yang sama
Speech to Speech Bahasa Taluk	Suara	Kelompok generik yang sama
Speech to Speech Bahasa Minang	Suara	aden.wav, Awak.wav, ... , sajak bilo.wav

Tabel 1 menunjukkan bahwa modul "Text to Speech Bahasa Taluk" menggunakan masukan teks, sedangkan modul "Text to Speech Bahasa Minang" justru menggunakan mikrofon dan pengenalan ucapan. Modul "Speech to Speech Bahasa Taluk" juga menggunakan masukan suara dengan kelompok berkas generik seperti `saya.wav` dan `kami.wav`, sementara modul "Speech to Speech Bahasa Minang" memanggil berkas seperti `aden.wav`, `Awak.wav`, dan `dima.wav`. Struktur ini merupakan representasi kode, bukan tabel akurasi atau kamus bahasa yang telah divalidasi. Tanpa data validasi penutur atau ahli bahasa, artikel tidak menetapkan ketepatan arti maupun pelafalan berdasarkan nama berkas tersebut.

Selain keterbatasan jumlah entri, implementasi menunjukkan ketidakkonsistenan bentuk *string* antarjalur. Jalur teks menggunakan "sama sama" dan "dimana", sedangkan jalur suara menggunakan "sama-sama" dan "di mana". Pencocokan eksak tanpa tahap normalisasi yang terdokumentasi membuat program sensitif terhadap perbedaan ortografi sederhana. Perbaikan teknis paling langsung adalah menormalkan input sebelum pencocokan atau menyimpan pasangan kosakata dalam struktur data yang lebih terpusat.

Struktur *if-elif* menghasilkan pemetaan deterministik dan mudah ditelusuri, tetapi tidak mempunyai kemampuan generalisasi. Karakteristik ini berbeda dengan penerjemahan berbasis LSTM atau *neural machine translation* yang mempelajari hubungan dari pasangan data bahasa [6], [7]. Karena itu, prototipe ini ditempatkan sebagai sistem pencocokan kosakata tetap dan bukan sebagai penerjemahan mesin.

Visualisasi dimulai setelah berkas audio keluaran ditentukan. Pustaka wave membaca WAV, `getframerate()` memperoleh frekuensi sampel, `getnframes()` memperoleh jumlah frame, dan `getnchannels()` memperoleh jumlah kanal. Data dibaca melalui `readframes()`, dikonversi menjadi larik `int16`, kemudian diplot sebagai amplitudo terhadap waktu.

Gambar 2 dan Gambar 3 memperlihatkan bentuk gelombang audio keluaran pada domain waktu. Visualisasi tersebut menunjukkan bahwa data WAV dapat dibaca dan ditampilkan sebagai sinyal, tetapi tidak memberikan informasi langsung mengenai ketepatan arti kosakata, kualitas pelafalan, ataupun keberhasilan pengenalan ucapan.

Kode membentuk dua deret melalui `signal_array[0::2]` dan `signal_array[1::2]`, sementara jumlah kanal dibaca melalui `getnchannels()`. Namun, dokumentasi tidak menunjukkan percabangan pengolahan berdasarkan nilai `n_channels` dan tidak melaporkan apakah berkas yang digunakan mono atau stereo. Oleh karena itu, dua deret tersebut tidak dapat dipastikan sebagai kanal stereo berdasarkan dokumentasi yang tersedia. Nilai frekuensi sampel aktual, jenis mikrofon, kondisi perekaman, dan tingkat derau juga tidak tersedia untuk analisis kualitas sinyal yang lebih rinci.

Dengan keterbatasan tersebut, bentuk gelombang diposisikan sebagai keluaran pendukung, bukan metrik performa. Pada penelitian ASR, WER digunakan untuk mengevaluasi kesalahan transkripsi, misalnya pada Whisper bahasa Jawa [5] dan Wav2Vec 2.0 bahasa Sunda [9]. Prototipe ini belum mempunyai transkrip referensi dan rangkaian pengujian yang diperlukan untuk menghitung metrik serupa.

3.3. Akar Permasalahan dan Kajian Bahasa Indonesia–Taluk Kuantan–Minangkabau

Hasil implementasi menunjukkan bahwa akar permasalahan sistem berada pada titik temu antara dua jenis variasi. Variasi pertama bersifat teknis, yaitu hasil transkripsi ASR dan penulisan string yang harus sama persis dengan kondisi `if-elif`. Variasi kedua bersifat linguistik, yaitu kemungkinan satu konsep mempunyai bentuk leksikal, pelafalan, atau penggunaan sosial yang berbeda pada bahasa Indonesia, varietas Kuantan, dan Minangkabau. Karena program tidak mempunyai lapisan normalisasi maupun kamus terstruktur, kedua variasi tersebut bercampur dalam satu keputusan biner: cocok atau tidak cocok. Akibatnya, kegagalan keluaran tidak dapat langsung ditafsirkan sebagai kesalahan bahasa, karena sumbernya dapat berupa ASR, ortografi, pemetaan, atau ketidaktepatan pasangan kosakata.

Dari sisi status kebahasaan, istilah "Bahasa Taluk" pada nama modul perlu diperlakukan sebagai label implementasi. Literatur mengenai Melayu Riau di Kuantan Singingi menempatkan Taluk Kuantan sebagai salah satu subdialek Bahasa Melayu Riau Dialek Kuantan [16]. Informasi ini penting karena unit yang dibandingkan tidak sepenuhnya sejajar: bahasa Indonesia merupakan bahasa standar nasional, Minangkabau merupakan bahasa daerah

dengan sistem gramatikal yang terdokumentasi, sedangkan Taluk Kuantan dapat diposisikan sebagai subdialek regional dalam kontinum Melayu Kuantan. Dengan demikian, penelitian lanjutan sebaiknya menggunakan istilah dan batas varietas yang telah diverifikasi oleh ahli bahasa atau penutur setempat.

Pada tingkat persamaan, Minangkabau memiliki kedekatan dengan bahasa Indonesia pada kosakata, morfem, dan sintaksis [17], sehingga beberapa konsep dasar dapat memiliki hubungan bentuk dan makna yang mudah dikenali. Pada tingkat perbedaan, varietas Kuantan memperlihatkan variasi internal pada pronomina dan penggunaannya; penelitian di Subdialek Baserah, misalnya, menunjukkan bahwa pilihan pronomina dapat dipengaruhi usia, status sosial, dan relasi penutur [16]. Temuan tersebut relevan bagi delapan konsep pada prototipe karena tiga di antaranya adalah pronomina atau rujukan persona ("saya", "kami", dan "kamu"). Artinya, pencocokan satu-ke-satu berdasarkan satu berkas audio berpotensi menyederhanakan variasi yang secara linguistik lebih kompleks.

Berdasarkan artefak program yang tersedia, perbedaan leksikal antarmodul belum dapat dianalisis secara penuh. Tiga modul memanggil kelompok nama berkas generik seperti `saya.wav` dan `kami.wav`, sedangkan modul "Speech to Speech Bahasa Minang" memanggil nama berkas seperti `aden.wav`, `Awak.wav`, dan `dima.wav`. Nama berkas tersebut menunjukkan adanya upaya membedakan keluaran, tetapi tidak cukup untuk membuktikan padanan makna atau pelafalan karena transkripsi isi audio, identitas penutur, dan prosedur validasi tidak terdokumentasi. Oleh sebab itu, pembahasan ini membedakan dengan tegas bukti implementasi dari klaim linguistik.

Kajian lebih mendalam yang paling relevan adalah membangun daftar paralel bahasa Indonesia–Subdialek Taluk Kuantan–Minangkabau untuk konsep yang sama, merekam variasi dari beberapa penutur, memberi anotasi bentuk tulis dan pelafalan, serta meminta validasi makna dan konteks pemakaian oleh penutur atau ahli bahasa. Data tersebut kemudian dapat digunakan untuk menilai apakah kemiripan antarbentuk membantu proses pencocokan atau justru menimbulkan ambiguitas. Bagi prototipe, manfaat langsungnya adalah pemetaan kosakata yang lebih dapat dipertanggungjawabkan, pengurangan kegagalan akibat variasi bentuk, dan peluang pengembangan dari pemutar audio berbasis aturan menuju sumber belajar bahasa daerah yang tervalidasi.

3.4. Posisi Sistem, Keterbatasan, dan Arah Pengembangan

Perbedaan utama prototipe ini dengan penelitian pemrosesan ucapan terkini terletak pada mekanisme pembelajaran dan kedalaman evaluasi. XLS-R, Whisper, dan Wav2Vec 2.0 memanfaatkan model yang dilatih pada data berskala besar dan dapat disesuaikan pada bahasa sasaran [4], [5], [9], sedangkan implementasi ini bekerja

melalui kondisi string yang ditetapkan secara manual. Sakti dan Titalim [8] juga menunjukkan pendekatan pembelajaran mandiri untuk ASR bahasa etnik Indonesia. Setelah akar permasalahan teknis dan linguistik pada Subbagian 3.3 dipisahkan, posisi prototipe menjadi lebih jelas: sistem ini adalah pemilih audio berbasis aturan dengan masukan yang sebagian bergantung pada ASR bahasa Indonesia, bukan model pengenalan atau penerjemah bahasa daerah.

Karena penelitian berbasis model menggunakan korpus dan protokol evaluasi, kinerjanya dapat dinilai menggunakan WER, CER, SER, atau metrik lain [10], [11]. Pada prototipe ini, data yang diperlukan untuk evaluasi tersebut tidak terdokumentasi. Dengan demikian, literatur ASR digunakan untuk menjelaskan perbedaan metodologi evaluasi, bukan untuk mengklaim kesetaraan performa.

Pada sisi penerjemahan, LSTM dan *neural machine translation* membangun pemetaan dari data pasangan bahasa sehingga dapat menangani masukan di luar daftar tetap dengan mekanisme inferensi [6], [7]. Utami dkk. [12] menggunakan pendekatan multimodal untuk penyelarasan leksikal lintas bahasa daerah. Prototipe pada penelitian ini tidak memiliki proses pembelajaran tersebut dan terbatas pada pasangan kondisi-berkas yang telah diprogram.

Model multibahasa berskala besar seperti Whisper [13], Massively Multilingual Speech [14], dan SEAMLESSM4T [15] menunjukkan arah sistem yang mendukung bahasa dan tugas dalam skala luas. Perbandingan langsung berdasarkan angka performa dengan prototipe delapan konsep kosakata tidak tepat karena ruang lingkup, data, dan metode evaluasinya berbeda.

Oleh karena itu, kontribusi yang dapat dipertanggungjawabkan pada penelitian ini adalah demonstrasi integrasi komponen dan analisis karakteristik implementasi. Klaim mengenai efisiensi, akurasi, keberhasilan penerjemahan, atau keunggulan terhadap metode lain tidak diberikan karena tidak didukung data kuantitatif.

Keterbatasan pertama adalah cakupan hanya delapan konsep kata atau frasa dan penggunaan pencocokan *string* eksak. Variasi penulisan antarjalur menunjukkan bahwa sistem sensitif terhadap spasi dan tanda hubung. Masukan di luar kondisi yang telah ditulis tidak mempunyai keluaran tanpa perubahan kode dan penambahan berkas audio.

Keterbatasan kedua adalah belum tersedianya evaluasi terstruktur pada jalur suara. Penelitian tidak mendokumentasikan jumlah penutur, jumlah pengulangan, akurasi transkripsi, tingkat keberhasilan pemetaan, maupun waktu respons. Karena itu, kesalahan pada tahap pengenalan ucapan dan tahap pencocokan belum dapat dipisahkan secara kuantitatif.

Keterbatasan ketiga berkaitan dengan aspek linguistik. Dokumentasi tidak menjelaskan sumber pasangan kosakata, jumlah atau kriteria penutur perekam, prosedur pemeriksaan arti dan pelafalan, maupun status linguistik istilah Taluk. Karena itu, artikel mengikuti penamaan modul program tetapi tidak mengklaim validitas linguistik keluaran tanpa validasi tambahan.

Keterbatasan keempat adalah ruang lingkup arah bahasa. Implementasi yang terdokumentasi menggunakan kondisi masukan bahasa Indonesia dan *recognize_google* dengan konfigurasi *id-ID*, sehingga artikel hanya membahas bahasa Indonesia sebagai masukan menuju audio bahasa daerah. Fungsi arah sebaliknya tidak ditunjukkan oleh kode yang tersedia dan karena itu tidak diklaim.

Pengembangan paling penting adalah menyusun protokol pengujian terstruktur. Setiap konsep perlu diuji berulang dengan beberapa penutur dan kondisi yang ditentukan, kemudian keberhasilan pengenalan ucapan dan keberhasilan pemetaan dicatat secara terpisah. Dari data tersebut dapat dihitung tingkat keberhasilan per tahap, tingkat keberhasilan keseluruhan (*end-to-end*), pola kesalahan, dan waktu respons.

Untuk jalur pengenalan ucapan, evaluasi dapat menggunakan akurasi transkripsi pada perintah pendek; WER atau CER dapat ditambahkan apabila data uji diperluas menjadi korpus yang memadai. Pemisahan evaluasi ASR dan pemetaan akan membantu menentukan sumber kegagalan serta memungkinkan perbandingan yang lebih objektif dengan penelitian [4], [5], [8], [9], [11].

Pada sisi bahasa daerah, pasangan kosakata dan rekaman audio perlu divalidasi oleh penutur atau ahli bahasa. Dokumentasi sumber kosakata, penutur, prosedur perekaman, dan prosedur validasi perlu disertakan. Pengembangan sumber daya yang lebih sistematis, seperti arah pengembangan korpus Minangkabau yang dibahas Handoko [1], dapat mendukung perluasan prototipe.

Secara teknis, struktur *if-elif* dapat digantikan dengan basis data atau struktur *dictionary* Python yang memusatkan pasangan input dan berkas audio, disertai normalisasi *string* sebelum pencocokan. Jika di masa mendatang tersedia korpus paralel yang memadai, sistem dapat dikembangkan menuju pendekatan berbasis model seperti [6], [7], [12] atau memanfaatkan model multibahasa [13]-[15].

4. Kesimpulan

Penelitian ini mengimplementasikan dan menganalisis prototipe pencocokan kosakata bahasa Indonesia dengan berkas audio pada modul yang dalam dokumentasi dinamai Bahasa Taluk dan Bahasa Minang berbasis Python. Metode implementatif-deskriptif menelusuri empat modul dan delapan konsep melalui jalur masukan teks atau suara, pencocokan *if-elif*, pemutaran WAV, serta visualisasi sinyal. Jalur suara menggunakan *speech_recognition* dengan *recognize_google* berkonfigurasi *id-ID*, sehingga pengenalan ucapan diarahkan pada bahasa Indonesia

sebelum hasil transkripsi dipetakan ke berkas audio. Hasil menunjukkan bahwa alur fungsional dapat dijalankan, tetapi sistem sensitif terhadap variasi string seperti "sama sama"/"sama-sama" dan "dimana"/"di mana". Akar permasalahan mencakup dua lapis, yaitu ketergantungan teknis pada transkripsi dan string eksak serta belum adanya validasi linguistik pasangan kosakata dan pelafalan. Literatur menempatkan Taluk Kuantan sebagai subdialek Bahasa Melayu Riau Dialek Kuantan, sedangkan Minangkabau memiliki kedekatan struktural dengan bahasa Indonesia; namun persamaan atau perbedaan pada delapan entri prototipe belum dapat dinyatakan sebagai hasil linguistik tanpa data penutur dan validasi ahli. Manfaat prototipe terutama sebagai demonstrasi integrasi dan dasar pengembangan sumber belajar kosakata. Pengembangan berikutnya perlu memprioritaskan normalisasi input, struktur pemetaan terpusat, penyusunan daftar paralel Indonesia–Taluk Kuantan–Minangkabau, validasi multi-penutur atau ahli bahasa, serta pengujian terstruktur dengan metrik terpisah untuk ASR dan pemetaan.

Daftar Pustaka

- [1] H. Handoko, "Developing the Corpus of Minangkabau Language: Insights, Challenges, and Future Directions," *Jurnal Arbitrer*, vol. 11, no. 3, pp. 413–429, 2024, doi: 10.25077/ar.11.3.413-429.2024.
- [2] H. Sujaini and A. B. Putra, "Analysis of Language Identification Algorithms for Regional Indonesian Languages," *IAES International Journal of Artificial Intelligence*, vol. 13, no. 2, pp. 1741–1752, 2024, doi: 10.11591/ijai.v13.i2.pp1741-1752.
- [3] A. Mehrish, N. Majumder, R. Bharadwaj, R. Mihalcea, and S. Poria, "A Review of Deep Learning Techniques for Speech Processing," *Information Fusion*, vol. 99, Art. no. 101869, 2023, doi: 10.1016/j.inffus.2023.101869.
- [4] P. Arisaputra, A. T. Handoyo, and A. Zahra, "XLS-R Deep Learning Model for Multilingual ASR on Low-Resource Languages: Indonesian, Javanese, and Sundanese," *ICIC Express Letters, Part B: Applications*, vol. 15, no. 6, pp. 551–559, 2024, doi: 10.24507/icicelb.15.06.551.
- [5] R. S. A. Pratama and A. Amrullah, "Analysis of Whisper Automatic Speech Recognition Performance on Low Resource Language," *Jurnal Pilar Nusa Mandiri*, vol. 20, no. 1, pp. 1–8, 2024, doi: 10.33480/pilar.v20i1.4633.
- [6] D. A. Sulisty, A. P. Wibawa, D. D. Prasetya, and F. A. Ahda, "LSTM-Based Machine Translation for Madurese-Indonesian," *Journal of Applied Data Sciences*, vol. 4, no. 3, pp. 189–199, 2023, doi: 10.47738/jads.v4i3.113.
- [7] F. A. Ahda, A. P. Wibawa, D. D. Prasetya, and D. A. Sulisty, "Comparison of Adam Optimization and RMSprop in Minangkabau-Indonesian Bidirectional Translation with Neural Machine Translation," *International Journal on Informatics Visualization*, vol. 8, no. 1, pp. 231–238, 2024, doi: 10.62527/joiv.8.1.1818.
- [8] S. Sakti and B. A. Titalim, "Leveraging the Multilingual Indonesian Ethnic Languages Dataset in Self-Supervised Models for Low-Resource ASR Task," in *2023 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, 2023, pp. 1–8, doi: 10.1109/ASRU57964.2023.10389730.
- [9] A. Cryssiover and A. Zahra, "Speech Recognition Model Design for Sundanese Language Using WAV2VEC 2.0," *International Journal of Speech Technology*, vol. 27, pp. 171–177, 2024, doi: 10.1007/s10772-023-10066-5.
- [10] A. Jarin, A. Santosa, M. T. Uliniansyah, L. R. Aini, E. Nurfadhilah, and Gunarso, "Automatic Speech Recognition for Indonesian Medical Dictation in Cloud Environment," *IAES International Journal of Artificial Intelligence*, vol. 13, no. 2, pp. 1762–1772, 2024, doi: 10.11591/ijai.v13.i2.pp1762-1772.
- [11] S. N. Endah, Suprpto, and Y. Suyanto, "Enhancing Low-Resource Dialectal ASR in Indonesian Using Speech-Transformer Models and Data Augmentation," *Engineering, Technology & Applied Science Research*, vol. 15, no. 5, pp. 28095–28101, 2025, doi: 10.48084/etasr.12734.
- [12] E. Utami et al., "A Multimodal Transformer-LSTM Framework for Cross-Lingual Lexical Alignment of Indonesian Regional Languages," *Engineering, Technology & Applied Science Research*, vol. 16, no. 2, pp. 34283–34292, 2026, doi: 10.48084/etasr.17086.
- [13] A. Radford, J. W. Kim, T. Xu, G. Brockman, C. McLeavey, and I. Sutskever, "Robust Speech Recognition via Large-Scale Weak Supervision," in *Proceedings of the 40th International Conference on Machine Learning*, PMLR, vol. 202, pp. 28492–28518, 2023.
- [14] V. Pratap et al., "Scaling Speech Technology to 1,000+ Languages," *Journal of Machine Learning Research*, vol. 25, no. 97, pp. 1–52, 2024.
- [15] SEAMLESS Communication Team, "Joint Speech and Text Machine Translation for up to 100 Languages," *Nature*, vol. 637, pp. 587–593, 2025, doi: 10.1038/s41586-024-08359-z.
- [16] Y. Ovtaroza, H. Faizah AR, dan N. Hakim, "Pronomina Persona Bahasa Melayu Riau Subdialek Baserah Kecamatan Kuantan Hilir Kabupaten Kuantan Singingi," *Jurnal Online Mahasiswa FKIP Universitas Riau*, 2015.
- [17] A. Ayub, N. Husin, Muhandi, A. H. Usman, dan A. Yasin, *Tata Bahasa Minangkabau*. Jakarta: Pusat Pembinaan dan Pengembangan Bahasa, Departemen Pendidikan dan Kebudayaan, 1993.