sudo
Jurnal Teknik Informatika
https://jurnal.ilmubersama.com/index.php/sudo

Deep Learning

Implementasi Algoritma *Deep Learning* pada Aplikasi *Speech to Text Online* dengan Metode *Recurrent Neural Network (RNN)*

Nadhira Lubis [1], Mhd. Zulfansyuri Siambaton [1], Rachmat Aulia [2]*

[1] *Fakultas Teknik, Teknik Informatika, Universitas Islam Sumatera Utara, Medan, Indonesia*
[2] *Fakultas Teknik, Teknik Informatika, Universitas Harapan Medan, Medan, Indonesia*

INFORMASI ARTIKEL

Diterima Redaksi: 17 Juli 2024
Revisi Akhir: 13 Agustus 2024
Diterbitkan *Online*: 03 September 2024

KATA KUNCI

Speech to Text
Speech Recognition
Algoritma *Deep Learning*
Metode *Recurrent Neural Network* (RNN)
Web Speech API

KORESPONDENSI (*)

Phone: +62 822-1302-2582
E-mail: nadhiralubis01@gmail.com

A B S T R A K

Pada era globalisasi, kemajuan teknologi informasi dan komunikasi telah membawa perubahan signifikan dalam berbagai aspek kehidupan manusia, termasuk cara kita berinteraksi dengan perangkat digital. Salah satu inovasi yang menonjol adalah teknologi Pengenalan Suara (*Speech Recognition*), yang merupakan bagian dari *Web Speech API* dalam bahasa pemrograman JavaScript. Teknologi ini mampu memahami dan memproses bahasa lisan, mengenali suara manusia, dan mengubahnya menjadi teks di komputer. Teknologi Pengenalan Suara memiliki peran penting dalam aplikasi *Speech to Text Online*. Teknologi ini diimplementasikan langsung di browser dan dapat diakses melalui *Web Speech API* dalam JavaScript. Karena manusia umumnya dapat berbicara lebih cepat daripada mengetik, aplikasi *Speech to Text Online* dirancang untuk memudahkan aktivitas yang berhubungan dengan pengetikan atau pembuatan catatan. Aplikasi ini menggunakan algoritma *Deep Learning* yang dilengkapi dengan metode *Recurrent Neural Network* (RNN). Metode RNN, sebagai bagian dari algoritma *Deep Learning*, memiliki kemampuan untuk mempelajari pola dari data yang kompleks dan abstrak. Hal ini membuatnya sangat efektif dalam pengenalan gambar, pemrosesan bahasa alami, pengenalan suara, dan banyak bidang lainnya. Oleh karena itu, RNN adalah metode yang tepat untuk diterapkan dalam algoritma *Deep Learning* pada aplikasi *Speech to Text Online*.

PENDAHULUAN

Dalam era globalisasi, kemajuan teknologi informasi dan komunikasi telah mengubah berbagai aspek kehidupan, termasuk interaksi dengan perangkat digital. Salah satu inovasi penting adalah teknologi *Speech Recognition* yang merupakan bagian dari *Web Speech API* di JavaScript. Teknologi ini mampu memahami dan mengubah bahasa lisan manusia menjadi teks, memainkan peran penting dalam aplikasi *Speech to Text Online*. Dengan *Web Speech API*, implementasi algoritma *Deep Learning* dalam aplikasi ini menjadi lebih mudah, meskipun tantangannya tetap ada, seperti kualitas data suara dan keberagaman aksen.

Penelitian ini bertujuan untuk mengimplementasikan algoritma *Deep Learning* dalam aplikasi *Speech to Text* dengan bantuan *Speech Recognition*. Aplikasi ini dirancang untuk mengubah suara manusia menjadi teks, memudahkan berbagai aktivitas seperti pencatatan dan pembuatan konten tanpa perlu mengetik manual. Algoritma *Deep Learning* memungkinkan komputer mengenali suara manusia dengan lebih baik, dengan *Recurrent Neural Network* (RNN) yang meningkatkan akurasi pengenalan ucapan dan menghasilkan transkripsi teks yang lebih akurat.

https://doi.org/10.56211/sudo.v3i3.583

TINJAUAN PUSTAKA

Speech to Text

Konversi suara menjadi teks yang biasanya dapat disebut juga dengan *Speech to Text* adalah perangkat lunak transkripsi yang secara otomatis mengenali ucapan dan mentranskripsikan apa yang diucapkan ke dalam format tertulis yang setara. Secara tradisional, manusia akan mendengarkan *file* audio dan mengetiknya ke dalam *file* teks guna menggunakan kembali konten yang diucapkan untuk media yang berbeda. Namun dengan menggunakan kecerdasan buatan, sekarang komputer dapat dengan mudah mengonversi suara menjadi teks dalam waktu singkat dan membuat konten tersebut dapat digunakan untuk tujuan yang berbeda, seperti pencarian, *subtitle*, dan wawasan. Pada *Speech to Text* terdapat beberapa pendekatan/metode dalam proses analisis pola suara/bahasa ke dalam bentuk tulisan/teks. *Speech to Text* ini dapat diterapkan dalam berbagai hal diantaranya sebagai alat bantu komunikasi bagi tuna rungu (tuli), *smart home*, atau penerjemah. Output dari *Speech to Text* akan menjadi input dan akan menggantikan input *text* secara manual sehingga mampu mempermudah dalam penerapannya di berbagai bidang. Beberapa pendekan/metode yang sering digunakan diantaranya yaitu: *Deep Neural Network* (DNN), *Recurrent Neural Network* (RNN), *Hidden Markov Model* (HMM), dan lain sebagainya[1].

Web Speech API (*Web Speech Application Programming Interface*)

Web Speech API merupakan *API* yang terdapat dalam JavaScript yang mengatur data suara dalam aplikasi. *Web Speech API* memiliki dua bagian yaitu *SpeechSyntesis* (*Text to Speech*) dan *Speech Recognition* [2].

1. *Speech Syntesis*, merupakan bagian *interface* dari *Web Speech API* yang mengontrol dalam hal pengucapan suara dari teks masukan (text to speech). Pada kenyataannya *Speech Synthesis* memerlukan *interface* lain dari *Web Speech API* yaitu *Speech Synthesis Utterance* yang berperan sebagai objek teks masukan yang telah diproses agar dapat diproses oleh *Speech Synthesis*.
2. *Speech Recognition*, merupakan bagian *interface* dari *Web Speech API* yang mengontrol dalam hal pengenalan suara sehingga suara dapat dikenali dan diproses menjadi teks.

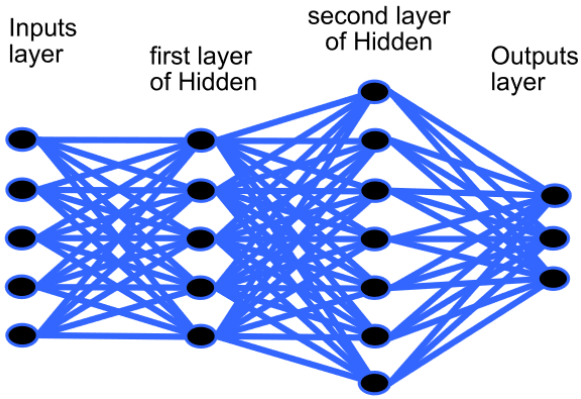
Speech Recognition

Speech Recognition atau yang biasa dikenal dengan *Automatic Speech Recognition* (ASR) adalah suatu pengembangan teknik dan sistem yang memungkinkan komputer dapat menerima masukan berupa kata yang diucapkan. Teknologi ini memungkinkan suatu perangkat dapat mengenali serta memahami kata-kata yang diucapkan dengan cara digitalisasi kata dan mencocokkan sinyal digital tersebut melalui suatu pola tertentu yang tersimpan dalam suatu perangkat. Kata-kata yang diucapkan kemudian diubah menjadi sinyal digital dengan cara mengubah gelombang suara menjadi suatu kumpulan angka-angka yang kemudian di terjemahkan dengan kode-kode tertentu untuk mengidentifikasikan kata-kata tersebut [3].

Algoritma Deep Learning

Deep Learning adalah cabang dari *Machine Learning* yang terinspirasi oleh struktur dan fungsi otak manusia, yang disebut sebagai jaringan saraf tiruan atau *Artificial Neural Network*. *Deep Learning* dapat didefinisikan sebagai teknik pembelajaran mesin yang menggunakan jaringan saraf tiruan dengan banyak lapisan neuron untuk mengekstraksi fitur yang kompleks dan mewakili data input yang kompleks dalam hierarki. Jumlah dan kompleksitas lapisan neuron adalah yang membedakan *Deep Learning* dari pembelajaran mesin konvensional. Kemampuan *Deep Learning* untuk mempelajari pola dari data yang kompleks dan abstrak menjadikannya sangat sukses dalam pengenalan citra, pemrosesan bahasa alami, pengenalan suara, dan banyak bidang lainnya [1]. Salah satu metode dari algoritma *Deep Learning* ialah *Recurrent Neural Network* (RNN).

 https://doi.org/10.56211/sudo.v3i3.583



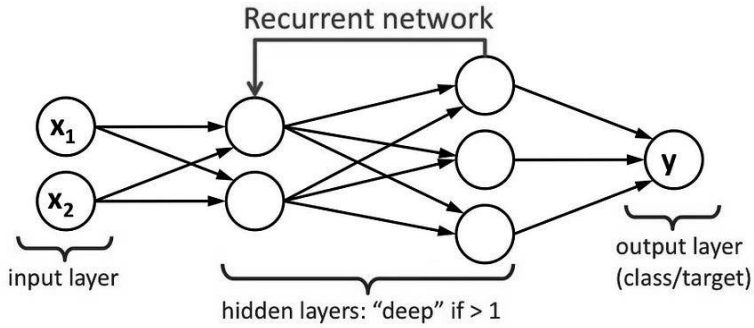
Gambar 1. Ilustrasi arsitektur pada *Deep Learning*

Prinsip Kerja Deep Learning

Deep Learning merupakan pembelajaran yang berbasis pada fitur yang berbentuk hirarki, yang mana bentuk fitur hirarki tersebut dapat diskalakan dalam ukuran tertentu yang dapat disesuaikan dengan kasus yang diproses. Sehingga membuat algoritma *Deep Learning* mampu untuk melakukan ekstraksi fitur otomatis dari data mentah secara detail, karena proses ekstraksi yang dilakukan menggunakan struktur eksploitasi, dimana fitur-fitur yang dieksploitasi tersebut tidak mungkin dapat dilihat secara kasat mata. Hal ini dikarenakan distribusi pembeda kelas data biasanya terlalu dalam, sehingga dari fitur level tinggi harus ditransformasi terlebih dahulu ke fitur yang paling rendah yang dapat mudah dipahami oleh mesin pembelajaran. Artinya jika fitur level rendah saja dapat mudah diindentifikasi oleh *Deep Learning*, apalagi yang fitur level tinggi. Perpaduan inilah yang menjadikan *Deep Learning* mampu menghasilkan representasi fitur yang baik dan optimal [4].

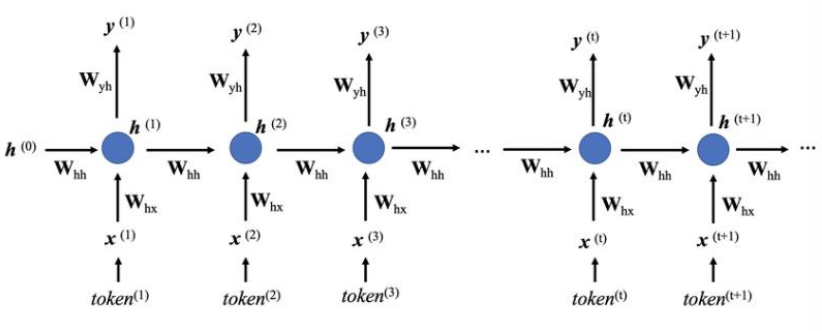
Metode Recurrent Neural Network (RNN)

Recurrent Neural Network (RNN) adalah sebuah metode *Deep Learning* yang dapat melakukan pembelajaran terhadap pola yang terdapat di dalam data sekuensial. *Recurrent Neural Network* (RNN) memiliki kemampuan untuk "mengingat" elemen data yang telah "dipelajari" sebelumya. Dengan kemampuan ini maka RNN mampu memprediksi sebuah elemen (urutan elemen) data yang akan muncul kemudian berdasarkan data input sebelumnya [5]. Pengenalan suara, mesin terjemahan, pemodelan bahasa tingkat karakter, klasifikasi gambar, keterangan gambar, prediksi saham, dan rekayasa keuangan merupakan contoh aplikasi RNN [6][7].



Gambar 2. Model pada *Recurrent Neural Network* (RNN)

Recurrent Neural Network (RNN) adalah sebuah kelas *neural network* yang memiliki sejumlah koneksi berupa *loop* yang berfungsi sebagai *input* sebuah *neuron* (selanjutnya disebut sel RNN) [5]. *Recurrent Neural Network* (RNN) akan memproses data *input* satu per satu secara sekuensial, *hidden layer* pun akan melempar data menuju ke *hidden layer* pada skala waktu selanjutnya. Begitu seterusnya secara sekuensial [8].



Gambar 3. Proses RNN ketika melakukan perhitungan

Persamaan Neuron Hidden pada gambar 3 ialah [9]:

$$h^{(i)} = \sigma_h (W_{hh} \cdot h^{(i-1)} + W_{hx} x^{(i)} + b_h)$$

Keterangan:

- $h^{(i)}$ adalah nilai neuron hidden pada timestep i
- $x^{(i)}$ adalah nilai input
- σ_h adalah fungsi aktivasi
- W_{hh} adalah parameter weight untuk hubungan antara neuron hidden sebelumnya
- $h^{(i-1)}$ adalah nilai output dari posisi sebelumnya
- W_{hx} adalah parameter weight untuk hubungan antara input x_t dan neuron hidden
- b_h adalah bias neuron hidden

Persamaan Output pada gambar 3 ialah [9]:

$$y^{(i)} = \sigma_y (W_{yh} h^{(i)} + b_y)$$

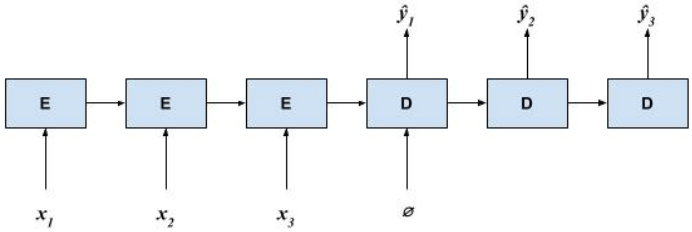
Keterangan:

- $y^{(i)}$ adalah output pada timestep i
- σ_y adalah fungsi aktivasi
- W_{yh} adalah parameter weight untuk hubungan antara neuron hidden dan output
- $h^{(i)}$ adalah nilai output
- b_y adalah bias output

Jenis-Jenis Pemrosesan Recurrent Neural Network (RNN)

Berikut ini ialah jenis-jenis dari pemrosesan pada *Recurrent Neural Network* (RNN) [10]:

1. *Many To Many* (Banyak Ke Banyak)
   Pada jenis ini, baik input dan output adalah data sekuensial yang jumlahnya banyak.

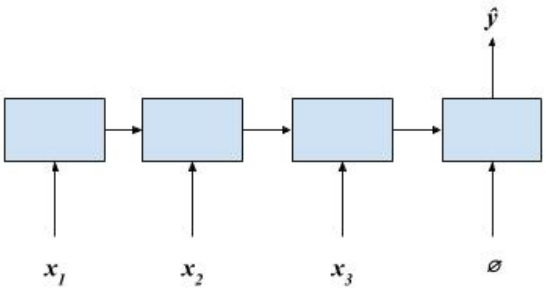


Gambar 4. *Many to many* pada RNN

   Pada jenis ini jumlah output yang dihasilkan tidak harus sama dengan jumlah input, dan biasanya model akan menghasilkan output hanya setelah semua input selesai diproses, agar model dapat mengetahui sebanyak mungkin konteks untuk menghasilkan output. Contoh penggunaan jenis ini adalah translasi bahasa dan pengenalan suara.

2. *Many to One* (Banyak Ke Satu)
   Pada jenis pemrosesan ini, model memproses banyak input, tapi output yang dihasilkan hanya satu.

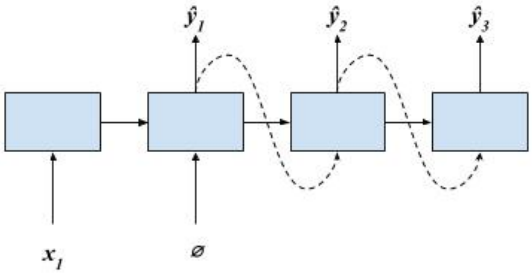
 https://doi.org/10.56211/sudo.v3i3.583



Gambar 5. *Many to one* pada RNN

Contoh penggunaan jenis ini adalah deteksi sentimen dan nilai ulasan film otomatis. Pada aplikasi deteksi sentimen misalnya, input berupa kata-kata dan model akan memberikan satu output berupa nilai sentimen yang sesuai.

3. *One To Many* (Satu Ke Banyak)
   Pada jenis pemrosesan ini, model hanya membutuhkan satu input saja, namun bisa menghasilkan banyak output.



Gambar 6. *One to many* pada RNN

Penggunaan arsitektur jenis ini terutama pada aplikasi sintesa, misalnya sintesa musik dan sintesa esai. Pada sintesa musik, inputnya hanya satu misalnya jenis music (jazz, dangdut, dan lain sebagainya), lalu model akan mensintesa nada demi nada sehingga menghasilkan musik yang dapat dinikmati.

METODOLOGI

Analisa

Analisa ini penulis dapatkan dari hasil selisih waktu yang dibutuhkan antara *input* suara ke dalam aplikasi secara langsung (dalam jarak dekat) dengan *input* suara rekaman dan antara *input* suara ke dalam aplikasi secara langsung (dalam jarak jauh) *input* suara rekaman.

Contoh:

Teks yang akan diinput ialah “Halo selamat siang”. Hasil selisih waktu antara penginputan dibawah ini ialah

a. *Input* suara ke dalam aplikasi secara langsung (dalam jarak dekat) = 0.87 detik
b. *Input* suara rekaman = 0.67 detik
c. *Input* suara ke dalam aplikasi secara langsung (dalam jarak jauh) = 1.27 detik

Jawab:

Diketahui:

Maksimal kata = 250 kata

Jumlah kata yang dihasilkan = 8 kata

Waktu input secara langsung (jarak dekat) = 0.87 detik
Waktu input melalui file rekaman = 0.67 detik
↔ Selisih antara keduanya ialah 0.2 detik

Maka: $\frac{250}{8} \times 0.2 = 6.25$ detik; jika *input* suara dengan jarak dekat

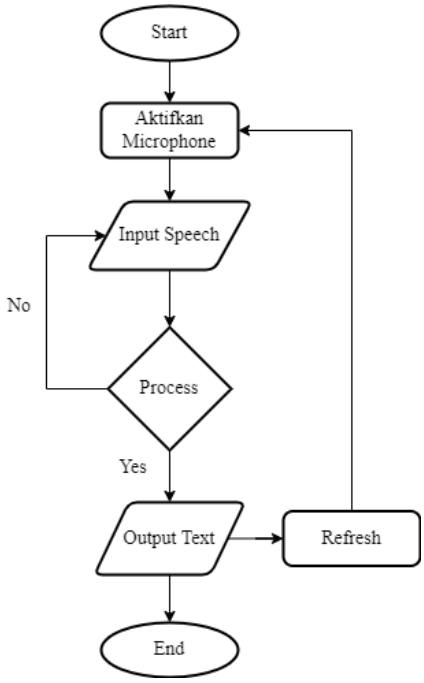
Waktu input secara langsung (jarak jauh) = 1.27 detik
Waktu input melalui file rekaman = 0.67 detik
↔ Selisih antara keduanya ialah 0.6 detik

Maka: $\frac{250}{8} \times 0.6 = 18.75$ detik; jika *input* suara dengan jarak jauh

Kesimpulannya ialah semakin jauh jarak *input* suara maka semakin tinggi juga nilai selisih waktu yang dibutuhkan untuk menghasilkan sebuah *output*.

Flowchart

Flowchart Pada Aplikasi Speech to Text Online



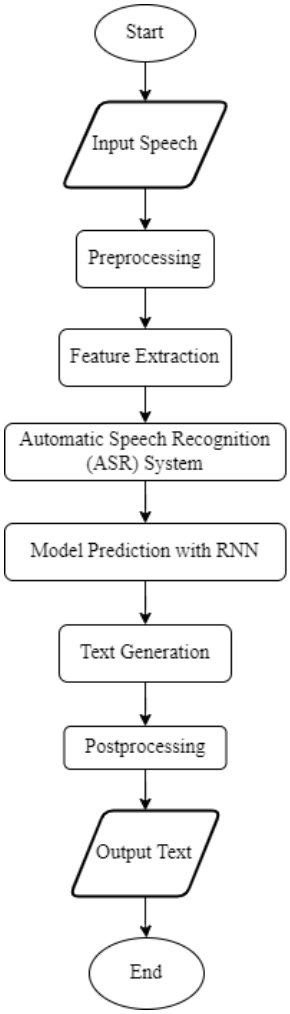
Gambar 7. *Flowchart* Pada Aplikasi *Speech to Text Online*

Tahapan pada proses di atas adalah sebagai berikut:

1. Aktifkan *microphone*: Mengaktifkan microphone agar dapat menerima input suara.
2. *Input* Suara: Menginput suara melalui microphone saat berbicara.
3. *Process*: Memproses suara untuk mendeteksi apakah suara terdeteksi atau tidak. Jika terdeteksi, suara akan diubah menjadi teks; jika tidak, proses input suara akan diulang.
4. *Output text*: Menampilkan teks secara otomatis jika suara terdeteksi.
5. *Refresh*: Mengulang proses kembali ke tampilan awal jika diinginkan oleh pengguna.

 https://doi.org/10.56211/sudo.v3i3.583

Flowchart Pada Recurrent Neural Network (RNN)



Gambar 8. *Flowchart* Pada *Recurrent Neural Network* (RNN)

Tahapan proses di atas adalah sebagai berikut:

1. *Input Speech*: Merekam suara atau mengambil rekaman suara dari file (misalnya mp3 atau wav).
2. *Preprocessing*: Mengolah data awal agar siap digunakan dalam proses selanjutnya.
3. *Feature Extraction*: Melakukan ekstraksi fitur suara.
4. *Automatic Speech Recognition (ASR) System:* Mengubah suara menjadi teks.
5. *Model Prediction with RNN*: Memproses urutan fitur suara yang telah diekstraksi dan mengonversinya menjadi urutan teks yang paling mungkin.
6. *Text Generation:* Hasil dari model RNN diubah menjadi teks awal.
7. *Postprocessing*: Mengoreksi teks menggunakan model bahasa atau aturan ejaan serta memformat teks agar mudah dibaca.
8. *Output Text:* Menampilkan teks yang dihasilkan dari model RNN sebagai output.

***Ilustrasi Alur Kerja Pada Recurrent Neural Network* (RNN)**

Berikut ilustrasi bagaimana cara kerja *Recurrent Neural Network* (RNN) dalam menghasilkan teks melalui suara dengan bantuan teknologi *Automatic Speech Recognition* (ASR):

Contoh:

1. *Input Speech*: “Selamat Pagi”
2. *Preprocessing*: Ekstraksi fitur
3. *RNN Processing*:
    - Frame 1: “S”
    - Frame 2: “e”

- Frame 3: “l”
- Frame 4: “a”
- Frame 5: “m”
- Frame 6: “a”
- Frame 7: “t”
- Frame 8: “P”
- Frame 9: “a”
- Frame 10: “g”
- Frame 11: “i”

4. *Decoding*: Menghasilkan urutan karakter: “S”, “e”, “l”, “a”, “m”, “a”, “t”, “P”, “a”, “g”, “i”
5. *Post-processing:* Menggabungkan dan memperbaiki teks menjadi: “Selamat Pagi”

HASIL DAN PEMBAHASAN

Pengujian Pada Blackbox Testing

Pengujian yang dilakukan pada penelitian ini ialah dengan menggunakan *blackbox testing*. *Blackbox testing* adalah metode pengujian perangkat lunak di mana penguji hanya akan fokus pada input dan output dari perangkat lunak.

Tabel 1. Hasil pada *Blackbox Testing*

No.	Skenario Uji Coba	Input	Langkah Uji	Hasil Yang Diharapkan	Status	Keterangan
1.	Pengujian dengan input suara jelas	Audio berisi kalimat sederhana dan jelas. Contoh: “Selamat Pagi”	Input suara ke dalam aplikasi	Selamat pagi	Valid	Input dan Output sesuai
			Menggunakan file rekaman suara	Teks yang dihasilkan sesuai dengan kalimat yang diucapkan	Valid	Input dan Output sesuai
2.	Pengujian dengan noise di latar belakang	Audio dengan kalimat yang sama dan noise rendah. Contoh: “Selamat Pagi”	Input suara ke dalam aplikasi	Selamat pagi	Valid	Input dan Output sesuai
			Menggunakan file rekaman suara dengan noise di latar belakang	Teks yang dihasilkan sesuai dengan kalimat yang diucapkan	Valid	Input dan Output sesuai
3.	Pengujian dengan aksen yang berbeda	Audio dengan kalimat sederhana dengan aksen. Contoh: “Selamat Pagi”	Input suara ke dalam aplikasi	Selamat Pagi	Invalid	Output tidak keluar
			Menggunakan file rekaman suara dengan aksen tertentu	Teks yang dihasilkan sesuai dengan kalimat yang diucapkan dengan sedikit atau tanpa kesalahan	Invalid	Output tidak keluar
4.	Pengujian dengan kata-kata yang ambigu atau homofon	Audio dengan kata-kata yang memiliki bunyi yang sama (homofon).	Input suara ke dalam aplikasi	“Bank Bunga” dan “Hai Bang”	Valid	Input dan Output sesuai
			Menggunakan file rekaman suara dengan mengandung	Teks yang diucapkan dapat membedakan kata-kata yang ambigu	Valid	Input dan Output sesuai

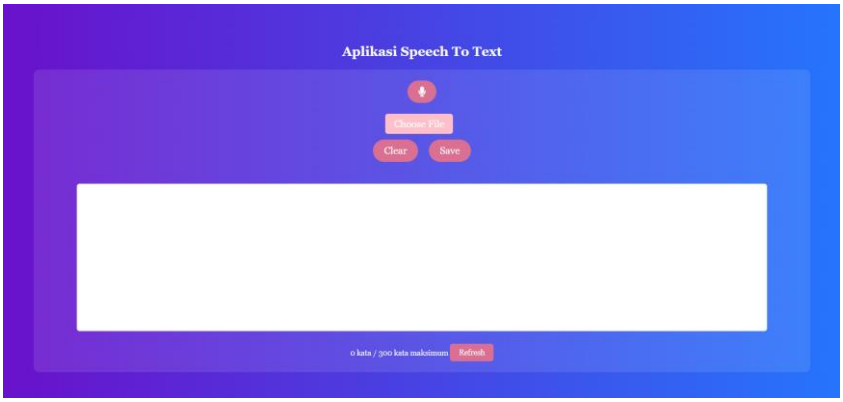
 https://doi.org/10.56211/sudo.v3i3.583

		Contoh: "Bank Bunga" dan "Hai Bang"	kata-kata yang ambigu	berdasarkan konteks		
5.	Pengujian dengan kalimat panjang	Audio dengan kalimat panjang. Contoh: "Cuaca hari ini sangat sejuk, tidak seperti cuaca di hari-hari biasanya yang selalu panas"	Input suara ke dalam aplikasi	Cuaca hari ini sangat sejuk, tidak seperti cuaca di hari-hari biasanya yang selalu panas	Valid	Input dan Output sesuai
			Menggunakan file rekaman suara dengan kalimat panjang	Teks yang dihasilkan mencerminkan seluruh teks tanpa terpotong	Valid	Input dan Output sesuai
6.	Pengujian dengan bahasa yang berbeda	Audio dalam bahasa selain bahasa utama aplikasi. Contoh: "What are you doing here?"	Input suara ke dalam aplikasi	What are you doing here?	Valid	Input dan Output sesuai
			Menggunakan file rekaman suara dengan bahasa yang berbeda dari bahasa utama pada aplikasi	Teks yang dihasilkan mencerminkan bahasa input	Valid	Input dan Output sesuai
7.	Pengujian dengan suara pelan	Audio yang diiput dengan volume suara yang kecil. Contoh: "Halo apa kabar"	Input suara ke dalam aplikasi	Halo apa kabar	Valid	Input dan Output sesuai
			Menggunakan file rekaman suara dengan volume suara yang kecil	Teks yang dihasilkan mencerminkan bahasa input	Valid	Input dan Output sesuai
8.	Pengujian dengan suara keras	Audio yang diinput dengan volume suara yang besar. Contoh: "Halo apa kabar"	Input suara ke dalam aplikasi	Halo apa kabar	Valid	Input dan Output sesuai
			Menggunakan file rekaman suara dengan volume suara yang besar	Teks yang dihasilkan mencerminkan bahasa input	Valid	Input dan Output sesuai
9.	Pengujian dengan tidak ada suara	Tidak ada suara yang diinput	Input suara ke dalam aplikasi	Teks tidak keluar	Valid	Teks tidak keluar
			Menggunakan file rekaman suara dengan tidak ada suara	Teks tidak keluar	Valid	Teks tidak keluar
10.	Pengujian dengan multiple voice	Audio input beberapa suara dalam	Input suara ke dalam aplikasi	Teks dapat memisahkan ucapan dua orang	Invalid	Teks keluar namun tidak dapat

		waktu yang sama		atau lebih dengan benar		memisahkan ucapan dua orang atau lebih
11.	Pengujian saat berada pada lingkungan yang bising	Audio input saat berada diruangan yang bising. Contoh: "Saya di dalam ruangan kerja"	Input suara ke dalam aplikasi	Saya di dalam ruangan kerja	Invalid	Teks tidak keluar
			Menggunakan file rekaman suara dalam lingkungan yang bising	Teks yang dihasilkan sesuai dengan input dalam rekaman	Invalid	Teks keluar, namun tidak sesuai dengan suara rekaman
12.	Kecepatan Suara konversi ke dalam Teks	Input audio dalam 0,5 detik. Contoh: "Halo apa kabar"	Input suara ke dalam aplikasi	Halo apa kabar	Valid	Input dan Output sesuai
			Menggunakan file rekaman suara kedalam aplikasi	Teks yang dihasilkan sesuai dengan input dalam rekaman	Valid	Input dan Output sesuai
13.	Pengujian dengan suara hewan	Input suara kucing	Input suara kedalam aplikasi	Teks tidak keluar	Valid	Teks tidak keluar
			Menggunakan file rekaman suara kucing ke dalam aplikasi	Teks tidak keluar	Valid	Teks tidak keluar
14.	Pengujian suara dengan menyebutkan angka	1 2 3 4 5	Input suara kedalam aplikasi	1 2 3 4 5	Valid	Input dan Output sesuai

Tampilan Pada Aplikasi Speech to Text Online

Tampilan Awal Antarmuka (Interface)

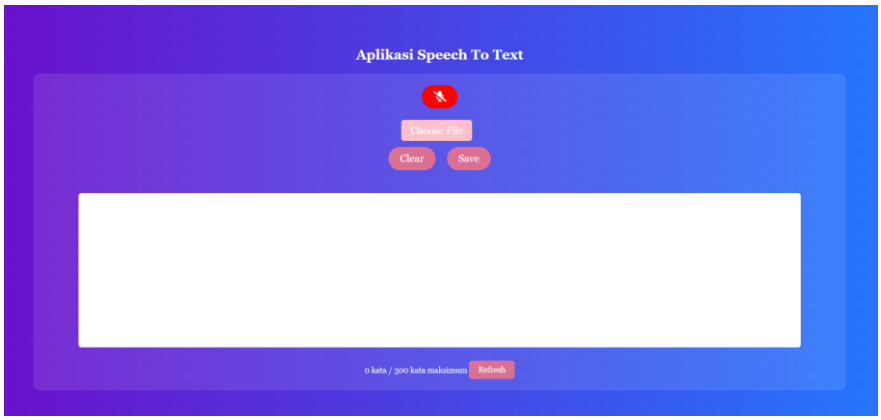


Gambar 9. Tampilan Awal Antarmuka (*Interface*)

 https://doi.org/10.56211/sudo.v3i3.583

Aplikasi *Speech to Text* memiliki tombol *microphone* untuk menginput suara menjadi teks, menu *choose file* untuk memilih file rekaman mp3 atau wav, tombol *Refresh* untuk menghapus teks dengan cepat, dan tombol *save* untuk menyimpan teks dalam format .txt.

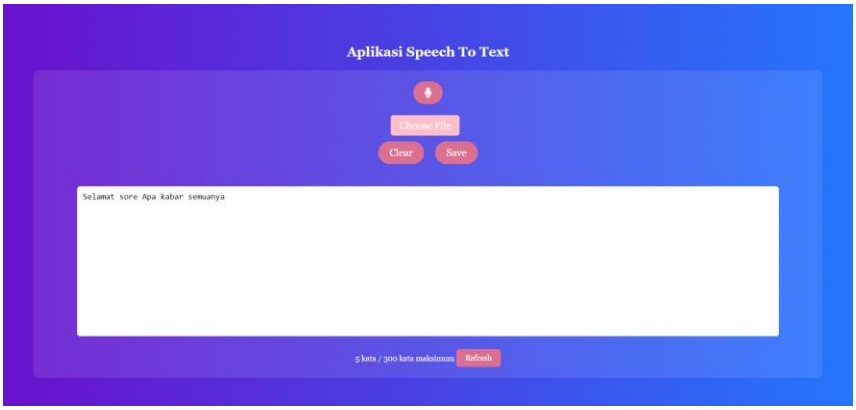
Tampilan Saat Microphone Aktif



Gambar 10. Tampilan Saat *Microphone* Aktif

Gambar diatas ialah tampilan saat *user* mengaktifkan tombol *microphone* dimana secara otomatis *microphone* akan berubah menjadi tombol *mute* dan teks akan langsung ditampilkan secara otomatis setelah selesai berbicara.

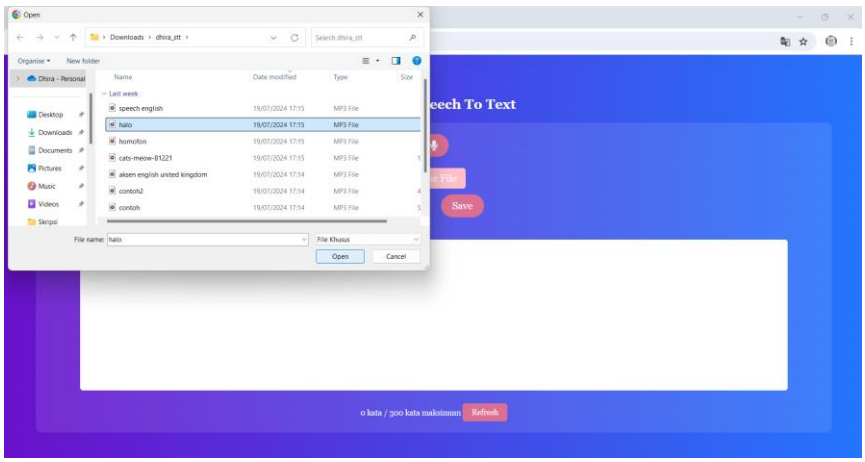
Tampilan Saat Microphone Non-Aktif



Gambar 11. Tampilan Saat *Microphone* Non-Aktif

Gambar diatas ialah tampilan saat *user* mengnon-aktifkan tombol *microphone* apabila telah selesai berbicara, maka secara otomatis teks akan muncul dan *microphone* kembali ke awal.

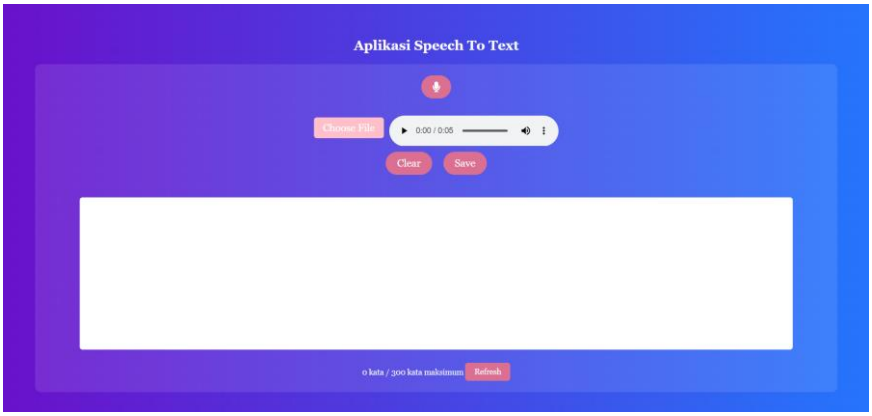
Tampilan Saat Memilih File Record



Gambar 12. Tampilan Saat Memilih *File Record*

Gambar diatas ialah tampilan saat *user* memilih *file record* (file rekaman) suara dalam bentuk mp3 yang terdapat pada *folder* dalam PC atau Laptop. Sebagai contoh file 'halo.mp3' yang akan dipilih, lalu klik open.

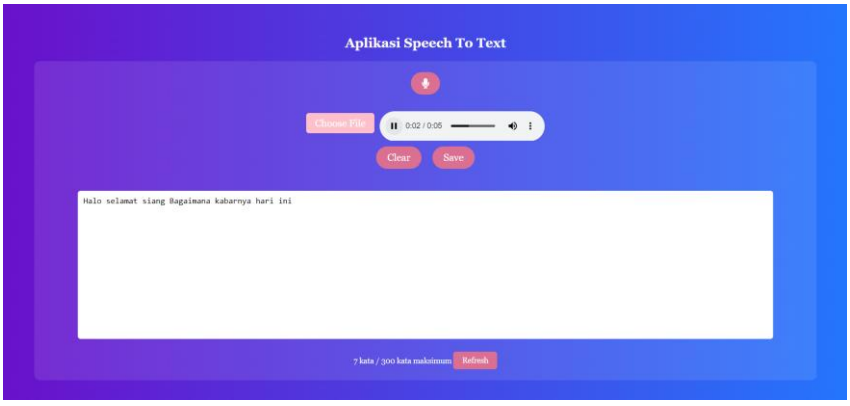
Tampilan Setelah Memilih File Record



Gambar 13. Tampilan Setelah Memilih *File Record*

Gambar diatas ialah tampilan saat *user* telah memilih file rekaman 'halo.mp3', maka secara otomatis akan adanya tampilan *button song bar*.

Tampilan Saat Memilih Tombol Play Pada File Record

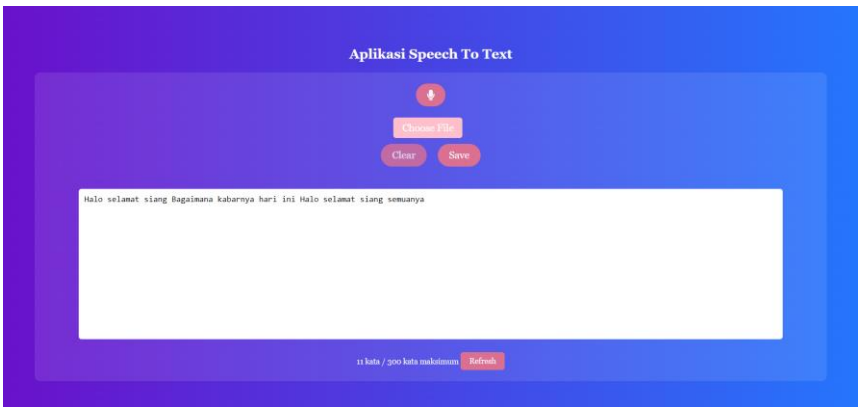


Gambar 14. Tampilan Saat Memilih Tombol *Play* Pada *File Record*

Gambar diatas ialah tampilan saat *user* memilih tombol *play* yang terdapat pada *button song bar*. Pada saat rekaman suara mulai diputar, maka secara otomatis teks akan ditampilkan setelah rekaman selesai diputar.

Tampilan Saat Menghapus File Record

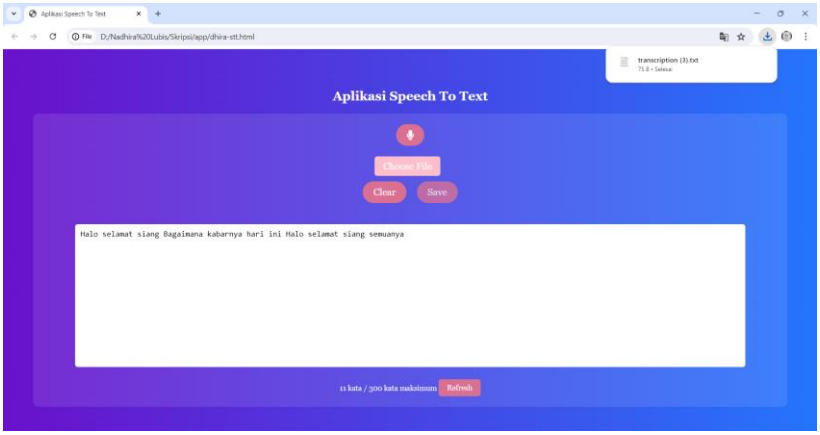
 https://doi.org/10.56211/sudo.v3i3.583



Gambar 15. Tampilan Saat Menghapus *File Record*

Gambar diatas ialah tampilan saat *user* memilih tombol *clear* (hapus), maka secara otomatis file ‘halo.mp3’ akan terhapus tetapi teks yang telah ditampilkan tidak akan terhapus.

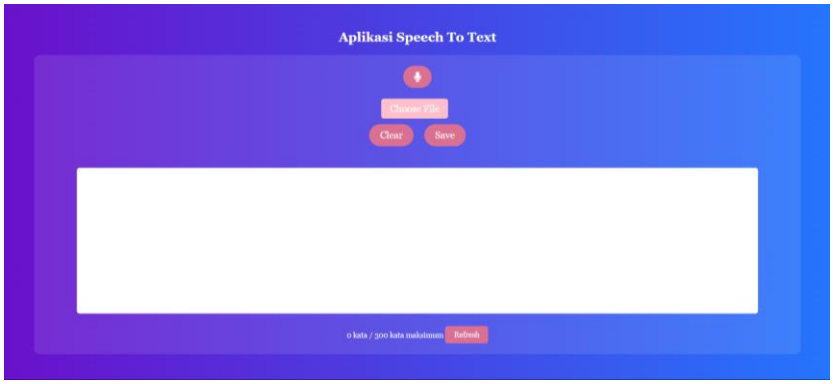
Tampilan Saat Memilih Save



Gambar 16. Tampilan Saat Memilih *Save*

Gambar diatas ialah tampilan saat *user* memilih tombol *save*, maka secara otomatis teks akan tersimpan dalam format berekstensi .txt**.**

Tampilan Saat Memilih Refresh



Gambar 17. Tampilan Saat Memilih *Refresh*

Gambar diatas ialah tampilan saat *user* memilih tombol *refresh*, maka secara otomatis tampilan akan balik seperti tampilan awal**.** Tombol ini berfungsi untuk mempermudah *user* jika ingin menghapus teks yang telah ada sebelumnya.

KESIMPULAN DAN SARAN

Kesimpulan

Aplikasi *"Speech to Text Online"* merupakan aplikasi *Online* yang dirancang agar dapat mengubah suara pada manusia menjadi sebuah teks dalam komputer. Aplikasi ini menggunakan *Speech Recognition*, yang merupakan bagian dari *Web Speech API*, untuk mengubah suara menjadi teks secara otomatis. *Web Speech API* ini sangat membantu dalam proses pengembangan aplikasi, membuatnya lebih interaktif, terutama pada perangkat yang memiliki *microphone*. Dalam pembuatan aplikasi *Speech to Text* ini juga menggunakan Algoritma *Deep Learning*, khususnya metode *Recurrent Neural Network* (RNN). *Recurrent Neural Network* (RNN) memainkan peran penting dalam proses ini dengan kemampuannya untuk memproses data urutan waktu, seperti sinyal suara, dengan mempertahankan konteks dari informasi sebelumnya. Hal ini membuat *Recurrent Neural Network* (RNN) untuk mengenali dan mengonversi ucapan menjadi teks secara lebih akurat, terutama dalam menangani tantangan pengenalan ucapan seperti variasi intonasi, pengurangan pada *noise*, aksen, dan kecepatan berbicara. Implementasi *Recurrent Neural Network* (RNN) dalam aplikasi ini meningkatkan akurasi dan kecepatan proses konversi ucapan menjadi teks, memberikan hasil yang lebih baik dibandingkan metode konvensional. Berdasarkan hasil *blackbox testing*, fungsionalitas aplikasi yang telah dibuat sudah sesuai yang diharapkan. Aplikasi ini diharapkan dapat membantu pengguna dalam berbagai bidang, seperti virtual assistans, penerjemah, sistem pengolahan audio, rekam medis, dan lain sebagainya**.**

Saran

Setelah dilakukannya penelitian implementasi algoritma *Deep Learning* pada aplikasi *Speech to Text Online* maka saran yang didapat ialah: Aplikasi *Speech to Text Online* ini dapat dikembangkan agar dapat menghapus beberapa kalimat yang diinginkan setelah dilakukannya konversi suara ke teks. Aplikasi *Speech to Text Online* ini dapat dikembangkan dengan memberikan penambahan fitur seperti pengenalan nama khusus (misalnya nama orang, nama tempat, dan lain sebagainya), penyesuaian konteks, kalimat homofon, dan dukungan untuk berbagai bahasa dan dialek.

DAFTAR PUSTAKA

[1] T. P. Laksono, "Speech To Text Untuk Bahasa Indonesia," *Skripsi*, 2018.
[2] M. E. Prastyo, "Aplikasi Text to Speech Berbasis Javascript," *Visualika*, vol. 7, no. 1, pp. 89–101, 2022.
[3] A. B. Jaman and A. Fergina, "Implementasi Speech Recognition Berbasis Android Dalam Optimalisasi Komunikasi Bagi Penyandang Tunarungu," *Jurnal Teknik Informatika UNIKA Santo Thomas*, vol. 06, no. 21, pp. 373–378, 2021, doi: 10.54367/jtiust.v6i2.1508.
[4] I. Cholissodin, Sutrisno, A. A. Soebroto, U. Hasanah, and Y. I. Febiola, "AI, Machine Learning & Deep Learning (Teori & Implementasi) 'from Basic Science to High Scientific Solution for Any Problem' Versi 1.01," p. 317, 2020.
[5] D. Y. H. dan D. T. Wahyono, *DASAR-DASAR DEEP LEARNING DAN IMPLEMENTASINYA*. PENERBIT GAVA MEDIA, 2021.
[6] Malau dan Nurjaman, "Kecerdasan Buatan," *Journal of Chemical Information and Modeling*, vol. 53, no. 9, pp. 8–24, 2019.
[7] R. Cahyadi *et al.*, "Recurrent Neural Network (RNN) Dengan Long Short Term Memory (LSTM) Untuk Analisis Sentimen Data Instagram," *Jurnal Informatika dan Komputer*, vol. 5, no. 1, pp. 1–9, 2020.
[8] E. Lapian, A. . Osmond, R. . Saputra, and E. Al., "RNN Dialek Manado," *Proceeding of Engineering*, vol. 5, no. 3, pp. 3–4, 2018.
[9] P. Ramadhika, "Deep Learning," *Gastronomía ecuatoriana y turismo local.*, vol. 1, no. 69, pp. 5–24, 2019.
[10] B. Prijono, "Pengenalan Recurrent Neural Network (RNN) - Bagian 1," *indoml.com*, 2018. .