



SNESTIK

Seminar Nasional Teknik Elektro, Sistem Informasi,
dan Teknik Informatika

<https://ejurnal.itats.ac.id/snestik> dan <https://snestik.itats.ac.id>



Informasi Pelaksanaan :

SNESTIK I - Surabaya, 26 Juni 2021

Ruang Seminar Gedung A, Kampus Institut Teknologi Adhi Tama Surabaya

Informasi Artikel:

DOI : 10.31284/p.snestik.2021.1690

Prosiding ISSN 2775-5126

Fakultas Teknik Elektro dan Teknologi Informasi-Institut Teknologi Adhi Tama Surabaya
Gedung A-ITATS, Jl. Arief Rachman Hakim 100 Surabaya 60117 Telp. (031) 5945043
Email : snestik@itats.ac.id

Software Bantu Pembuka Aplikasi Pada Dekstop Berbasis Voice Recognition

Akuwan Saleh¹, Rido Akbar I², Hari Wahjuningrat S³

^{1,2,3}Program Studi Teknik Telekomunikasi, Departemen Teknik Elektro,
Politeknik Elektronika Negeri Surabaya
e-mail: akuwan@pens.ac.id

ABSTRACT

The human desire to make it easier to use a tool in order to increase the effectiveness of work production becomes a guide in the world of work. However, physical limitations or physical disabilities, especially the hands, make it difficult for someone to work using these tools. Therefore an assistive software was made to solve these problems, by utilizing the recognition of natural human characteristics using voice or voice recognition. Assistive software opening applications on a computer desktop with voice commands which are then converted into text using processing software with the help of the websocket and speech-to-text library from Google and the results are processed to open the desired application by the user. In this paper, an application that has been made is able to convert voice to text and can open applications on the desktop so that it makes it easier for users to operate applications on a computer with a system success rate for speech recognition reaching 91.5%. The time taken to open the application is 2 to 5 ms.

Keywords: Assistive software; desktop application; voice recognition; speech-to-text.

ABSTRAK

Keinginan manusia untuk mempermudah penggunaan suatu alat agar bisa meningkatkan efektivitas produksi kerja menjadi tuntutan dalam dunia kerja. Namun keterbatasan fisik atau cacat fisik khususnya tangan menyebabkan seseorang mengalami kesulitan untuk bekerja menggunakan alat tersebut. Oleh karena itu dibuatlah suatu software bantu untuk memecahkan masalah tersebut, dengan memanfaatkan pengenalan karakteristik alami manusia menggunakan suara atau voice recognition. Software bantu pembuka aplikasi pada desktop komputer dengan perintah suara yang selanjutnya diubah dalam bentuk teks menggunakan software processing dengan bantuan library websocket dan speech-to-text dari google dan hasilnya diproses untuk membuka aplikasi yang diinginkan oleh user. Pada paper ini, aplikasi yang telah dibuat

mampu mengkonversi suara menjadi teks dan dapat membuka aplikasi pada dekstop sehingga mempermudah user dalam pengoperasian aplikasi pada komputer dengan tingkat keberhasilan sistem untuk pengenalan ucapan mencapai 91,5 % diuji terhadap pengaruh lingkungan dalam keadaan tenang dan berisik. Waktu yang dibutuhkan untuk membuka aplikasi adalah 2 hingga 5 ms.

Kata kunci: Software bantu; aplikasi desktop; voice recognition; speech-to-text.

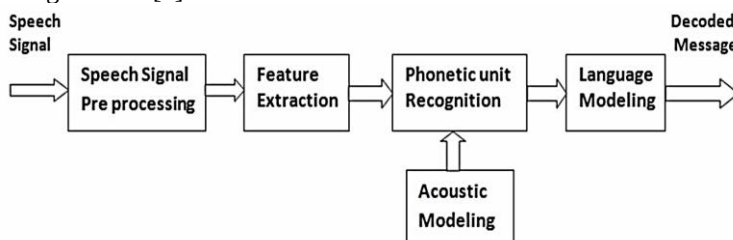
PENDAHULUAN

Penggunaan teknologi komputer membuat segalanya menjadi lebih efisien dan praktis. Tetapi ada beberapa masalah yang menyebabkan user mengalami kesulitan untuk menggunakan komputer. Masalah lain yang muncul dalam perkembangan teknologi komputer adalah kesulitan user atau pengguna komputer untuk menggunakan sejumlah tools yang terdapat di suatu aplikasi komputer, terutama bagi pengguna cacat fisik. Untuk memecahkan masalah tersebut, pengenalan karakteristik alami manusia dapat digunakan, misalkan suara. Pemrosesan ucapan atau suara banyak digunakan di beberapa aplikasi seperti perangkat keamanan, peralatan rumah tangga, telepon seluler, mesin ATM dan komputer. Antarmuka komputer manusia telah dikembangkan untuk berkomunikasi atau berinteraksi dengan nyaman bagi orang yang menderita beberapa jenis kecacatan[1]. Pemrosesan ucapan telah dimanfaatkan dalam penelitian sebelumnya yang dilakukan oleh Ayu D M [10] tentang Speech Recognition dengan query berbahasa Indonesia untuk pengiriman sms, dengan teknik pengenalan suara ke teks menggunakan Application Programming Interface (API) speech recognition google yang terdapat dalam smartphone android. Penelitian berikutnya dilakukan oleh M Fikri S [11] tentang Rancang bangun aplikasi voice translator berbasis android menggunakan hidden markov model. Dalam penelitian ini dibuat sebuah voice translator yang mampu menterjemahkan hasil konversi suara ke teks dalam Bahasa Inggris menggunakan algoritma hidden markov model.

Voice Recognition untuk keamanan PC juga telah dilakukan penelitian oleh Destian dkk [12] pada penelitiannya sistem keamanan menggunakan suara untuk mengunci desktop dari orang yang tidak memiliki otoritas. Nilai amplitudo diambil dari sinyal suara masukan, sehingga didapatkan kumpulan angka real yang menjadi nilai masukan untuk ekstraksi ciri. Metode ekstraksi ciri yang digunakan adalah MFCC dan DTW. Bila dibandingkan dengan person *identification* yang lain, pengenalan suara pembicara (*speaker recognition*) tidak membutuhkan biaya yang besar dan tidak membutuhkan peralatan khusus. Sistem yang disebut *voice command*, sebuah sistem yang menggunakan suara sebagai kunci utama untuk mengenali perintah dalam bahasa mesin yang kemudian perintah itu dijalankan. Adanya aplikasi yang mampu mengkonversi suara menjadi teks atau *speech-to-text* (STT) untuk membuka aplikasi pada dekstop dapat mempermudah user dalam pengoperasian komputer.

Pengenalan Suara (*speech recognition*)

Speech Recognition atau yang biasa dikenal dengan *automatic speech recognition* (ASR) merupakan suatu pengembangan teknik dan sistem yang memungkinkan komputer untuk menerima masukan berupa kata yang diucapkan. Skema utama sistem pengenalan ucapan ditunjukkan pada gambar 1 [2].



Gambar 1. Blok diagram sistem *speech recognition*

Speech recognition merupakan sistem pengenalan pola ucapan yang memiliki kemampuan untuk mencocokkan suara dengan data yang ada pada rekaman maupun kumpulan pembendaharaan kata [3]. *Speech recognition* diterapkan pada aplikasi yang dibuat sebagai pengubah suara menjadi teks, misalnya penekanan tombol pada telepon genggam yang dilakukan secara otomatis dengan komando suara. Salah satu penerapan *speech recognition* yang terkenal saat ini yaitu munculnya aplikasi *google voice search* yang dirilis oleh google untuk iphone yang memanfaatkan data centers untuk menghitung sejumlah besar analisis data yang diperlukan untuk mencocokkan permintaan pengguna dengan contoh-contoh nyata ucapan manusia [4]. Contoh aplikasi untuk memudahkan memanggil penyandang tunarungu telah dirancang dan diimplementasikan sebuah aplikasi mengubah suara menjadi teks. Aplikasi ini bekerja jika ada suara yang memanggil nama penyandang tunarungu, maka smartphone milik penyandang tunarungu dapat memberikan notifikasi getar. Untuk mengimplementasikan aplikasi tersebut perlu menggunakan library *pocketsphinx*, sebagai pengubah suara menjadi teks [8]. Tujuan utama *speech recognition* adalah untuk mencapai komunikasi bahasa yang lebih baik antara manusia dan mesin [5].

Processing

Lingkungan Pemrograman Processing terdiri dari teks editor terintegrasi dan jendela tampilan untuk menampilkan program. Jika tombol “run” ditekan, program akan mengkompilasi dan berjalan di jendela grafis. Dari jendela lingkungan utama, bisa menjalankan, menghentikan, menyimpan, membuka, dan mengekspor file [6]. Pada paper ini processing digunakan sebagai *websocket* server yang menerima konversi suara ke text dari *websocket client* yang selanjutnya dari text tersebut dicocokkan dengan source code perintah untuk membuka aplikasi yang ada pada processing sehingga aplikasi dapat terbuka. Agar *webclient* dapat berkomunikasi dengan *webserver* diperlukan library *websocket* yaitu “import *websocket*” pada pemrograman processing. *Websockets* sendiri merupakan teknologi yang menyediakan jalur tersendiri dalam mendukung komunikasi dua arah (full-duplex / bi-directional) dalam satu protokol TCP [7]. Dengan menggunakan mekanisme koneksi TCP tunggal, *websocket* dapat membangun satu jalur koneksi antara server dan klien. Dimana *websocket* menggunakan metode *handshaking* untuk membuat koneksi terbuka antara server dan klien [9].

Rencana Analisa Performansi Sistem

Pada proses analisa data digunakan suara uji untuk suara yang sudah ditentukan sebagai data latih hasil uji data ini adalah *Result Training Data* (RTD) dan suara uji data random hasil data ini adalah *Result Random Data* (RRD). Setiap suara diuji untuk mengetahui apakah suara teridentifikasi dengan benar ke dalam klasifikasi yang sebenarnya atau tidak.

Result Training Data (RTD)

Merupakan hasil dari uji data yang telah di tentukan sebelumnya, apakah sistem tersebut dapat mengenali dengan benar, tidak mengenali dengan benar, atau tidak mengenali sama sekali. Dari sistem tersebut dapat dirumuskan dengan persamaan 1.

$$RTD = \frac{Sukses}{Jumlah\ data} \times 100 \% \quad (1)$$

Result Random Data (RRD)

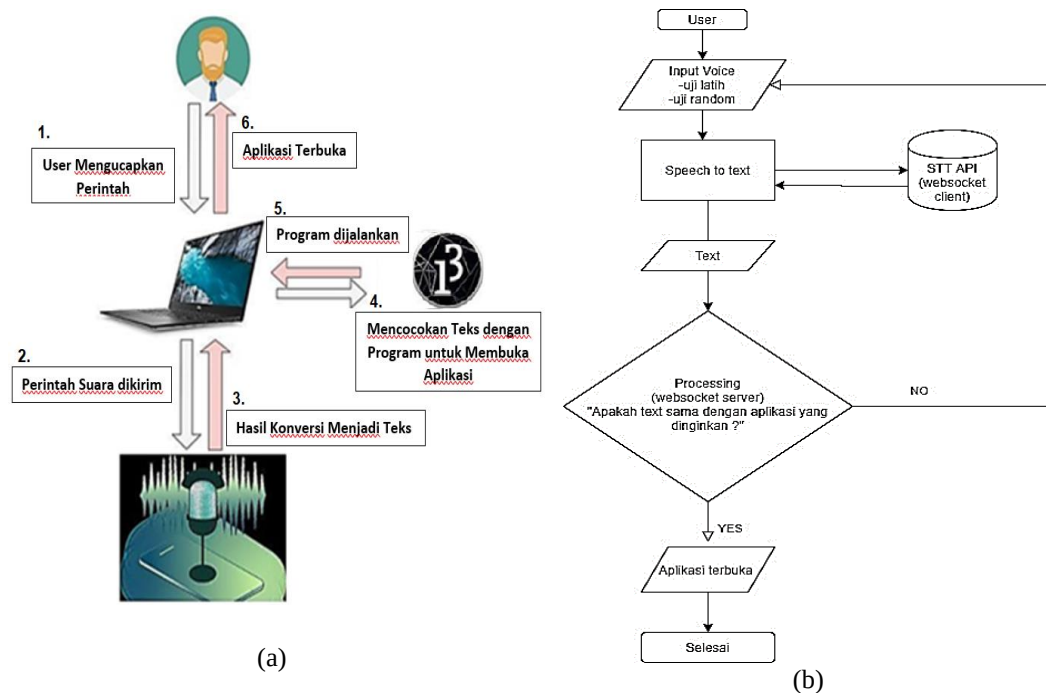
Merupakan hasil dari uji data yang sama sekali belum dikenali oleh sistem sehingga sistem diharapkan dapat mengenali dengan baik data yang dimasukan walaupun sebelumnya belum pernah dilakukan pembelajaran tentang pola data masukan. Proses Analisa data random menggunakan persamaan 2.

$$RRD = \frac{Sukses}{Jumlah\ data} \times 100 \% \quad (2)$$

METODE

Perancangan sistem untuk software bantu sebagai pembuka aplikasi pada desktop dengan *voice recognition* dapat dilihat pada gambar 2. Pengguna atau user memberikan perintah suara lewat microphone setelah itu perintah tersebut masuk ke komputer dan di konversi ke text oleh client. lalu diterima oleh processing sebagai server. Processing mencocokkan teks yang diterima dengan kumpulan perintah-perintah untuk membuka aplikasi. Setelah cocok maka aplikasi yang diinginkan user akan terbuka. Data yang digunakan pada sistem ini terdiri atas 2 jenis data suara yaitu data uji latih dan uji random.

- Uji latih adalah data yang digunakan telah ditentukan sebelumnya. Datanya ada 2 sampel data suara untuk 1 sampel suara pria dan 1 sampel suara wanita yang mengucapkan 8 kata-kata pesan yang sama. Maka total sampel data suara sebanyak 16 sampel suara. Adapun kata-kata yang digunakan sebagai berikut: *Notepad, Microsoft Word, Chrome, Power Point, Excel, Wordpad, Music* dan *Zoom*.
- Data uji random adalah data yang belum diujikan ke sistem. Data yang digunakan pada sistem uji suara ini sebanyak 2 sampel data suara untuk 1 sampel suara pria yang berbeda dan 1 sampel suara wanita yang berbeda. Adapun kata-kata yang digunakan adalah *Adobe Reader, Open Browser, Open Explorer, Pick Up Jacket* dan *My Name is Gina*.

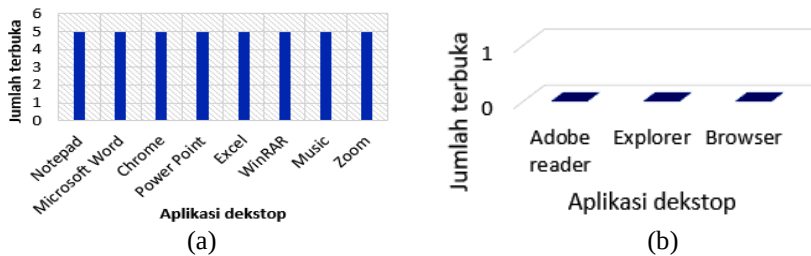


Gambar 2. a) Rancangan sistem, b) Flowchart sistem

HASIL DAN PEMBAHASAN

Membuka Aplikasi

Tujuan dari pengujian ini untuk mengetahui apakah program di processing dapat membuka aplikasi dengan tepat. Data yang diuji adalah suara uji aplikasi latih dan uji aplikasi random.



Gambar 3. a) Data aplikasi terbuka dengan uji latih, b) Data aplikasi terbuka dengan uji random
Pada gambar 3a terdapat 8 aplikasi masing-masing aplikasi diuji 5 kali sehingga jumlah sample adalah 40 dan aplikasi yang sukses terbuka sebanyak 40 maka nilai RTD adalah:

$$RTD = \frac{\text{Sukses}}{\text{Jumlah data}} \times 100\% = \frac{40}{40} \times 100\% = 100\%$$

Sedangkan pada gambar 3b terdapat 3 aplikasi masing-masing diuji 5 kali sehingga jumlah sampelnya adalah 15 dan aplikasi yang sukses tidak terbuka sebanyak 15 karena perintah suara tidak ada didatabase sistem, maka nilai RRD adalah:

$$RRD = \frac{\text{Sukses}}{\text{Jumlah data}} \times 100\% = \frac{15}{15} \times 100\% = 100\%$$

Mengenal Masukan Suara

Data yang digunakan pada sistem *Speech-to-text* ini yaitu 2 sampel data suara untuk 1 sampel suara pria yang berbeda dan 1 sampel suara wanita. Pengujian suara masukan adalah suara uji latih dan uji random. Agar masukan suara yang dikirim oleh user dapat dituliskan ke console yang ada pada software processing maka digunakan kode program sebagai berikut; `void websocketServerEvent(String msg){ println(msg);}` dan hasilnya seperti pada gambar 4.



Gambar 4. a) Masukan suara dengan uji latih, b) Masukan suara dengan uji random

Pada gambar 4a, input suara *open Zoom* baik pria maupun wanita tidak dikenali karena teks yang dihasilkan *open Xoom*, Sedangkan input suara *open Notepad* dari wanita tidak dikenali, hasil teksnya adalah *open Notebad*. Jumlah kata uji = 8, Jumlah sample = 16, Sukses = 13. Tingkat keberhasilannya: $\text{Presentase} = \frac{\text{Sukses}}{\text{Jumlah data}} \times 100\% = \frac{13}{16} \times 100\% = 81,25\%$

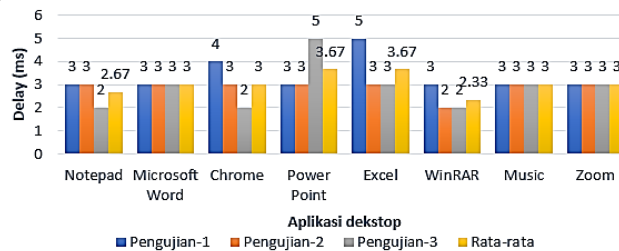
Pada gambar 4b, input suara *Pick up jacket* baik pria maupun wanita tidak dikenali karena teks yang dihasilkan *Pick up jacob*, Sedangkan input suara *open Adobe reader* dari wanita tidak dikenali, hasil teksnya adalah *open at operator*. Jumlah kata uji = 5, Jumlah sample = 10, Sukses = 7. Tingkat keberhasilannya:

$$\text{Presentase} = \frac{\text{Sukses}}{\text{Jumlah data}} \times 100\% = \frac{7}{10} \times 100\% = 70\%$$

Input suara tidak dikenali disebabkan pelafalan dan artikulasi yang kurang baik saat pengambilan sampel suara dan suara yang masuk harus sesuai dengan program *Speech Recognition* pada database Google *Speech API*.

Delay Aplikasi Terbuka

Dilakukan pengukuran waktu software bantu ini hingga aplikasi yang diinginkan terbuka, dengan tiga jumlah pengujian dan dirata-rata hasilnya.

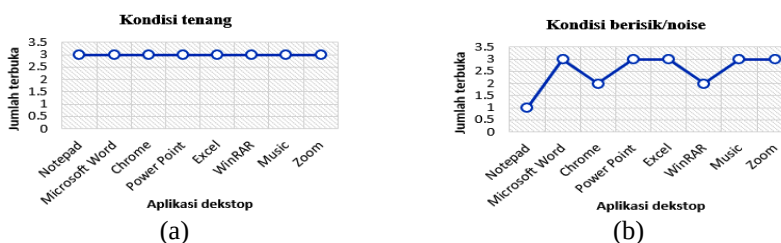


Gambar 5. Delay aplikasi terbuka

Data pada gambar 5 menunjukkan aplikasi terbuka langsung saat user mengucapkan masukan suara, dengan waktu tercepat 2 ms sedangkan waktu rata-rata tercepat adalah 2,33 ms untuk masukan suara WinRAR.

Pengaruh Kondisi Lingkungan

Dilakukan pengujian sebanyak 3 kali dengan 2 kondisi yaitu tenang dan berisik dengan menggunakan sample uji latih. Data hasil pengujian diperlihatkan pada gambar 6. Pada kondisi tidak terdapat noise aplikasi dapat terbuka 100 %, sedangkan pada kondisi terdapat noise aplikasi dapat terbuka mencapai 83 % dimana aplikasi tidak terbuka dengan perintah suara *Notepad* terjadi 2 kali dipengujian kedua dan ketiga, pengucapan *Chrome* 1 kali dipengujian kedua dan input suara *WinRAR* sekali dipengujian ketiga. Nilai total rata-rata keberhasilan aplikasi terbuka terhadap pengaruh lingkungan adalah 91,5 %.



Gambar 6. a) Aplikasi terbuka pada kondisi tenang, b) Aplikasi terbuka pada kondisi noise

Analisa Performansi Sistem

RTD (Result Training Data) : Pengujian Processing dapat membuka aplikasi = 100 %, Pengujian Processing Dapat mengenali masukan suara = 81,25 % sehingga Nilai total rata-rata keberhasilan pengujian pengaruh lingkungan dalam keadaan tenang dan berisik = 91,5 %

$$\text{Sehingga nilai } RTD \text{ TOTAL} = \frac{100+81,25+91,5}{3} = \frac{272,75}{3} = 90,92 \%$$

RRD (Result Random Data) : Pengujian Processing dapat membuka aplikasi = 0 %, Pengujian Processing dapat mengenali masukan suara = 70 %

$$\text{Sehingga nilai } RRD \text{ TOTAL} = \frac{0+70}{2} = \frac{70}{2} = 35 \%$$

KESIMPULAN

Kesimpulan berdasarkan data dan pembahasan adalah, tingkat keberhasilan sistem berdasarkan pengujian yang dilakukan pada pengujian data uji latih 90,92 % dan tingkat

keberhasilan sistem untuk pengujian data random yang belum dilatih 35 %; Tingkat keberhasilan terhadap pengaruh lingkungan dalam keadaan tenang dan berisik mencapai 91,5 %. Software bantu yang telah dibuat mampu mengkonversi suara menjadi teks dan dapat digunakan membuka aplikasi pada dekstop dengan waktu yang dibutuhkan adalah 2 hingga 5 ms sehingga software bantu ini mempermudah user dalam pengoperasian komputer. Keberhasilan dari penelitian ini menjadi dasar penelitian selanjutnya mengenai aplikasi speech recognition sebagai sistem pengendali gerakan obyek animasi.

DAFTAR PUSTAKA

- [1] Su Myat M, Hia Myo T, "Speech-To-Text Conversion (STT) System Using Hidden Markov Model (HMM)", 2015, International Journal Of Scientific & Technology Research (IJSTR), Vol 4, pp.349-352. June 2015.
- [2] Mousmita Sarma, Kandarpa Kumar S, "Acoustic Modeling of Speech Signal using Artificial Neural Network", IGI Global, pp. 287-304, June 2015, doi: 10.4018/978-1-4666-8493-5.ch012.
- [3] Jianliang Meng, Junwei Zhang and Haoquan Zhao, "Overview of the Speech Recognition Technology", International Conference on Computational and Information Sciences (ICCIS), pp. 199-202, 2012, doi: 10.1109/ICCIS.2012.202
- [4] Sani M .Isa, "Speech Recognition", 2019, <https://mti.binus.ac.id/2019/05/08/speech-recognition/> (accessed Jan. 8, 2020).
- [5] Manoj Kumar S, Omendri K, "Speech Recognition: A Review", International Journal of Advanced Networking and Applications (IJANA), pp.62-71, 2014
- [6] Florian Schulz, "Speech Recognition for Java/Processing", <https://florianschulz.info/stt/> (accessed Jan. 8, 2020).
- [7] Soewito, B., Christian, Gunawan, F. E., Diana, & Kusuma, I. G. P, "Websocket to Support Real Time Smart Home Applications", International Conference on Computer Science and Computational Intelligence (ICCSCI), pp.260-266, September 2019, doi:10.1016/j.procs.2019.09.014
- [8] Nurul H. A, Mahardeka T. A, Ratih K. D., "Pengembangan Alat Bantu Pemanggil Penyadang Tunarungu Menggunakan Library Pocketsphinx Berbasis Android (Studi Kasus PSLD Universitas Brawijaya)", Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer, vol.2, no. 8, pp. 2488-2496, Agustus 2018, e-ISSN: 2546-964X
- [9] Sunardi, Aulia N. A, Fiftin N., "Real Time Monitoring and Irrigation Control Using the Websocket Protocol", ICSTI 2018, October 19-20, Yogyakarta, Indonesia, doi: 10.4108/eai.19-10-2018.2282548
- [10] Ayu Dhiya M, "Speech Recognition Dengan Query Barbahasa Indonesia untuk Pengiriman SMS", Tugas Akhir, Departemen Teknik Informatika, Universitas Hasanudin, November 2017, Makasar.
- [11] M.Fikri S, "Rancang Bangun Aplikasi Voice Translator Berbasis Android Menggunakan Hidden Markov Model", Jurusan Teknik Informatika, Universitas Islam Negeri Maulana Malik Ibrahim, November 2016, Malang.
- [12] Destia Tri H, dan Patmi K, "Voice Recognition untuk Sistem Keamanan PC Menggunakan Metode MFCC dan DTW", Generation Journal, vol.2 No.1, Januari 2018, pp.57-68,e-ISSN: 2549-2233.

- halaman ini sengaja dikosongkan -