

Implementasi *Offline Speech Recognition* Pada *Home Device*

Wiwin Styorini^{*1}, Karina Indra Wijaya², Cyntia Widiyari³, Wahyuni Khabzli⁴

^{1,2} Teknik Elektronika, Politeknik Caltex Riau

^{3,4} Teknologi Rekayasa Jaringan Telekomunikasi

Email: ¹wiwin@pcr.ac.id, ²karina@alumni.pcr.ac.id, ³cyntia@pcr.ac.id, ⁴ayu@pcr.ac.id

Abstrak

Speech Recognition (pengenalan suara) merupakan proses untuk mengkonversikan sinyal akustik (suara) melalui microphone sebagai perintah untuk mengoperasikan komputer yang digunakan untuk mengenali suatu kata yang diucapkan oleh seseorang. Pada penelitian ini *speech recognition* (pengenalan suara) digunakan untuk mengontrol *Home Device* dengan menggunakan ekstrak ciri *Mel-Frequency Cepstral Coefficients* (MFCC) dan pemodelan *Hidden Markov Models* (HMM). MFCC merupakan teknik yang mengambil contoh suara sebagai input. Untuk dapat memproses suara yang masuk pada mikrofon, maka sinyal suara diambil cirinya. Setelah cirinya didapat maka akan dibentuk pola oleh pemodelan HMM. Tujuan digunakan metode MFCC dan HMM diharapkan dapat meningkatkan keakuratan dalam mengontrol *Home Device*. *Home device* yang digunakan dalam pengujian yaitu lampu dan kipas angin. Untuk pengontrolan tersebut dilakukan secara *offline*. Berdasarkan pengujian menggunakan 2 user didapat tingkat akurasi sebesar 95,5 % dari 200 percobaan dengan kondisi tidak ada *noise* sedangkan pengujian dengan keadaan ada *noise* didapat tingkat akurasi sebesar 7% dari 100 percobaan.

Kata kunci: *Home Device*, *Hidden Markov Models* (HMM), *Mel-Frequency Cepstral Coefficients* (MFCC), *offline*, *Speech Recognition*

Abstract

Speech Recognition is the process of converting an acoustic signal (voice) through a microphone as a command to operate a computer that is used to recognize a word spoken by someone. In this study *speech recognition* is used to control the *Home Device* by using extracts of the characteristics of *Mel-Frequency Cepstral Coefficients* (MFCC) and modeling of *Hidden Markov Models* (HMM). *Mel-Frequency Cepstral Coefficients* (MFCC) is a technique that takes sound samples as input. To be able to process the sound that enters the microphone, the voice signal is characterized. After the characteristics are obtained, the pattern will be formed by modeling *Hidden Markov Models* (HMM). The purpose of using the MFCC and HMM methods is expected to increase the accuracy of the time in controlling the *Home Device*. *Home devices* used in testing are lights and fans. For controlling done by *offline*. Based on testing obtained an accuracy rate of 95.5% from 200 experiments using 10 command words. For testing using 2 users. For testing with noise obtained an accuracy rate of 7% from 100 experiments

Keywords: : *Home Device*, *Hidden Markov Models* (HMM), *Mel-Frequency Cepstral Coefficients* (MFCC), *offline*, *Speech Recognition*.

1. Pendahuluan

Rumah menjadi kebutuhan pokok setiap orang. Rumah juga digunakan sebagai tempat berlindung dan tempat istirahat. Sebelum berkembangnya teknologi, perangkat – perangkat yang terdapat di rumah digerakkan secara konvensional seperti menekan saklar lampu secara manual. Namun seiring berkembangnya teknologi, banyak perangkat tersebut digerakkan secara otomatis khususnya berbasis IoT, akan tetapi perangkat berbasis IoT membutuhkan internet atau pengaktifan perangkat dilakukan secara *online*. Hal tersebut menjadi masalah ketika internet yang digunakan tiba-tiba mati, tidak berjalan karena faktor cuaca, dan juga membutuhkan biaya untuk membeli paket aktivisasi internet. Untuk mengatasi permasalahan tersebut, maka perangkat dapat digerakkan secara *offline* dengan menggunakan sistem pengenalan suara (*speech recognition*).

Penelitian sebelumnya terkait dengan penelitian ini yaitu menggunakan perintah suara untuk mengontrol pintu menggunakan ekstrak ciri *Linear Predictive Coding* (LPC) dan pemodelan *Hidden Markov Models* (HMM). Berdasarkan pengujian didapat tingkat keakuratan pengontrolan pintu untuk kata buka sebesar 98% dan untuk kata tutup 98% berdasarkan suara

yang terdaftar di *database*. Untuk suara yang tidak terdaftar di *database* didapat tingkat keakurasian untuk kata buka sebesar 26 % dan untuk kata tutup 27%. [1]

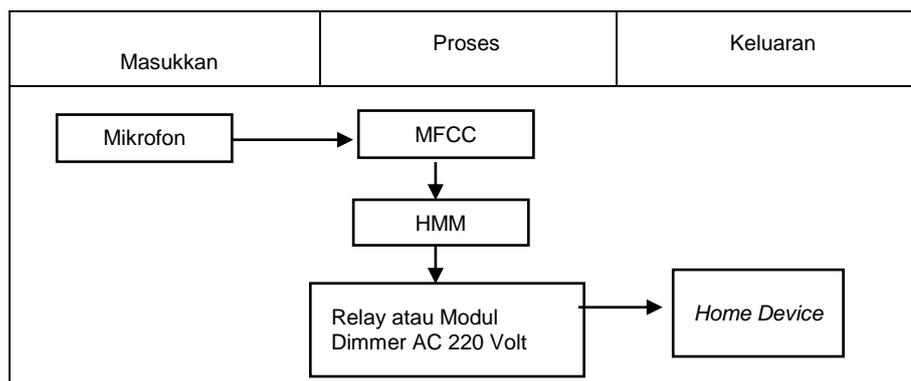
Penelitian berikutnya yaitu pembuatan sistem yang dapat mengenali kata kerja dasar yang diucapkan dan sistem dapat mengubah ucapan menjadi teks. Metode yang digunakan pada penelitian ini yaitu Metode MFCC (*Mel Frequency Cepstrum Coefficient*). Hasil yang diperoleh yaitu, tingkat keberhasilan pengujian 20 kata pada Andhika, Ilmi, Eko, dan Azza terhadap kata yang sudah disimpan pada database (data latih) sebesar 100%, pada Eko yang dimana dengan ucapan yang sama tetapi tidak ada pada database (data uji) sebesar 69%, dan pada Azza yang dimana dengan ucapan yang sama tidak ada pada database (data uji) sebesar 63%. [2]

Selanjutnya penelitian tentang penerapan Hidden Markov Model (HMM) dan Mel-Frequency Cesprtral Coefficients (MFCC) pada E-Learning Bahasa Madura untuk Anak Usia Dini, hasil dari penelitian ini adalah rata-rata akurasi untuk pengujian sistem dengan satu model yaitu 73% dengan akurasi tertinggi 75% dan rata-rata akurasi untuk pengujian sistem dengan multi model yaitu 80% dengan akurasi tertinggi 81%. [3]

Berdasarkan penelitian yang sudah ada maka pada penelitian ini penulis memanfaatkan metode MFCC dan HMM untuk mengontrol *home device*. Fokus penelitian ini adalah menghitung berapa persen akurasi yang dihasilkan dari *speech recognition* dengan menggunakan dua user untuk 200 kali percobaan dengan kondisi sekitar tidak ada *noise* serta 100 kali percobaan dengan kondisi ada *noise*.

2. Metode Penelitian

2.1. Perancangan Sistem



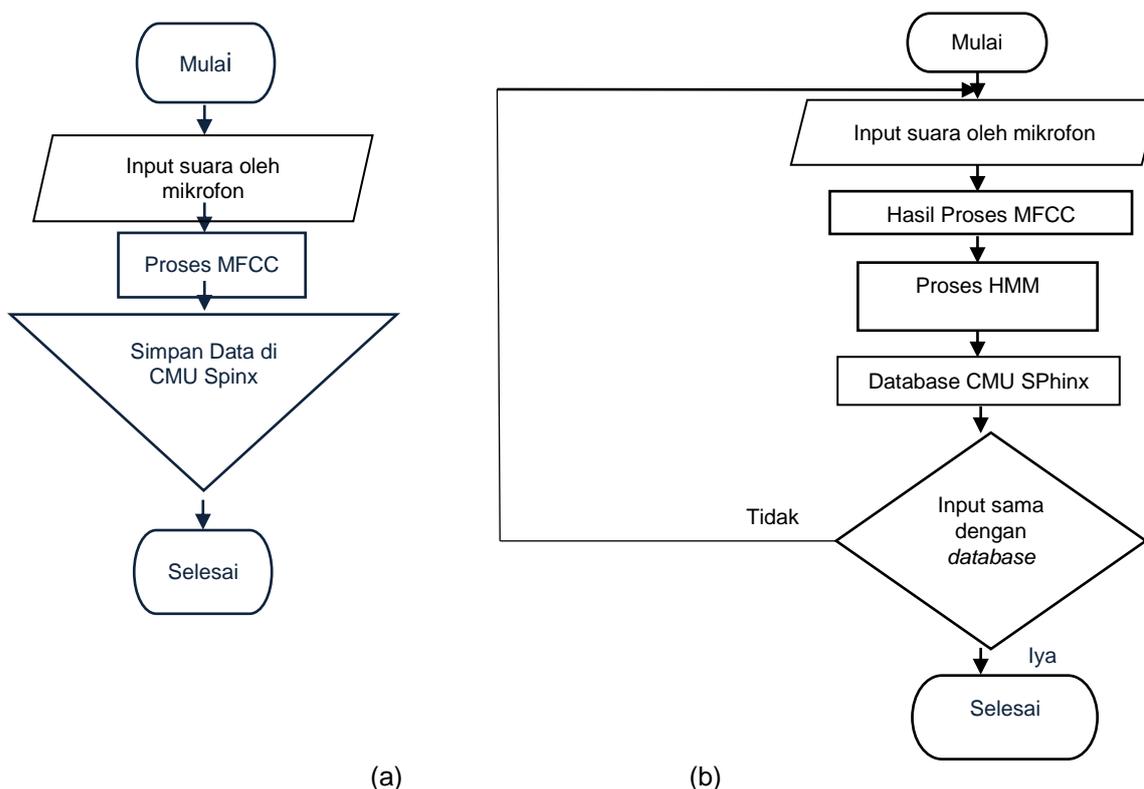
Gambar 1. Blok diagram sistem

Perancangan sistem secara umum pada penelitian ini dijelaskan pada gambar 1. Pada blok masukkan dilakukan proses input suara dari mikrofon. Input suara tersebut masih dalam bentuk sinyal analog dan mengandung noise. Untuk mendapatkan hasil yang lebih baik maka sinyal suara tersebut akan diproses pada blok proses. Blok Proses disini suara yang masuk akan diproses dengan ekstrak ciri menggunakan MFCC dan metode HMM. Proses ekstrak ciri menggunakan MFCC akan memfilter sinyal suara yang memiliki frekuensi tinggi dengan cara mengurangi nilai frekuensi sinyalnya. Proses tersebut digunakan untuk mengurangi noise dari sinyal suara. Hasil dari proses ekstrak ciri menggunakan MFCC akan menghasilkan model ciri dari sebuah kata yang telah diucapkan. Hasil tersebut akan menjadi input untuk membuat pemodelan HMM. Proses HMM akan membentuk model ciri dari sebuah kata yang selanjutnya akan dibandingkan dan di cocokkan. Kata tersebut akan dikonversikan ke dalam bentuk bilangan biner. Ketika data nya cocok maka keluarannya akan menjadi perintah untuk menggerakkan motor DC di blok keluaran. Pada Raspberry PI digunakan 4 pin output, pin tersebut digunakan untuk mengontrol *home device* pada blok keluaran sesuai perintah pada input suara. Pin tersebut berisikan logika 1 atau 0 dan kontrol PWM. perintah tersebut akan dikirimkan ke modul Relay atau Modul Dimmer AC 220 Volt. Untuk dapat mengontrol *home device* digunakan beberapa perintah suara yaitu Light On, Light Off, Light Minimal, Light Medium, Light Maximal, Fan On, Fan Off, Minimal Speed, Medium Speed, Maximum Speed.

Kemudian Blok keluaran adalah hasil akhir berupa *home device* yang dapat dikontrol sesuai dengan perintah yang di ucapkkan pada blok masukan.

2.2. Perancangan Software

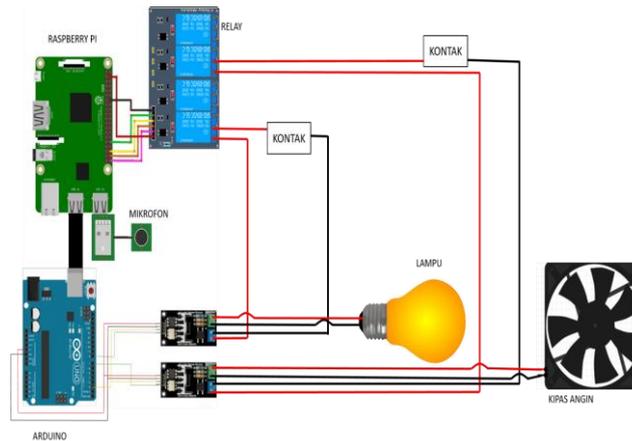
Gambar 2 merupakan alur perancangan *training* suara referensi dan pengujian *speech recognition*. Suara referensi berasal dari input mikrofon dan digunakan untuk membuat pola pengenalan. Pengambilan contoh suara referensi dilakukan sampai didapat beberapa contoh suara yang sesuai. Suara referensi tersebut akan diproses dengan ekstraksi ciri MFCC seperti yang dijelaskan pada blok diagram. Proses tersebut dilakukan agar *sample* yang didapatkan tidak tercampur dengan noise. Sample-sample yang telah melewati proses tersebut akan disimpan di CMU spinx. CMU spinx dipilih karena mendukung penyimpanan secara offline. Untuk perancangan pengujian *speech recognition* pengontrolan sistem suara dimulai saat input suara yang berasal dari mikrofon diproses dengan metode ekstraksi ciri MFCC yang ada di Raspberry Pi (RPI). Hasil dari proses tersebut akan dicocokkan dengan suara referensi yang ada di database menggunakan metode Hidden Markov Models (HMM). Jika Suara yang masuk di mikrofon cocok dengan kata yang telah ada maka sistem hardware akan bekerja. Saat input tersebut tidak cocok maka input suara dari mirofon harus diambil Kembali. Untuk Pembuatan suara referensi menggunakan spesifikasi perangkat keras Komputer Processor Intel(R) Core(TM) i3 CPU @2.20GHz RAM 6.00 Gb, dengan sistem operasi Windows 10 Professional. Perancangan *software* untuk pengujian pada deteksi suara di jalankan pada perangkat keras Raspberry Pi (RPI) tipe 3B+ dengan spesifikasi 1.4GHz 64-bit quad-core ARM Cortex-A53 CPU RAM 1Gb.



Gambar 2. (a) Flowchart perancangan dan (b) pengujian *database* suara referensi

2.3. Perancangan Hardware

Perancangan *hardware* pada penelitian ini ditampilkan pada Gambar 3. Pada perancangan hardware, untuk menginputkan suara digunakan mikrofon. Pemerosesan input suara dan penyimpan hasil serta *database* terletak di *Raspberry Pi* (RPI). RPI yang digunakan pada penelitian ini yaitu RPI 3 yang memiliki 40 pin I/O. Menggunakan Relay untuk menghubungkan RPI ke *home device* dengan pin output RPI.



Gambar 3. Perancangan hardware

3. Hasil dan Analisa

3.1. Hasil perancangan hardware

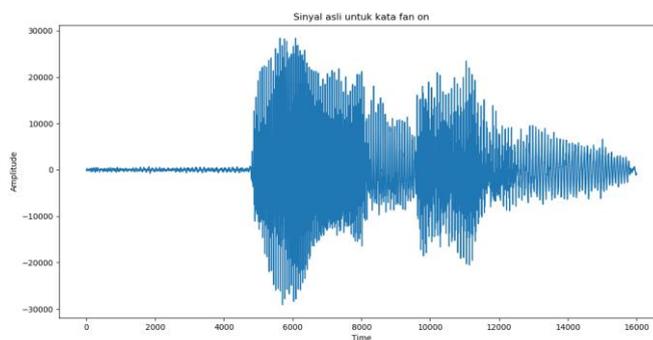
Gambar 4 merupakan gambar dari keseluruhan perangkat yang telah dihubungkan. Raspberry Pi (RPi) digunakan untuk proses *Speech Recognition* dari user dan pengontrol untuk modul relay. Untuk mengontrol modul Dimmer AC 220 Volt digunakan Arduino yang mendapatkan input perintah user dari Raspberry Pi (RPi). Raspberry Pi (RPi) dan Arduino dihubungkan melalui komunikasi serial USB.



Gambar 4 . Hardware hasil perancangan

3.2 Hasil pengambilan suara referensi

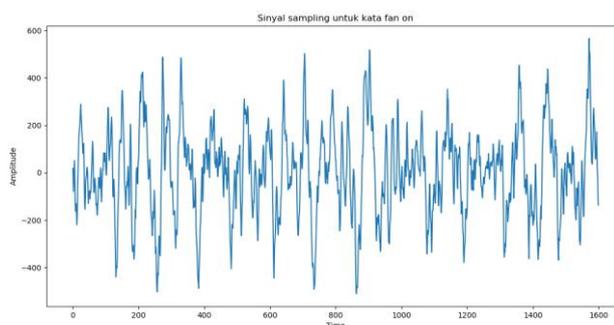
Pembuatan suara referensi bertujuan untuk mengambil sampel suara yang akan di simpan di dalam Raspberry Pi (RPi) dan akan digunakan sebagai *database*. Sampel suara yang akan digunakan sebagai suara referensi menggunakan 10 kata. Pengambilan suara referensi menggunakan mikrofon dengan 3 kali rekaman atau pengambilan untuk tiap kata.



Gambar 5. Sinyal asli untuk kata *fan on*

File suara yang direkam memiliki format Wav (*Waveform Audio*) dengan rentang frekuensi antara 300 Hz sampai 3400 Hz dan memiliki representasi sinyal analog untuk dapat

mengolah file tersebut maka file suara tersebut harus diubah ke frekuensi 16000 Hz dengan representasi sinyal digital. Alasan digunakan frekuensi 16000 Hz karena untuk melakukan adaptasi data pada CMU Sphinx. Data yang berasal dari CMU Sphinx akan digunakan sebagai database. Untuk mengubah sinyal tersebut menggunakan Pulse Code Modulation (PCM). Pulse Code Modulation (PCM) adalah proses mengubah sinyal analog menjadi sinyal digital. Proses pertama yaitu melakukan pengambilan sampel atau sampling dari sinyal suara dengan rentang frekuensi 300 Hz sampai 3400 Hz. Untuk mendapatkan informasi yang menyerupai informasi asli, penyamplingan dilakukan sesuai dengan teorema *nyquist* yaitu frekuensi sampling harus minimal dua kali lebih besar dari frekuensi asli frekuensi sampling yang digunakan yaitu 16000 Hz. Sinyal output sampler disebut sinyal PAM (Pulse Amplitude Modulation). Proses selanjutnya yaitu kuantisasi dengan memberikan level pada sinyal. Proses terakhir yaitu coding dengan mengubah sinyal PAM (Pulse Amplitude Modulation) menjadi digital di dalam blok analog to digital converter. Gambar merupakan sinyal asli dari kata *fan on*, setelah dilakukan proses perubahan sinyal dari analog to digital berubah seperti yang terdapat pada Gambar 6.



Gambar 6. Sinyal pencuplikan (sampling) untuk kata *fan on*

Proses untuk mengubah sinyal audio menjadi digital dilakukan pada aplikasi Audacity yang terdapat pada Gambar 6. Pada Audacity ini akan merubah frekuensi sampling yang awalnya dari rentang frekuensi 300 Hz sampai 3400 Hz menjadi 16000 Hz dan channel yang awalnya stereo menjadi mono. Perubahan ini dilakukan agar suara yang akan di proses menjadi *database*.



Gambar 7. Tampilan aplikasi Audacity

3.3 Pengujian Suara Dengan Noise

Pengujian suara saat keadaan ribut (noise) untuk melihat tingkat akurasi terhadap suara gangguan (noise) dilingkungan sekitar. Pengujian dilakukan saat keadaan sekitar sedang ribut atau noise. Untuk noise yang digunakan yaitu suara musik dengan *volume full* dan jarak antara mikrofon dengan sumber suara yaitu 10 cm. Percobaan dilakukan dengan 10 kali percobaan pada masing-masing kata oleh 1 *user*. Noise dengan suara musik memiliki format 'mp3' sehingga frekuensi nya yaitu 44000 Hz. Frekuensi vokal pada sebuah rekaman lagu memiliki jangkauan antara 100 Hz sampai 2000 Hz sedangkan frekuensi untuk instrument memiliki jangkauan antara 20 Hz sampai 8 KHz keatas. Untuk dapat melihat pengaruh noise terhadap pengontrolan *device* data dilihat pada Tanda X menunjukkan tidak terdeteksi, sedangkan tanda V menunjukkan suara terdeteksi.

Tabel Tanda X menunjukkan tidak terdeteksi, sedangkan tanda V menunjukkan suara terdeteksi.

Tabel 1. Percobaan dengan noise

No	Kata	Percobaan									
		1	2	3	4	5	6	7	8	9	10
1	Light On	X	X	X	X	X	V	X	X	X	X
2	Light Off	X	V	X	X	X	X	X	X	X	X
3	Light Maximum	X	X	X	V	X	X	X	X	X	X
4	Light Medium	X	V	X	X	X	X	X	X	X	X
5	Light Minimum	X	X	X	X	X	X	X	X	X	X
6	Fan on	X	X	X	V	X	X	X	X	X	X
7	Fan off	X	X	X	X	X	V	X	X	X	X
8	Maximum Speed	X	X	X	X	X	X	X	X	X	X
9	Medium Speed	V	X	X	X	X	X	X	X	X	X
10	Minimum Speed	X	X	X	X	X	X	X	X	X	X

Berdasarkan tabel 1 tingkat akurasi untuk pengontrolan *device* disaat keadaan sekitar terpengaruh noise yaitu 7% yang didapat keseluruhan dari 100 percobaan dengan keberhasilannya hanya 7. Kurangnya tingkat keberhasilan ini dikarenakan mikrofon digunakan untuk menangkap sinyal suara berdasarkan jarak mikrofon dengan *user*. Dikarenakan jarak antara mikrofon dengan noise dekat sehingga mendeteksi sinyal noise tersebut sebagai sinyal suara.

3.2 Pengujian alat

Device yang digunakan untuk pengujian alat yaitu kipas angin dan lampu. Pengujian alat pada *device* menggunakan 10 perintah. Untuk mengaktifkan (ON) dan mematikan (OFF) *device* menggunakan modul Relay yang perintahnya dikirim dari mikrokontroler Raspberry Pi (RPi) berupa angka biner (1 atau 0). Untuk mengatur intensitas cahaya pada lampu dan kecepatan pada kipas angin menggunakan modul Dimmer AC 220 Volt dengan menggunakan perintah kontrol *pulse width modulation* (PWM) dari Arduino. Perintah tersebut selanjutnya akan dikirim dan akan memodulasi lebar pulsa sehingga didapatkan intensitas cahaya sesuai dengan perintah *user*.

Tabel 1 Percobaan dengan User Perempuan

No	Kata	Percobaan (detik)										Rata-rata
		1	2	3	4	5	6	7	8	9	10	
1	Light On	6,60	5,95	5,61	5,63	5,55	5,15	5,23	5,26	5,56	4,91	5,55
2	Light Off	5,75	6,23	5,86	5,35	6,71	5,91	5,03	5,17	5,41	3,35	5,48
3	Light Maximum	5,82	5,52	9,56	11,79	10,15	5,65	8,37	6,01	3,35	7,58	7,38
4	Light Medium	5,93	6,68	8,98	10,74	10,62	5,91	10,26	7,63	6,13	5,00	7,79
5	Light Minimum	6,07	7,44	10,48	10,50	13,70	8,37	10,71	-	6,01	3,04	7,63
6	Fan on	6,86	6,77	2,70	8,99	8,63	10,26	12,52	6,44	5,97	6,18	7,53
7	Fan off	5,09	5,91	6,33	8,97	6,70	10,71	6,15	9,18	8,64	5,74	7,34

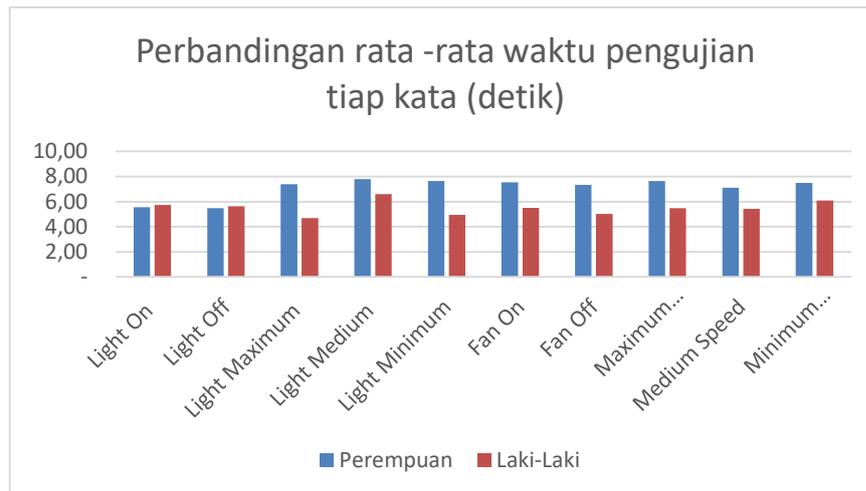
8	Maximum Speed	3,40	6,81	5,31	5,98	6,31	12,52	15,71	9,81	4,31	6,28	7,64
9	Medium Speed	5,43	6,64	-	4,56	7,63	6,15	9,99	13,66	12,00	5,01	7,11
10	Minimum Speed	5,01	6,23	5,81	7,40	6,10	15,71	4,74	14,37	5,28	4,27	7,49
											Rata-rata keseluruhan	7,09
											Waktu minimum	2,70
											Waktu Maximum	15,71

Pada pengujian alat ini menggunakan 2 *user* yaitu *user* perempuan dan laki-laki. Pada pengujian untuk yang menggunakan *user* perempuan terdapat Tabel 1. Pada pengujian ini didapat waktu rata-rata pengontrolan yaitu 7,09 detik dengan waktu tercepat 2,70 detik dan waktu pengontrolan terlama yaitu 15,71 detik.

Tabel 2. Percobaan dengan User Laki-Laki

No	Kata	Percobaan (detik)										Rata-rata
		1	2	3	4	5	6	7	8	9	10	
1	Light On	5,44	6,02	6,10	5,38	5,05	5,65	5,05	5,14	6,13	7,34	5,73
2	Light Off	5,85	5,05	6,15	5,00	5,80	5,32	5,91	4,96	5,41	6,90	5,64
3	Light Maximum	7,44	3,26	5,14	-	4,79	5,29	4,80	5,47	5,31	5,43	4,69
4	Light Medium	14,16	4,67	5,95	6,12	6,11	5,96	6,20	6,28	5,21	5,31	6,60
5	Light Minimum	9,37	-	6,37	5,20	8,68	-	5,32	4,72	5,15	4,57	4,94
6	Fan on	5,75	7,16	4,66	4,78	4,32	6,06	4,15	4,76	6,14	7,28	5,51
7	Fan off	3,86	4,57	4,31	4,82	7,36	5,29	5,24	5,47	5,23	4,16	5,03
8	Maximum Speed	8,87	6,22	5,07	5,42	6,82	5,38	-	5,13	6,30	5,47	5,47
9	Medium Speed	7,86	-	7,06	5,39	-	5,85	6,30	8,10	7,44	6,35	5,44
10	Minimum Speed	8,03	5,10	-	8,51	4,32	6,39	7,00	7,60	6,35	7,62	6,09
											Rata-rata keseluruhan	5,51
											Waktu minimum	3,26
											Waktu Maximum	14,16

Untuk pengujian *user* laki-laki terdapat pada Tabel 2. Pada pengujian ini didapat waktu rata-rata 5,51 detik dengan waktu pengontrolan tercepat yaitu 3,26 detik dan waktu terlama yaitu 14,16 detik. Waktu yang didapatkan dalam percobaan merupakan waktu dari *user* pertama kali mengucapkan perintah sampai dengan alat pengujian bergerak. Perbedaan waktu yang didapat di setiap percobaan dapat disebabkan lingkungan sekitar saat *user* memberi perintah, waktu pemroses pengenalan pola oleh Raspberry Pi (RPi) dan aksentuasi pengucapan oleh *user*. Untuk tingkat akurasi keseluruhan dari 200 percobaan yaitu 95,5 %.



Gambar 11. Perbandingan rata-rata waktu pengujian tiap kata

Perbandingan data percobaan dapat dilihat pada

Gambar . Rata-rata waktu pengujian untuk user laki laki lebih cepat dari pada user perempuan. Perbedaan waktu yang didapatkan pada pengujian dikarenakan pengaruh lingkungan sekitar saat pengambilan. Waktu pengujian untuk user perempuan lebih lama karena saat pengujian lingkungan sekitar memiliki noise suara yang lebih banyak dari ada saat waktu pengujian untuk user laki-laki. Suara noise tersebut berasal dari suara manusia.

4. Kesimpulan

Berdasarkan hasil pengujian yang sudah dilakukan didapat beberapa kesimpulan pada penelitian ini yaitu :

1. Tingkat keakurasian untuk pencocokan pola dan pengontrol *device* yaitu 95,5% dengan 200 percobaan.
2. Tingkat keakurasi pengujian suara dengan keadaan sekitar ribut (noise) yaitu 7% dari 100 percobaan.
3. Untuk pengontrolan *home device* memiliki waktu rata-rata keseluruhan, untuk user perempuan yaitu 7,09 detik dan untuk user laki-laki yaitu 5,51 detik.

Referensi

- [1] Gojali, B. A. *Implementasi Speech Recognition Untuk Kontrol Pintu*. Pekanbaru: Politeknik Caltex Riau., 2018
- [2] Muttaqin, Khairul., & Syahputra, Muhammad. *Implementasi Speech Recognition Pada Kata Kerja Dasar Menggunakan Metode MFCC*: Jurnal Elektronika dan Teknologi Komputer, Vol 2, No.1. Fakultas Teknik Universitas Samudra, 2021.
- [3] Ubaidi., & Nindian. *Penerapan Hidden Markov Model (HMM) dan Mel-Frequency Cepstral Coefficients (MFCC) pada E-Learning Bahasa Madura untuk Anak Usia Dini* :Jurnal Teknologi Inforasi dan ilmu computer, Vol 7, No.6. 2020.
- [4] Chakraborty, K., Talele, A., & Upadhy, S. . *Voice Recognition Using MFCC Algorithm*. International Journal of Innovative Research in Advanced Engineering, 1(10), 158-161. 2018.
- [5] Irfan, Farouk, dkk. *Implementasi Metode Hidden Markov Mode l(HMM) Pengenal Ucapan Serta Aplikasinya Pada Sistem Otomatisasi Pintu Gerbang Berbasis Wireless*. ALHAZEN Journal of Physics. 2018
- [6] Irham, Permana, . dkk. *Implementasi Metode MFCC dan DTW untuk Pengenalan Jenis Suara Pria Dan Wanita*. MIND Journal. 2018.

- [7] Alexander Vincent, William,.dkk. *Implementasi Speech Recognition Dalam Melakukan Automasi Pada Perangkat Elektronik Rumah Menggunakan ESP8266*. Jurnal Ilmiah Komputer. Vol 16, No.1. 2019.
- [8] Sanjaya, W. M., & Anggraeni, D. *Sistem Kontrol Robot Arm 5 DOF Berbasis Pengenalan Pola Suara Menggunakan Mel-Frequency Cepstrum Coefficients (MFCC) dan Adaptive Neuro-Fuzzy Inference System (ANFIS)*. Wahana Fisika. 2016; 1(2): 152-165.
- [9] Banamtuan, p. W., djahi, h., & maggang, a. A. *Pemanfaatan speech recognition pad smartphone android sebagai sistem pengontrolan pintu berbasis mikrokontroler*. Media elektro. 2019; 8(1): 72-78.
- [10] Nugraha, R. I., & Nugraha, A. R. *Simulasi Smart Home Berbasis Arduino*. Jurnal Manajemen Dan Teknik Informatika (JUMANTAKA). 2018; 1(1): 241-250