

Implementasi Algoritma Deep Learning Pada Aplikasi Penerjemah Suara Otomatis Indonesia-Jepang Online

Vega Fajar Habibi¹, Khairuddin Nasution², Aulia Ichsan³

Fakultas Teknik Universitas Islam Sumatera Utara^{1,2,3}

Email vegafajrhabibi@gmail.com

Informasi

Volume : 2
Nomor : 7
Bulan : Juli
Tahun : 2025
E-ISSN : 3062-9624

Abstract

In the era of globalization, information and communication technology has rapidly developed, driving changes in various aspects of life, including the way people communicate. One of the main challenges is cross-linguistic communication, particularly in understanding foreign languages such as Japanese, which uses characters that are different from the Latin alphabet. This study develops a web-based automatic voice translation application that can recognize speech, translate it automatically, and generate human-like speech based on the translation results. The application utilizes three main technologies: Speech to Text, Machine Translation, and Text to Speech. Speech to Text and Text to Speech are implemented using the Web Speech API, while Machine Translation is implemented using the Google Translation API. The Web Speech API uses Recurrent Neural Network (RNN), and the Google Translate API uses Transformer, both of which are methods from Deep Learning algorithms. This application is designed to facilitate cross-lingual communication without the need for typing, manually translating, or directly speaking in a foreign language.

Keywords: *Speech to Text, Machine Translation, Text to Speech, Deep Learning, Recurrent Neural Network, Transformer, Web Speech API, Google Translate API*

Abstrak

Di era globalisasi, teknologi informasi dan komunikasi semakin berkembang sampai mendorong perubahan di berbagai aspek kehidupan, termasuk cara manusia berkomunikasi. Salah satu tantangan utama adalah berkomunikasi antarbahasa, khususnya dalam memahami bahasa asing seperti bahasa Jepang yang memiliki huruf yang berbeda dari huruf latin. Penelitian ini mengembangkan aplikasi penerjemah suara otomatis berbasis website yang dapat mengenali ucapan, menerjemahkan secara otomatis, dan menghasilkan suara berupa ucapan manusia berdasarkan hasil terjemahan. Aplikasi ini menggunakan tiga teknologi utama, yaitu : Speech to Text, Machine Translation, dan Text to Speech. Speech to Text dan Text to Speech diimplementasikan menggunakan Web Speech API, sedangkan Machine Translate diimplementasikan menggunakan layanan Google Translation API. Web Speech API menggunakan Recurrent Neural Network (RNN), Google Translate API menggunakan Transformer, keduanya merupakan metode dari algoritma Deep Learning. Aplikasi ini dirancang untuk memudahkan komunikasi lintas bahasa tanpa perlu mengetik, menerjemahkan secara manual, atau berbicara dalam bahasa asing secara langsung.

Kata kunci: *Speech to Text, Machine Translation, Text to Speech, Deep Learning, Recurrent Neural Network, Transformer, Web Speech API, Google Translate API*

A. PENDAHULUAN

Di era globalisasi ini, teknologi informasi dan komunikasi meningkat semakin pesat, sehingga hal ini membawa perubahan di segala aspek kehidupan manusia, ini termasuk cara manusia saling berkomunikasi. Untuk saling berkomunikasi, seorang manusia harus bisa memahami bahasa yang digunakan oleh lawan bicara. Jika lawan bicara kita menggunakan bahasa asing dan kita tidak menguasai bahasa asing tersebut, salah satu contohnya adalah jika lawan bicara kita menggunakan bahasa Jepang. Seperti yang dijelaskan pada penelitian (Istiqomah et al., 2015) bahwa bahasa Jepang sangat berbeda dengan bahasa asing lainnya, bahasa Jepang memiliki struktur bahasa dan gaya bicara yang cukup rumit, sehingga ini cukup menyusahkan bagi individu yang tidak menguasai atau yang sedang mempelajari bahasa Jepang. Untuk itu, individu membutuhkan penerjemah bahasa yang akurat agar dapat memahami konteks yang disampaikan.

Untuk menerjemahkan suatu bahasa, individu biasanya menggunakan kamus untuk menerjemahkannya secara kata per kata. Namun seiring perkembangan teknologi, banyak kamus digital mulai bermunculan. Kamus digital dapat dibuat dengan berbagai bahasa pemrograman tergantung platform yang digunakan, jika untuk Website bisa menggunakan bahasa pemrograman PHP, JavaScript, dan Python, jika untuk mobile (Android/IOS) bisa menggunakan bahasa pemrograman kotlin, java, dan swift. Tidak seperti kamus tradisional yang masih menerjemahkan kata per kata, kamus digital ini dapat menerjemahkan dalam bentuk kalimat ataupun dalam paragraf. Namun, ada beberapa kamus digital yang masih memiliki kekurangan, kekurangannya yaitu individu harus memasukkan teks terlebih dahulu ke aplikasi kamus digital untuk melakukan penerjemahan dan juga ada kemungkinan individu tersebut tidak bisa mengucapkan hasil terjemahan dalam bahasa asing dengan benar, sehingga bisa terjadi miskomunikasi antar individu.

Salah satu solusi yang bisa diberikan yaitu dengan menggunakan teknologi Speech to Text dan Text to Speech. Speech to Text (STT) merupakan suatu proses konversi ucapan manusia secara otomatis menjadi teks (Lubis et al., 2024). Dengan menggunakan STT, pengguna tidak perlu lagi memasukkan teks untuk menerjemahkannya, pengguna hanya perlu berbicara dan secara otomatis sistem akan mengonversi ucapan menjadi teks. Sedangkan Text to Speech (TTS) merupakan teknologi yang mengizinkan komputer untuk berbicara seperti layaknya pembicara asli atau mendekati pembicara asli berdasarkan teks masukan (Prastyo, 2022). Dengan menggunakan TTS, pengguna tidak akan kesusahan saat membaca hasil terjemahan.

Speech to Text dan Text to Speech dapat diimplementasikan menggunakan Web Speech API dalam bahasa pemrograman JavaScript.

Web Speech API merupakan Application Programming Interface (API) yang dikembangkan oleh Google dan dapat diakses secara gratis oleh publik. API ini memiliki kemampuan untuk mengenali suara pengguna dan juga dapat mengonversi suatu teks menjadi ucapan manusia. Web Speech API dapat diimplementasikan menggunakan Deep Learning berbasis Recurrent Neural Network (RNN). Deep Learning memiliki kemampuan untuk mempelajari pola suara yang rumit dengan memberikan model rekaman suara ke sistem. Salah satu jenis dari Deep Learning adalah Recurrent Neural Network (RNN). RNN memiliki kemampuan untuk mempertahankan informasi dalam “memori” untuk waktu yang lama dan memproses input dalam urutan, sehingga cocok untuk menangani data seperti audio.

Selain menggunakan Speech to Text dan Text to Speech, Machine Translation (MT) diperlukan untuk melakukan penerjemahan. Machine Translation merupakan proses penerjemahan dari satu bahasa ke bahasa lainnya secara otomatis yang dilakukan oleh sistem dengan cara memasukkan dataset yang besar untuk melatih model penerjemahan. Dataset ini berupa pasangan kalimat dalam dua bahasa yang berbeda dan keduanya sudah diterjemahkan. Dataset ini memiliki fungsi untuk melatih model penerjemah agar dapat memahami teks dari satu bahasa ke bahasa lain. Dengan menggunakan Google Translate API, pengembang tidak perlu melatih model lagi dari awal, karena Google Translate API sudah dilengkapi dengan Machine Translation (MT). Pada tahun 2020, Google mengumumkan model generatif baru untuk API terjemahan Google Cloud, yang berbasis Transformer dan telah meningkatkan kualitas secara signifikan (Google Cloud Translation AI, 2025). Google Translate API dilengkapi dengan dataset dalam jumlah yang sangat besar sehingga dapat menerjemahkan ke berbagai bahasa, salah satunya adalah bahasa Indonesia-Jepang. Transformer merupakan salah satu bagian dari Deep Learning. Dalam konteks Machine Translation, Transformer memiliki kemampuan untuk memproses seluruh data seperti kalimat sekaligus, bukan kata per kata. Maksudnya Transformer dapat memahami hubungan antar kata dalam suatu kalimat, meskipun kata tersebut saling berjauhan, sehingga menjadikan model ini lebih cepat dan tepat dalam memahami konteks kalimat secara keseluruhan (AWS, 2025). Hal ini menjadikan Transformer sangat bagus digunakan dalam berbagai tugas, salah satunya adalah penerjemahan bahasa.

Sesuai dengan latar belakang di atas, penulis menyusun skripsi tentang pembuatan aplikasi berbasis website dengan judul **IMPLEMENTASI ALGORITMA DEEP LEARNING PADA**

APLIKASI PENERJEMAH SUARA OTOMATIS INDONESIA-JEPANG. Berdasarkan uraian tersebut, dirumuskan beberapa permasalahan, yaitu: bagaimana algoritma Deep Learning dengan arsitektur Recurrent Neural Network (RNN) dan Transformer diimplementasikan pada pengembangan aplikasi penerjemah suara otomatis Indonesia-Jepang, serta bagaimana proses pembuatan aplikasi penerjemah suara otomatis tersebut. Adapun batasan penelitian ini meliputi penggunaan bahasa pemrograman JavaScript dengan arsitektur RNN dan Transformer, penerapan layanan Web Speech API untuk Speech to Text dan Text to Speech, serta Google Translate API untuk penerjemahan dua arah Indonesia-Jepang. Aplikasi juga mendukung input suara melalui microphone maupun unggahan file berformat mp3 atau wav.

Penelitian ini bertujuan untuk mengimplementasikan algoritma Deep Learning dengan arsitektur RNN pada aplikasi penerjemah suara otomatis Indonesia-Jepang online guna meningkatkan kinerja serta akurasi dalam proses Speech to Text dan Text to Speech. Selain itu, penelitian ini juga bertujuan untuk menerapkan arsitektur Transformer pada Machine Translation agar dapat menghasilkan terjemahan yang lebih akurat dan sesuai konteks, sehingga aplikasi yang dikembangkan mampu menjadi solusi penerjemahan suara dua bahasa secara efektif dan efisien.

B. METODE PENELITIAN

Dalam penelitian ini, penulis menggunakan berbagai perangkat keras dan perangkat lunak. Perangkat keras yang digunakan meliputi laptop dengan prosesor Intel Core i3-1115G4 3.00 GHz, RAM 8 GB, serta microphone dan speaker internal. Sedangkan perangkat lunak yang digunakan terdiri dari sistem operasi Windows 11 64-bit, text editor Visual Studio Code, browser Google Chrome, dengan bahasa pemrograman JavaScript. Aplikasi ini memanfaatkan Web Speech API untuk Speech Recognition dan Speech Synthesis serta Google Translate API untuk penerjemahan dua arah Indonesia-Jepang. Teknik pengumpulan data dilakukan melalui studi literatur dengan mengumpulkan teori dari buku, jurnal, dan artikel terkait pengembangan aplikasi. Dalam tahap perancangan, terdapat beberapa flowchart yang menggambarkan proses kerja sistem, mulai dari input suara, pre-processing, feature extraction dengan MFCC, speech recognition menggunakan RNN, decoding, hingga penerjemahan menggunakan Transformer dan konversi kembali ke suara melalui speech synthesis.

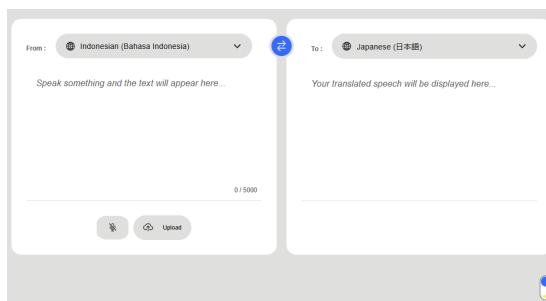
Antarmuka aplikasi dirancang sederhana dengan menyediakan tombol microphone untuk merekam suara langsung, serta fitur upload file audio berformat mp3 atau wav. Setelah suara diproses, hasil konversi ke teks ditampilkan di kotak kiri, sedangkan hasil terjemahan ke dalam bahasa Jepang muncul di kotak kanan, yang kemudian dikonversi ke audio dan dapat

langsung diputar. Jika pengguna memilih file rekaman, tombol akan berubah menjadi tombol play, dan setelah audio diputar, proses konversi dilanjutkan secara otomatis hingga audio hasil terjemahan dapat diputar kembali. Dengan penerapan algoritma Recurrent Neural Network (RNN) untuk Speech to Text dan Text to Speech, serta Transformer untuk Machine Translation, aplikasi ini diharapkan mampu memberikan hasil terjemahan suara Indonesia-Jepang yang akurat dan terdengar alami.

C. HASIL PENELITIAN DAN PEMBAHASAN

Implementasi Aplikasi

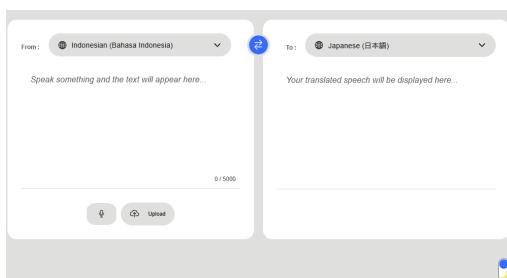
1. Tampilan Awal Antarmuka (*Interface*)



Gambar 1 Tampilan Awal Antarmuka (*Interface*)

Gambar 1 merupakan tampilan awal dari Aplikasi Penerjemah Suara Otomatis Indonesia-Jepang *Online*. Pada aplikasi ini, *user* dapat menekan *button microphone* untuk memasukkan suara. Selain *button microphone*, aplikasi ini memiliki *button* lainnya, seperti *button upload* yang berfungsi untuk memasukkan file rekaman dalam format mp3 atau wav, *button dropdown* bahasa yang memberikan pilihan bahasa Indonesia atau bahasa Jepang, *button swap* bahasa yang berfungsi untuk menukar bahasa yang berada dikanan dan kiri, dan *toggle mode* yang jika diklik akan membuat tampilan awal aplikasi yang awalnya terang menjadi gelap.

2. Tampilan Aplikasi Saat *Microphone* Aktif



Gambar 2 Tampilan aplikasi saat *microphone* diaktifkan

Gambar 2 merupakan tampilan aplikasi saat *user* mengaktifkan *microphone* untuk berbicara. Jika *user* mengatakan sesuatu, maka aplikasi akan mendeteksi perkataan *user* dan

aplikasi akan melanjutkan proses. Jika aplikasi tidak mendeteksi perkataan *user* selama 5 detik, maka *microphone* akan dinonaktifkan secara otomatis.

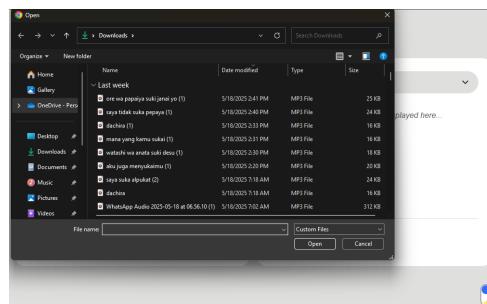
3. Tampilan Aplikasi Saat *Microphone* Dinonaktifkan



Gambar 3 Tampilan aplikasi saat *microphone* dinonaktifkan

Gambar 3 merupakan tampilan aplikasi saat *microphone* sudah dinonaktifkan. Beberapa *button* baru akan ditampilkan setelah *microphone* dinonaktifkan. *Button* tersebut adalah *speaker* yang terdapat pada *output Speech to Text*, *speaker* yang terdapat pada *output translate*, dan *refresh*. *Button speaker* pada *output Speech to Text* memiliki fungsi untuk memutar kembali suara user yang sebelumnya sudah dideteksi oleh aplikasi. *Button speaker* pada *output translate* berguna untuk memutar kembali hasil *translate* yang sebelumnya sudah dikonversi menjadi suara secara otomatis. Dan *button refresh* berguna untuk mengembalikan semua tampilan aplikasi ke kondisi awal. Pada **Gambar 4. 3**, *output* dari *Speech to Text* sudah ditampilkan pada *text area Speech to Text*, hasil *translate* ditampilkan pada *text area translate*, dan akan otomatis mengeluarkan suara manusia berdasarkan hasil dari *translate* pada *text area translate*.

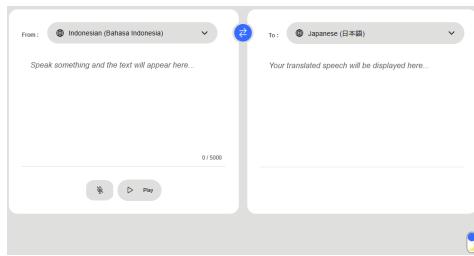
4. Tampilan Aplikasi Saat Memilih *Button Upload*



Gambar 4 Tampilan aplikasi saat memilih *button Upload*

Gambar 4 merupakan tampilan aplikasi saat klik *button Upload*. Aplikasi akan membuka file pada *device user* dan dapat memasukkan file rekaman dalam format mp3. Recording(4).mp3 akan digunakan sebagai contoh, kemudian klik *open*.

5. Tampilan Aplikasi Setelah Memilih File Rekaman



Gambar 5 Tampilan aplikasi setelah memilih file rekaman

Pada gambar 5 ditampilkan perubahan aplikasi setelah memilih file rekaman. *Button Upload* yang berguna untuk memilih file *rekaman* berubah menjadi *button Play* yang berguna untuk memainkan hasil rekaman. *Web Speech API* hanya mendukung *input* suara dari *microphone*, sehingga agar aplikasi bisa mendeteksi suara *user* dari file rekaman, maka digunakan *button play* yang berfungsi untuk memainkan suara dan mengaktifkan *microphone* sehingga aplikasi dapat mendeteksi suara *output audio* dari *device user*.

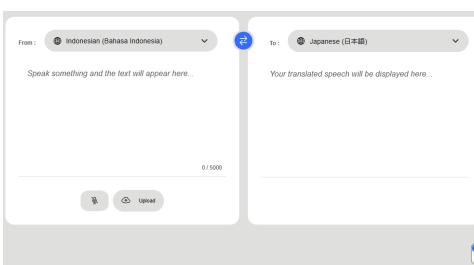
6. Tampilan Aplikasi Setelah Memilih *Button Play*



Gambar 6 Tampilan aplikasi setelah memilih *button play*

Berdasarkan gambar 6, setelah *user* klik *button Play*, maka file rekaman akan dimainkan dan *microphone* akan aktif, kemudian aplikasi memulai untuk mendeteksi suara file rekaman dari *output device*. Jika aplikasi selesai mendeteksi suara file rekaman, maka teks hasil *Speech to Text* dan hasil *translate* akan ditampilkan, kemudian *output Text to Speech* akan dihasilkan secara otomatis. *Button speaker* dan *refresh* akan ditampilkan.

7. Tampilan Aplikasi Setelah Memilih *Button Refresh*

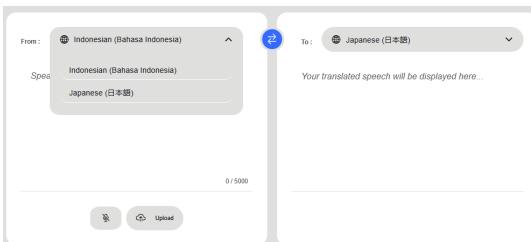


Gambar 7 Tampilan setelah memilih *button Refresh*

Gambar 7 menampilkan aplikasi setelah *user* klik *button Refresh*. Semua kondisi akan kembali seperti semula. Fitur ini diharapkan mempermudah *user* untuk menghapus file

rekaman yang sudah di *upload* dan menghapus *output* dari suara *user* yang dideteksi aplikasi dari *microphone*.

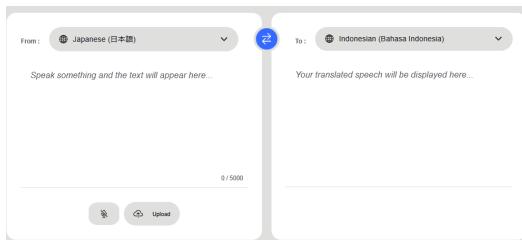
8. Tampilan Aplikasi Saat Memilih Dropdown Menu



Gambar 8 Tampilan aplikasi saat memilih Dropdown Menu

Gambar 8 menampilkan *dropdown menu* yang memberikan pilihan bahasa yang ingin digunakan. Aplikasi ini hanya menggunakan bahasa Indonesia dan bahasa Jepang, sehingga *dropdown menu* hanya memberikan dua pilihan bahasa. Walaupun terdapat *button swap* yang berfungsi untuk mengubah arah terjemahan secara instan tanpa mengubah secara manual dari *dropdown menu*, *dropdown menu* akan tetap dipertahankan jika ada kemungkinan untuk penambahan bahasa yang ingin digunakan.

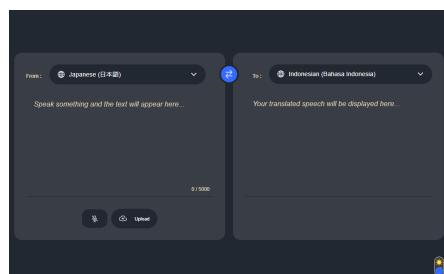
9. Tampilan Aplikasi Saat Memilih Button Swap



Gambar 9 Tampilan aplikasi setelah memilih button swap

Gambar 9 menampilkan aplikasi setelah *user* klik *button swap*. *Button swap* akan mengubah arah terjemahan secara instan. Penerjemahan bahasa Indonesia-Jepang adalah terjemahan *default* dari aplikasi ini, jika klik *button swap* maka akan menjadi terjemahan bahasa Jepang-Indonesia.

10. Tampilan Aplikasi Saat Memilih Toggle mode



Gambar 10 Tampilan aplikasi saat memilih Toggle mode

Gambar 10 menampilkan aplikasi yang menjadi gelap setelah klik *toggle mode*. Fitur ini diharapkan memberikan kenyamanan kepada mata *user* saat menggunakan aplikasi ini yang sebelumnya memiliki tampilan yang cukup terang.

Pengujian

Pengujian yang dilakukan pada penelitian ini menggunakan metode *blackbox testing* secara *full-flow (end-to-end)*, yang berarti aplikasi ini akan diuji mulai dari *Speech to Text*, *Translate*, sampai *Text to Speech* dalam satu alur. *Blackbox testing* merupakan salah satu metode pengujian perangkat lunak dimana penguji fokus pada *input* dan *output* dari perangkat lunak yang sudah dibangun.

Tabel 1 Tabel Pengujian Aplikasi

No.	Skenario Uji Coba	Bahasa	Input	Langkah Uji	Output Yang diharapkan	Status	Keterangan
1.	Pengujian dengan memasukkan suara yang jelas	IND	“Selamat Pagi”	<i>Input</i> menggunakan <i>micro-phone</i>	Audio “oha-you”	valid	Menghasilkan terjemahan dan <i>output</i> audio yang diharapkan
				<i>Input</i> file rekaman	Audio “oha-you”	valid	Menghasilkan terjemahan dan <i>output</i> audio yang diharapkan
		JPN	“oha-you”	<i>Input</i> menggunakan <i>micro-phone</i>	Audio “selamat pagi”	valid	Menghasilkan terjemahan dan <i>output</i> audio yang diharapkan
				<i>Input</i> file rekaman	Audio “selamat pagi”	valid	Menghasilkan terjemahan dan <i>output</i> audio yang diharapkan
2.	Pengujian <i>noise</i> rendah di latar belakang	IND	“aku menyukai-mu”	<i>Input</i> menggunakan <i>micro-phone</i> dengan	Audio “watashi wa anata ga suki desu”	valid	Menghasilkan terjemahan dan audio sesuai

				latar belakang <i>noise</i> berupa suara kipas angin yang diarahkan ke <i>micro-phone</i> laptop			dengan harapan
				<i>Input</i> file rekaman yang memiliki latar belakang <i>noise</i> berupa suara kipas angin yang diarahkan ke <i>micro-phone</i> laptop saat perekaman	Audio “watashi wa anata ga suki desu” berdasarkan hasil terjemahan file rekaman	valid	Menghasilkan terjemahan dan audio sesuai dengan harapan
	JPN	“wata-shi wa anata ga suki desu”		<i>Input</i> menggunakan <i>micro-phone</i> dengan latar belakang <i>noise</i> berupa suara kipas angin yang diarahkan ke <i>micro-phone</i> laptop	Audio “aku juga menyukaimu”	invalid	Menghasilkan terjemahan “aku menyukaimu”, kata “juga” tidak ada pada terjemahan

				<i>Input</i> file rekaman yang memiliki latar belakang <i>noise</i> berupa suara kipas angin yang diarahkan ke <i>micro-phone</i> laptop saat perekaman	Audio “aku juga menyukaimu” berdasarkan hasil terjemahan file rekaman	valid	Berhasil menghasilkan terjemahan yang sesuai dengan hasil terjemahan dan audio
3.	Pengujian dengan <i>noise</i> tinggi di latar belakang	IND	“man a yang kamu su-kai?”	<i>Input</i> menggunakan <i>micro-phone</i> dengan latar belakang <i>noise</i> berupa suara kipas angin yang diarahkan sangat dekat dengan <i>micro-phone</i> laptop	Audio “dochiraga suki desu”	invalid	Aplikasi mendeteksi kalimat “apa yang kamu suka”, sehingga menghasilkan terjemahan dan <i>output</i> yang berbeda dari yang diharapkan
				<i>Input</i> file rekaman dengan latar belakang <i>noise</i> berupa suara kipas	Audio “dochiraga suki desu” berdasarkan file rekaman	Valid/ invalid	Aplikasi mengalami beberapa kali kegagalan dan ada keberhasilan saat mendeteksi

				angin yang diarahkan sangat dekat dengan <i>micro-phone</i> saat perekaman			audio dari file rekaman dengan tepat
JPN	“do-chira ga suki desuka?”	<i>Input</i> menggunakan <i>micro-phone</i> dengan latar belakang <i>noise</i> berupa suara kipas angin yang diarahkan sangat dekat dengan <i>micro-phone</i> laptop	Audio “mana yang kamu sukai”	invalid	Aplikasi mendeteksi kalimat “kirā ga suki desuka”, sehingga terjemahan dan audio memiliki <i>output</i> yang berbeda dari yang diharapkan	Valid/invalid	Aplikasi beberapa kali mengalami kegagalan dan keberhasilan saat mendeteksi suara file rekaman dengan tepat

				saat perekaman			
4.	Pengujian dengan kalimat panjang.	IND	“Be-sok pagi saya pergi ke kantor dan mengirim dokumen penting ke bos saya”	<i>Input</i> menggunakan <i>micro-phone</i>	Audio “ashita no asa , watashi ha ofisu ni iki , joushi ni juuyou na bunsho o okuri masu”	valid	Aplikasi berhasil mendeteksi suara <i>user</i> dan menerjemahkan serta menghasilkan audio sesuai dengan harapan
				<i>Input</i> file rekaman	Audio “ashita no asa , watashi ha ofisu ni iki , joushi ni juuyou na bunsho o okuri masu”	valid	Aplikasi berhasil mendeteksi suara <i>user</i> yang berada di file rekaman dan menerjemahkan serta menghasilkan audio sesuai dengan harapan
		JPN	“ashi-ta no asa , wata-shi ha ofisu ni iki , joushi ni juu-you na bun-sho o okuri masu”	<i>Input</i> menggunakan <i>micro-phone</i>	Audio “Besok pagi saya pergi ke kantor dan mengirim dokumen penting ke bos saya”	valid	Aplikasi berhasil mendeteksi suara <i>user</i> dan menerjemahkan serta menghasilkan audio sesuai dengan harapan
				<i>Input</i> file rekaman	Audio “Besok pagi saya pergi ke kantor	valid	Aplikasi berhasil mendeteksi suara <i>user</i> yang berada di file

					dan mengirim dokumen penting ke bos saya "		rekaman dan menerjemahkan serta menghasilkan audio sesuai dengan harapan
5.	Pengujian dengan suara pelan	IND	"Saya suka alpukat"	<i>Input</i> menggunakan <i>micro-phone</i>	valid	valid	Aplikasi dapat mendeteksi suara yang pelan dan memberikan terjemahan serta audio yang sesuai
				<i>Input</i> file rekaman	Audio "ore wa abokado suki desu" berdasarkan file rekaman	valid	Aplikasi dapat mendeteksi suara pelan dan memberikan terjemahan serta audio yang sesuai
		JPN	"Ore wa abo-kado ga suki desu"	<i>Input</i> menggunakan <i>micro-phone</i>	Audio "saya suka alpukat"	valid	Aplikasi dapat mendeteksi suara yang pelan dan memberikan terjemahan serta audio yang sesuai
				<i>Input</i> file rekaman	Audio "saya suka alpukat" berdasarkan file rekaman	valid	Aplikasi dapat mendeteksi suara yang pelan dan memberikan terjemahan serta audio yang sesuai
6.	Pengujian dengan suara keras	IND	"Saya tidak suka pepaya"	<i>Input</i> menggunakan <i>micro-phone</i>	Audio "ore wa papaiya janai yo"	invalid	Aplikasi dapat mendeteksi suara user dan

							menghasilkan teks sesuai dengan yang diucapkan oleh <i>user</i> , namun menghasilkan terjemahan “watashi ha papaiya ga suki de ha ari mase n” yang terkesan lebih formal sehingga menghasilkan audio yang berbeda dari harapan
				<i>Input</i> file rekaman	Audio “ore wa papaiya janai yo” berdasarkan file rekaman	invalid	Aplikasi dapat mendeteksi suara <i>user</i> dan menghasilkan teks sesuai dengan yang diucapkan oleh <i>user</i> , namun menghasilkan terjemahan “watashi ha papaiya ga suki de ha ari mase n” yang terkesan lebih formal sehingga menghasilkan audio yang berbeda dari harapan
JPN	“Ore wa papai ya ga	“Input menggunakan	Audio “saya tidak	valid	Aplikasi dapat mendeteksi suara <i>user</i>		

			suki janai yo"	<i>micro-phone</i>	suka pepaya"		dan menghasilkan terjemahan serta audio yang sesuai dengan harapan
				<i>Input</i> file rekaman	Audio "saya tidak suka pepaya" berdasarkan file rekaman	valid	Aplikasi dapat mendeteksi suara <i>user</i> dan menghasilkan terjemahan serta audio yang sesuai dengan harapan
7.	Pengujian dengan multiple voice	IND	Dua orang berbicara secara bersamaan	<i>Input</i> menggunakan <i>micro-phone</i>	Menghasilkan kalimat yang terpisah	invalid	<i>Output</i> menghasilkan kalimat yang berbeda dari yang diucapkan kedua <i>user</i>
				<i>Input</i> file rekaman	Menghasilkan kalimat yang terpisah	invalid	<i>Output</i> menghasilkan kalimat yang berbeda dari yang diucapkan kedua <i>user</i>
		JPN	Dua orang berbicara secara bersamaan	<i>Input</i> menggunakan <i>micro-phone</i>	Meghasilkan kalimat yang terpisah	invalid	<i>Output</i> menghasilkan kalimat yang berbeda dari yang diucapkan kedua <i>user</i>
				<i>Input</i> file rekaman	Meghasilkan kalimat yang terpisah	invalid	<i>Output</i> menghasilkan kalimat yang berbeda dari

							yang diucapkan kedua user
8.	Pengujian dengan tidak ada suara	IND	Tidak ada suara yang dimaksudkan	<i>Input</i> menggunakan <i>micro-phone</i>	Tidak menghasilkan suara	valid	Aplikasi tidak menghasilkan <i>output</i> apapun
					Tidak menghasilkan suara	valid	Aplikasi tidak menghasilkan <i>output</i> apapun
		JPN	"satu juta dua ratus lima puluh ribu"	<i>Input</i> file rekaman	Tidak menghasilkan suara	valid	Aplikasi tidak menghasilkan <i>output</i> apapun
					Tidak menghasilkan suara	valid	Aplikasi tidak menghasilkan <i>output</i> apapun
9.	Pengujian dengan menyebutkan angka	IND	"satu juta dua ratus lima puluh ribu"	<i>Input</i> menggunakan <i>micro-phone</i>	Audio "hyaku ni juu go man"	invalid	Menghasilkan terjemahan "100 man nin no 25 man"
					<i>Input</i> file rekaman	invalid	Menghasilkan terjemahan "100 man nin no 25 man"
		JPN	"hyaku ni juu go man"	<i>Input</i> menggunakan <i>micro-phone</i>	Audio "satu juta dua ratus lima puluh ribu"	invalid	Menghasilkan terjemahan "1,25 juta" dan menghasilkan suara "satu koma dua lima juta"
					<i>Input</i> file rekaman	invalid	Menghasilkan terjemahan "1,25 juta"

					lima puluh ribu” berdasar -kan file rekaman		dan menghasilkan suara “satu koma dua lima juta”
10.	Pengujian dengan kata informal	IND	“Gue lagi makan siang sama mama”	<i>Input</i> menggunakan <i>micro-phone</i>	Audio “Ore wa mama to ohiru-gohan tabeteru yo	invalid	Menghasilkan terjemahan “mama to chuushoku o totsu te i masu” yang terkesan lebih formal
				<i>Input</i> file rekaman	Audio “Ore wa mama to ohiru-gohan tabe-teru yo” berdasar -kan file rekaman	invalid	Menghasilkan terjemahan “mama to chuushoku o totsu te i masu” yang terkesan lebih formal
			“Ore wa mama to ohiru - gohan tabe-teru yo”	<i>Input</i> menggunakan <i>micro-phone</i>	Audio “gue lagi makan siang dengan ibu”	invalid	Menghasilkan terjemahan “Saya makan siang dengan ibu” yang menggunakan kata baku
				<i>Input</i> file rekaman	Audio “gue lagi makan siang dengan ibu” berdasar -kan file rekaman	invalid	Menghasilkan terjemahan “Saya makan siang dengan ibu” yang menggunakan kata baku

Berdasarkan pengujian aplikasi yang telah dilakukan penulis, dapat ditemukan bahwa sistem mengalami kendala dalam mengenali dan menerjemahkan beberapa kata dari bahasa Indonesia yang memiliki makna dalam bahasa Jepang, contohnya saat kata-kata tersebut

digunakan sebagai nama orang. Misalnya, pada kalimat seperti “Nama saya Bintang”, “Nama saya Langit”, atau “Nama saya Sakura”, sistem akan menerjemahkan kata-kata tersebut sesuai dengan makna dalam bahasa Jepang, seperti “Bintang” menjadi 星 (hoshi), “Langit” menjadi 空 (sora), dan “Sakura” tetap menjadi 桜 (sakura). Hal ini membuktikan bahwa sistem belum mampu untuk membedakan antara kata yang digunakan sebagai nama dengan kata yang memiliki makna dalam bahasa Jepang, sehingga dapat menimbulkan ketidaksesuaian dalam hasil terjemahan.

Selain itu, hasil pengujian juga menunjukkan adanya perbedaan respon pada aplikasi pada proses penerjemahan dua arah, yaitu antara bahasa Indonesia ke bahasa Jepang dan bahasa Indonesia ke bahasa Jepang. Perbedaan akan terjadi saat aplikasi menemukan kata yang memiliki makna dalam bahasa Jepang yang digunakan sebagai nama pada suatu kalimat. Pada penerjemahan Indonesia-Jepang, aplikasi akan menerjemahkan nama tersebut ke bahasa Jepang sesuai dengan maknanya. Selain itu, jika nama tersebut tidak memiliki makna dan terdapat kombinasi huruf atau bunyi yang tidak tersedia dalam bahasa Jepang, maka akan terjadi penyesuaian pelafalan dan penulisan kata berdasarkan bunyi dan karakter huruf yang dimiliki bahasa Jepang. Sedangkan pada arah Jepang-Indonesia, sistem pengenalan suara (*Speech Recognition*) mengalami kesulitan dalam mengenali pengucapan nama asing atau kata-kata yang tidak umum digunakan dalam bahasa Jepang. Sehingga, beberapa kata sering mengalami tidak terdeteksi secara akurat atau bahkan tidak terdeteksi sama sekali oleh aplikasi. Hal ini menandakan bahwa aplikasi belum sepenuhnya mampu untuk mengenali nama-nama yang tidak umum, dan arah penerjemahan yang berbeda menyebabkan hasil yang berbeda pula.

Analisa

Analisa ini penulis dapat berdasarkan waktu yang dibutuhkan oleh aplikasi untuk mengonversi ucapan *user* menjadi teks, menerjemahkan teks, dan mengonversi teks menjadi suara antara *input* suara ke dalam aplikasi melalui *microphone* dengan *input* suara menggunakan file rekaman.

Tabel 2 Waktu proses skenario uji

No.	Skenario Uji	Bahasa	Input	Speech to Text (detik)	Translate (detik)	Text to Speech (detik)
1.	Kalimat sederhana	IND	Ucapkan “Selamat”	8.01	0.26	1.06

			pagi” di <i>microphone</i>			
		JPN	Ucapkan “Ohayou” di <i>microphone</i>	9.17	0.68	1.86
2.	Kalimat sederhana	IND	Masukkan file rekaman yang berisi “Selamat pagi”	2.41	0.29	1.07
		JPN	Masukkan file rekaman yang berisi “Ohayou”	2.62	0.28	1.86
3.	Kalimat panjang	IND	Ucapkan “Besok pagi saya pergi ke kantor dan mengirim dokumen penting ke bos saya” di <i>microphone</i>	11.96	2.16	6.36
		JPN	Ucapkan “ashita no asa , watashi wa ofisu ni iki , joushi ni juu-you na bunsho o okuri masu” di <i>micro-phone</i>	16.17	1.54	6.44
4.	Kalimat Panjang	IND	Masukkan file rekaman yang berisi “Besok pagi saya pergi ke kantor dan mengirim dokumen penting ke bos saya ”	6.14	0.17	6.36
		JPN	Masukkan file rekaman yang berisi “ashita no asa , watashi wa ofisu ni iki , joushi ni juu-	8.10	1.04	5.55

			you na bunsho o okuri masu”			
--	--	--	---	--	--	--

Berdasarkan tabel diatas, peneliti menyimpulkan bahwa: (1) *Input* melalui *microphone* membutuhkan waktu yang lebih lama untuk mengonversi ucapan menjadi teks daripada *input* file rekaman. (2) Bahasa Jepang membutuhkan waktu yang lebih lama untuk diproses daripada bahasa Indonesia. (3) *Translate* merupakan tahap yang paling ringan dalam proses *full-flow*. (4) Panjang pendeknya suatu kalimat sangat berpengaruh terhadap waktu proses, terutama pada *Speech to Text* dan *Text to Speech*.

D. KESIMPULAN

Berdasarkan hasil penelitian dan implementasi yang telah dilakukan, dapat disimpulkan bahwa algoritma Deep Learning berhasil diterapkan pada Aplikasi Penerjemah Suara Otomatis Indonesia-Jepang Online yang dapat bekerja secara otomatis, mulai dari mengenali suara pengguna, mengonversi suara menjadi teks, menerjemahkan teks ke bahasa sasaran, hingga menghasilkan suara berdasarkan hasil terjemahan. Aplikasi ini meningkatkan efisiensi pengguna dalam penerjemahan, sehingga pengguna tidak perlu lagi mengetik teks yang ingin diterjemahkan atau membaca hasil terjemahan. Berdasarkan pengujian dengan berbagai skenario, aplikasi ini menghasilkan output yang cukup akurat, baik dalam konversi suara menjadi teks, penerjemahan, maupun konversi teks menjadi suara. Penerapan Recurrent Neural Network (RNN) pada sistem pengenalan dan sintesis suara juga membantu mengoptimalkan konversi antara teks dan suara. Namun, akurasi pengenalan suara dapat menurun jika terdapat kebisingan yang tinggi di sekitar saat memasukkan suara. Penggunaan Transformer pada proses penerjemahan memberikan hasil yang baik pada kalimat sederhana, meskipun keakuratannya berkurang pada kalimat informal dan dalam menerjemahkan ungkapan angka.

E. REFERENSI

- Admin. (2021, December 28). Definisi dan Cara Kerja Sistem Text to Speech. Widya Wicara. <https://widywawicara.com/definisi-dan-cara-kerja-sistem-text-to-speech/>
- AWS. (2025). Apa itu Terjemahan Mesin? - Penjelasan tentang Terjemahan Mesin - AWS. Amazon Web Services, Inc. <https://aws.amazon.com/id/what-is/machine-translation/>
- Dr. Budi Raharjo, S.Kom., M.Kom., MM. (n.d.). Deep Learning dengan Python. Yayasan Prima Agus Teknik.

- Fitriani, Y., Utami, S., & Junadi, B. (2022). Perancangan Sistem Informasi Human Capital Management Berbasis Website. 6.
- Google Cloud Translation AI. (n.d.). Google Cloud Blog. Retrieved April 13, 2025, from <https://cloud.google.com/blog/products/ai-machine-learning/google-cloud-translation-ai>
- Habash, F. (2023, November 27). Understanding the Basics: What is Machine Translation? Localization Services by BLEND. <https://www.getblend.com/blog/what-is-machine-translation/>
- Handayani, R., Rachmat, Z., & S, W. (2022). Perancangan Aplikasi E-Learning Berbasis Website Pada SMP Negeri 3 Watansoppeng. *Jurnal Manajemen Informatika, Sistem Informasi dan Teknologi Komputer (JUMISTIK)*, 1(1), 43–54. <https://doi.org/10.70247/jumistik.v1i1.8>
- Hawari Nasution, M. A. A., Siswanto, S., & Suryana, E. (2023). RANCANGAN MEDIA PEMBELAJARAN BERUPA APLIKASI AUGMENTED REALITY BERBASIS ANDROID. *JURNAL MEDIA INFOTAMA*, 19(2), 528–537. <https://doi.org/10.37676/jmi.v19i2.4771>
- Hosting, R. J. (2022, June 22). Apa itu Flowchart? Fungsi, Simbol & Jenisnya. Blog Jagoan Hosting. <https://www.jagoanhosting.com/blog/flowchart-adalah/>
- Istiqomah, D., Diner, L., & Wardhana, C. K. (2015). ANALISIS KESULITAN BELAJAR BAHASA JEPANG SISWA SMK BAGIMU NEGERIKU SEMARANG.
- Jaman, A. B., & Fergina, A. (2021). Implementasi Speech Recognition Berbasis Android Dalam Optimalisasi Komunikasi Bagi Penyandang Tunarungu. *Jurnal Teknik Informatika UNIKA Santo Thomas*, 373–378. <https://doi.org/10.54367/jtiust.v6i2.1508>
- Khairiah, I. (n.d.). Algoritma Pemrograman: Studi Pustaka Pemahaman Algoritma Pemrograman. 1(4). <https://doi.org/10.59581/konstanta.v1i4.1673>
- Khoiriyah, R. (n.d.). PROGRAM STUDI S1 – TEKNIK INFORMATIKA JURUSAN TEKNOLOGI INFORMASI FAKULTAS TEKNOLOGI INFORMASI DAN KOMUNIKASI UNIVERSITAS SEMARANG 2023.
- Lubis, N., Siambaton, M. Z., & Aulia, R. (2024). Implementasi Algoritma Deep Learning pada Aplikasi Speech to Text Online dengan Metode Recurrent Neural Network (RNN). 3(3).
- Mallisa, I. G. (2021). IMPLEMENTASI INTEGRATED DEVELOPMENT ENVIRONMENT (IDE) BERBASIS WEB UNTUK PEMROGRAMAN JAVASCRIPT.

- Nugroho, P. A., Fenriana, I., Arijanto, R., & Kom, M. (2020). IMPLEMENTASI DEEP LEARNING MENGGUNAKAN CONVOLUTIONAL NEURAL NETWORK (CNN) PADA EKSPRESI MANUSIA. 2(1).
- Oleh, D., & Irawan, F. (n.d.). PREDIKSI CURAH HUJAN MENGGUNAKAN RECURRENT NEURAL NETWORK (RNN) DAN LONG SHORT-TERM MEMORY (LSTM).
- Parjito, Oktavia Rahmawati, & Faruq Ulum. (2022). Rancang Bangun Aplikasi E-Agribisnis Untuk Meningkatkan Penjualan Hasil Tanaman Hortikultura. Jurnal Informatika Dan Rekayasa Perangkat Lunak (JATIKA), 3(3), 354–365.
<http://jim.teknokrat.ac.id/index.php/informatika>
- Prastyo, M. E. (n.d.). PROGRAM STUDI TEKNIK INFORMATIKA FAKULTAS KOMUNIKASI DAN INFORMATIKA UNIVERSITAS MUHAMMADIYAH SURAKARTA 2022.
- Saiantisna, M. D., Sukarini, N. W., & Juliana, K. A. (n.d.). Implementasi Aplikasi Penerjemah Multi Bahasa Berbasis Python dengan Integrasi Google Translate API dan GUI Tkinter. 17.
- Silverwood Summit. (2023). Memahami Dasar-Dasar HTML dan CSS: Fondasi Pengembangan Web. Teknologiterkini.Org, 3(9).
- Swapurba, G., & Pratama, I. (2023). Implementasi Voice To Text Pada Invoice Checking Berbasis Web. Intechno Journal (Information Technology Journal), 5(2), 65–74.
<https://doi.org/10.24076/intechojournal.2023v5i2.1391>
- UMA, B. (2024, June 29). Teknologi Speech Synthesis: Transformasi dan Inovasi. Biro Perencanaan Mutu Pendidikan dan Pembelajaran Terbaik di Sumatera Utara.
<https://bpmpp.uma.ac.id/2024/06/29/teknologi-speech-synthesis-transformasi-dan-inovasi-dalam-komunikasi-digital/>