

# Analisa Kinerja Smart Door Berbasis Voice Recognition (Analysis Smart Door Performance Based On Voice Recognition)

1<sup>st</sup> Anggit Bagaskara  
Fakultas Teknik Elektro  
Universitas Telkom  
Bandung, Indonesia  
anggitbagaskaraa@student.telkomu  
niversity.ac.id

2<sup>nd</sup> Iman Hedi Santoso  
Fakultas Teknik Elektro  
Universitas Telkom  
Bandung, Indonesia  
imanhedis@telkomuniversity.ac.id

3<sup>rd</sup> Sevierda Raniprima  
Fakultas Teknik Elektro  
Universitas Telkom  
Bandung, Indonesia  
sevierda@telkomuniversity.ac.id

## Abstrak

Biometrik adalah autentifikasi secara biologis yang memungkinkan sistem dapat mengenali penggunaannya. Beberapa pengenalan biometrik antara lain retina, pengenalan wajah (face recognition), pengenalan sidik jari (fingerprint recognition) dan pengenalan suara (voice recognition). Voice Recognition mengalami perkembangan sehingga dapat dikombinasikan dengan Hidden Markov Model (HMM) agar fungsinya saling bersinergi satu sama lain. Harapannya dapat saling mendukung dan membantu manusia kedepannya agar dapat membantu kehidupan sehari-hari. Voice Recognition secara umum digunakan untuk proses identifikasi dan verifikasi menggunakan suara. Identifikasi suara adalah proses untuk mengenali identitas seseorang, sedangkan verifikasi suara adalah proses untuk memastikan data suara yang diberikan cocok dengan database library yang sebelumnya sudah di inisialisasi. Voice Recognition dengan konfigurasi tertentu yang dikombinasikan dengan HMM diharapkan dapat menjadi solusi untuk mendukung identifikasi dan inisiasi tersebut. Tugas Akhir ini menggunakan Raspberry Pi 4B yang dikombinasikan dengan HMM berupa Google Voice untuk memproses suara Raspberry Pi agar dapat membuka Solenoid Door Lock. Hasil yang didapatkan dari Tugas Akhir ini adalah menggunakan USB Microphone untuk mengubah suara menjadi gelombang elektromagnetik agar dapat di proses oleh Raspberry pi. Dari hasil uji coba di setiap perintah suara yang dilakukan sebanyak 10 kali dengan jarak 1 cm, 5 cm, 10 cm, 15 cm, 20 cm, 30 cm, 40 cm, 50 cm, 60 cm, dan 70 cm, dapat disimpulkan bahwa jarak ideal sumber suara terhadap microphone adalah 50 cm dengan tingkat keberhasilan sebesar 83%.

Kata kunci : Raspberry Pi, Voice Recognition, Biometrik, HMM

## Abstract

Biometrics is biological authentication that allows the system to recognize the user. Several biometric recognitions include retinal, face recognition, fingerprint recognition and voice recognition. Voice has developed so that it can be combined with the Hidden Markov Model (HMM) so that its functions synergize with each other. The hope is that they can support each other and help humans in the future so that they can

help in everyday life. Voice Recognition is generally used for the identification and verification process using voice. Voice identification is a process to identify a person's identity, while voice verification is a process to ensure that the voice data provided matches the library database that was previously initialized. Voice Recognition with certain configurations combined with HMM is expected to be a solution to support the identification and initiation. This final project uses a Raspberry Pi 4B combined with an HMM in the form of Google Voice to process the sound of the Raspberry Pi in order to open the Solenoid Door Lock. The results obtained from this Final Project are using a USB Microphone to convert sound into electromagnetic waves so that it can be processed by Raspberry pi. From the test results, each voice command was carried out 10 times with a distance of 1 cm, 5 cm, 10 cm, 15 cm, 20 cm, 30 cm, 40 cm, 50 cm, 60 cm, and 70 cm, it can be concluded that the ideal distance from the sound source to the microphone is 50 cm with a success rate of 83%.  
Keywords: Raspberry Pi, Voice Recognition, Biometrik, HMM.

## I. PENDAHULUAN

Pembobolan rumah merupakan salah satu bentuk kejahatan yang sering melanda Indonesia akhir-akhir ini. Pembobolan rumah menjadi sorotan karena banyak pencuri yang membobol rumah dengan cara membuka pintu rumah dengan kawat untuk membuka pintu agar tidak terlacak[1]. Oleh karena itu, Pintu merupakan komponen pertama dari kebutuhan keamanan pada rumah. Selain itu, Pintu merupakan akses utama untuk memasuki suatu ruangan, efisiensi fungsionalitas dan kenyamanan dalam penggunaan pintu sangat dibutuhkan untuk mempermudah manusia dalam mengakses ruangan tersebut[2]. Saat ini pada umumnya kunci pintu masih bersifat konvensional dengan kata lain membuka atau menutup pintu masih dilakukan secara manual menggunakan kunci fisik. Kunci pintu konvensional ini terkadang hilang, tertinggal bahkan patah yang berdampak pada terhambatnya pengguna untuk memasuki suatu ruangan[3].

Langkah yang bisa diambil oleh masyarakat

ialah menciptakan sebuah teknologi yang dimana kita tidak harus membutuhkan kunci untuk membuka akses pintu secara manual. Dengan demikian, kasus kejahatan tentang pembobolan rumah dapat diatasi Berdasarkan uraian permasalahan tersebut, maka dibuatlah suatu sistem kontrol penguncian pintu secara elektrik yang bersifat digital dengan perintah suara atau dikenal dengan voice recognition yang dapat digunakan untuk sistem penguncian buka tutup pintu suatu ruangan berbasis raspberry pi 4B. Kunci fisik yang bersifat konvensional diganti dengan kunci elektrik digital menggunakan solenoid magnetic doorlock.

Pada penelitian tahun 2017 yang berjudul "Implementasi Kunci Otomatis Menggunakan Face Recognition dan Pintu Otomatis menggunakan Speech Recognition Berbasis Raspberry Pi" oleh Ali Ar Ridho menjelaskan mengenai bagaimana Face Recognition dan Speech Recognition digunakan agar mempermudah orang yang memiliki disabilitas dapat memasuki suatu ruangan, namun dalam penelitian ini terdapat kekurangan seperti tidak adanya kunci konvensional berupa kunci fisik karena jika aliran listrik mati ataupun suara fisik tidak dapat dikenali karena kondisi tertentu maka prototipe ini tidak dapat digunakan untuk membuka pintu.

Pada Penelitian Tahun 2018 yang berjudul "Perancangan Smart Door Lock Menggunakan Voice Recognition Berbasis Raspberry Pi 3" oleh Diah aryani, Dedy Iskandar, dan Fitri Indriyani. Menjelaskan bagaimana Voice Recognition meminimalisir dampak akibat kehilangan kunci pintu konvensional dan meningkatkan sistem keamanan ruangan server di suatu Perguruan Tinggi di Indonesia. Namun penelitian ini memiliki kekurangan yaitu user diharuskan melakukan login pada halaman login terlebih dahulu menggunakan Smartphone yang terhubung dengan koneksi Bluetooth. Jika tidak melakukan login atau tidak mendaftar di halaman login terlebih dahulu maka seseorang yang ingin memasuki ruangan tersebut tidak dapat memasuki ruangan tersebut.

Pada penelitian-penelitian sebelumnya metode yang digunakan ialah menggunakan Raspberry pi 3 yang sudah sering digunakan di penelitian dengan tema yang sama. Pada Tugas Akhir ini metode yang digunakan adalah Raspberry Pi 4 dimana metode ini adalah pengembangan dari Raspberry Pi 3. Raspberry Pi 4 mempunyai prosesor lebih Tangguh dibandingkan Raspberry Pi 3 dan memiliki Ram yang lebih besar dibandingkan dengan Raspberry Pi 3. Selain itu Raspberry pi 4 mendukung penggunaan USB Type C.

## II. KAJIAN TEORI

Pada bagian ini, akan dijelaskan konsep dasar dan tinjauan pustaka mengenai Voice dan Speech Recognition, prinsip kerja Voice Recognition, Penggunaan

Konsep Hidden Markov Model untuk Voice Recognition.

### a. Voice Recognition

Voice Recognition adalah proses pengenalan secara otomatis suatu sinyal suara dengan membandingkan pola karakteristik dengan sinyal suara yang menjadi referensi atau acuan, yang secara umum digunakan untuk proses identifikasi dan verifikasi[3]. Identifikasi suara memiliki makna yaitu proses untuk mengidentifikasi seseorang. Dan verifikasi suara adalah proses untuk memastikan apakah data suara yang diberikan cocok dengan database library yang sebelumnya sudah di inialisasi atau ditetapkan[4]. Jika terdapat kecocokan dari apa yang diucapkan dengan salah satu perintah tadi, maka perintah bash akan dieksekusi.

### b. Prinsip Kerja Voice Recognition

Voice Command dari user atau pengguna di tangkap oleh microphone yang terhubung ke Raspberry Pi 4B, sinyal user berupa analog yang kemudian dikonversi menjadi sinyal digital menjadi sebuah bentuk teks dengan memanfaatkan google voice API. Teks tersebut lalu dibandingkan dengan perintah yang sudah ditetapkan atau diinisialisasikan sebelumnya yang kemudian jika terdapat kecocokan maka perintah akan di eksekusi dengan pemrograman python.



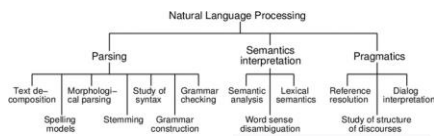
Gambar 1. Ilustrasi Voice Recognition

### c. Penggunaan Konsep Hidden Markov Model untuk Voice Recognition

Metode yang digunakan pada voice recognition yaitu menggunakan hidden markov model (HMM) karena metode ini memiliki keunggulan response time. Pendekatan ini bekerja

dengan asumsi bahwa sinyal ucapan, jika dilihat dalam skala waktu yang cukup singkat persepuluh milidetik sehingga dapat diperkirakan dan disebut proses stasioner atau proses dimana properti statistik tidak berubah seiring waktu[8]. Dalam HMM, sinyal ucapan dibagi menjadi fragmen 10 milidetik. Spektrum sinyal setiap fragment pada dasarnya adalah tinggi rendahnya sinyal dalam

satuan frekuensi yang dipetakan atau di konversikan ke dalam vektor bilangan nyata yang dikenal sebagai koefisien cepstral dan merupakan hasil akhir dari HMM. Cara kerja HMM dalam memecahkan code speech menjadi suatu teks yakni dengan mengelompokkan vektor dan mencocokkan dengan satu atau lebih fonem unit dasar ucapan. Bunyi fonem bervariasi dari satu sumber suara pembicara ke sumber lainnya. Algoritma kemudian diterapkan untuk menentukan kata demi kata yang paling memungkinkan sehingga dihasilkan urutan fonem tertentu. Disamping HMM, analisa sintaksis Natural Language Processing digunakan juga untuk memodelkan data teks dengan tahapan-tahapan



antara lain :

Gambar 2. Ilustrasi Penggunaan Konsep Hidden Markov Model untuk Voice Recognition

1. Stemming

Proses pemotongan awal atau akhir kata yang diubah ke bentuk akarnya dengan tujuan menghilangkan imbuhan seperti “o” diawal kata “open” dan “se” diakhir kata “close”.

2. Lemmatization

Proses pengurangan berbagai bentuk kata yang dirubah menjadi satu bentuk kata untuk memudahkan analisa “speak”, “speaking”, “speaks”, “spoken” adalah bentuk dari “speak”. Maka lemmatization dari kata-kata diatas adalah “speak”

3. Tokenization

Proses pembagian satu kalimat atau teks yang berkelanjutan menjadi unit-unit yang berbeda. Seperti “Open the doorplease”. Setiap katanya akan dipisah menjadi 1 unit “open”, “the”, “door”, “please”. Unit-unit ini biasa disebut dengan token

4. Morphological Parsing

Proses analisa teks menjadi komponen sintaksis logis biasanya untuk menguji kesesuaian tata bahasa. Contoh seperti kita saat kita kecil diajarkan untuk memecahkan kata “B, U, BU” “D, I, DI”. “BUDI”

d. Anaconda Platform

Anaconda merupakan platform Python/R Data Science yang populer dan banyak digunakan oleh

data scientist dan komunitas scientist lainnya Platform ini tersedia sebagai distribusi open source, yang berarti bahwa setiap menjalankan Python/R Data Science and Machine Learning pada system operasi yang umum digunakan. Seperti Linux, Windows, dan Mac OS X[9]. Anaconda merupakan Package gratis. Platform ini juga berfungsi sebagai pengelola lingkungan perangkat yang mencakup seluruh package dan library yang dibutuhkan untuk Bahasa pemograman Python atau R Development Ketika menginstall IDE (Integrated Development Environment).



Gambar 3. Anaconda Platform

e. Algoritma Klasifikasi

Pada Tugas Akhir ini akan menggunakan dependencies. Dependencies adalah perpustakaan (library) / perangkat lunak pihak ketiga atau sekumpulan file code program yang sudah dibuat orang lain untuk digunakan untuk mendukung suatu proyek penelitian dan bersifat open source yang artinya anda dapat mendownload secara gratis dari internet. Dependencies yang digunakan pada penelitian ini terbagi menjadi 2 yaitu Dependencies package di raspberry pi dan Dependencies package di Windows. Pada Tabel 1 menunjukkan perbandingan versi dependencies yang terinstall di beberapa Operating System.

Tabel 1. Perbandingan Dependencies pada beberapa Operating System

	Package	Or	O	O	Rasp
		r	r	n	be
		i	i	i	rr
		g	i	i	y
		i	r	n	Pi
		n	7	1	4
		a		0	
		l	(	(	
		(	M	x	
		M	a	6	
		a	c	4	
		c	)	6	
		)	)	4	
		)	)	)	
1	astunparse=	1.6.	1.6.3	1.6.	1.6.3
	=1.6.3	3		3	
2	audioread=	2.1.	2.1.9	2.1.	2.1.9
	=2.1.9	9		9	

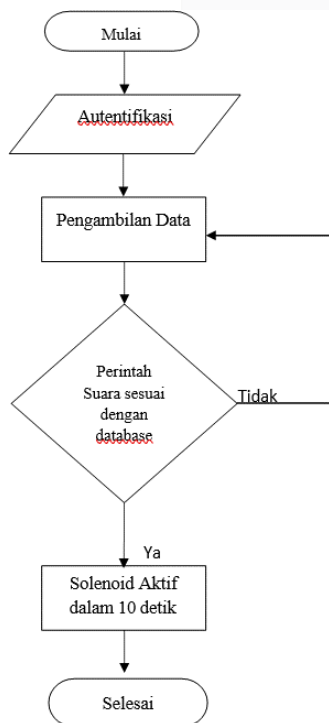
3	certifi==20.12.5	200.12.5	200.12.5	200.12.5	200.12.5
4	dataclasses==0.6	0.6	0.6	0.6	0.6
5	dill==0.3.3	0.3.3	0.3.3	0.3.3	0.3.3
6	filelock==3.0.12	3.0.12	3.0.12	3.0.12	3.0.12
7	fsspec==0.8.5	0.8.5	0.8.5	0.8.5	0.8.5
8	future==0.18.2	0.18.2	0.18.2	0.18.2	0.18.2
9	gast==0.3.3	0.3.3	0.3.3	0.4.0	0.3.3
10	joblib==1.0.0	1.0.0	1.0.0	1.0.0	1.0.0
11	Keras-Preprocessing==1.1.2	1.1.2	1.1.2	1.1.2	1.1.2
12	llvmlite==0.35.0	0.35.0	0.35.0	0.35.0	0.35.0
13	matplotlib==3.3.3	3.3.3	3.3.3	3.3.3	3.3.3
14	nlTK==3.5	3.5	3.5	3.5	3.5
15	numba==0.52.0	0.52.0	0.52.0	0.52.0	0.52.0
16	numpy==1.18.5	1.18.5	1.18.5	1.19.2	1.19.2
17	oauthlib==3.1.0	3.1.0	3.1.0	3.1.0	3.1.0
18	pandas==1.1.5	1.1.5	1.1.5	1.1.5	1.1.5
19	Pillow==8.0.1	8.0.1	8.0.1	8.0.1	8.0.1
20	protobuf==3.14.0	3.14.0	3.14.0	3.14.0	3.14.0
21	pydub==0.23.1	0.23.1	0.23.1	0.23.1	0.23.1
22	pytorch-lightning==1.1.1	1.1.1	1.1.1	1.1.1	1.1.1
23	regex==2020.11.13	2020.11.13	2020.11.13	2020.11.13	2020.11.13
24	resampy==0.2.2	0.2.2	0.2.2	0.2.2	0.2.2
25	sacremoses==0.0.43	0.0.43	0.0.43	0.0.43	0.0.43
26	scikit-learn==0.22.2.post1	0.22.2.post1	0.22.2.post1	0.22.2.post1	0.22.2.post1
27	scipy==1.4.1	1.4.1	1.4.1	1.4.1	1.4.1
28	seaborn==0.11.1	0.11.1	0.11.1	0.11.1	0.11.1
29	sentencepiece==0.1.94	0.1.94	0.1.94	0.1.94	0.1.94
30	sonopy==0.1.2	0.1.2	0.1.2	0.1.2	0.1.2
31	SoundFile==0.10.3.post1	0.10.3.post1	0.10.3.post1	0.10.3.post1	0.10.3.post1
32	tabulate==0.8.7	0.8.7	0.8.7	0.8.7	0.8.7

3	tensorboard==2.4.0	2.4.0	2.4.0	2.6.0	2.6.0
3	tensorboard-plugin-wit==1.7.0	1.7.0	1.7.0	1.7.0	1.7.0
3	tensorflow==2.3.0	2.3.0	1.14.0	2.6	2.6
3	tensorflow-estimator==2.3.0	2.3.0	2.3.0	2.3.0	2.3.0

3	tokenizers=	0.9.	0.9.4	0.9.	0.9.
7	=0.9.4	4		4	4
3	torch==1.7.	1.7.	1.9.0	1.9.	0.1.
8	1	1	/1.14	0	2.po
			.0		st2
3	torchaudio	0.5.	0.9.0	0.9.	0.9.
9	==0.5.0	0		0	0
4	torchtext==	0.6.	0.6.0	0.6.	0.6.
0	0.6.0	0		0	0
4	torchvision	0.8.	0.10.	0.10	0.2.
1	==0.8.2	2	0	.0	2.po
					st3
4	tqdm==4.5	4.54	4.54.	4.54	4.54
2	4.1	.1	1	.1	.1
4	transformer	4.1.	4.1.1	4.1.	4.1.
3	s==4.1.1	1		1	1
4	typing-	3.7.	3.7.4.	3.7.	3.7.
4	extensions=	4.3	3	4.3	4.3
	=3.7.4.3				

II. METODE

Pada Tugas Akhir dilakukan simulasi alat menggunakan aplikasi perangkat lunak yaitu Programming untuk merancang Voice Recognition pada Raspberry Pi. Gambar 4 menunjukkan Diagram Alir proses perancangan Voice Recognition hingga selesai.



Gambar 4. Diagram Alir Sistem

Tahapan dalam merancang alat pada Tugas Akhir ini yaitu merancang wiring diagram dari alat voice recognition menggunakan Fritzing. Setelah perancangan wiring diagram sudah sesuai kebutuhan maka selanjutnya dilakukan autentikasi yang bertujuan untuk memastikan alat terhubung

dengan internet. Setelah alat terhubung dengan internet, Langkah selanjutnya ialah melakukan pengambilan data berupa suara. Jika perintah suara sesuai dengan perintah yang ada pada database, maka solenoid akan aktif selama 10 detik. Jika perintah suara tidak sesuai dengan perintah maka solenoid tidak akan terbuka dan alat akan meminta untuk mengambil suara Kembali. Database tersebut terdapat pada kodingan Raspberry Pi.

a. Konfigurasi Alat dan Perintah Suara

Pada Tugas Akhir ini akan menggunakan alat berupa Raspberry pi 4B dengan operation system Raspbian dengan tegangan sekitar 5 volt 3 Ampere. Untuk perancangan Coding menggunakan bantuan aplikasi bawaan dari Raspberry Pi yaitu Programming yang menggunakan konfigurasi 3 jenis perintah suara. Adapun perintah suaranya berupa open, open the door, dan please open the door. Konfigurasi perintah suara yang telah ditetapkan sebelumnya kemudian akan diuji berada pada jarak 1 cm sampai dengan 70 cm dari microphone dengan rentang pengujian 10 kali untuk masing-masing perintah suara. Perintah suara yang diterima kemudian akan diproses Raspberry Pi. Hasil dari pemrosesan tersebut kemudian akan memicu Solenoid door lock untuk terbuka. Proses pengukuran dilakukan beberapa kali secara bertahap dan akan dilakukan beberapa pengambilan data untuk menentukan jarak ideal hasil pengujian yang telah dilakukan.

IV. HASIL DAN PEMBAHASAN

Bagian ini berisi hasil pengujian alat saat melakukan pengujian perintah suara dan pengambilan data dengan beberapa kali pengujian.

a. Tahapan Pengujian

Adapun tahapan pengujian sistem sebagai berikut:

i. Pengujian Jarak Ideal Perintah Suara Terhadap Microphone

Pengujian ini dilakukan untuk mengetahui berapa jarak ideal dari sumber suara dengan microphone untuk dapat menerima perintah dengan baik. Pengujian ini dilakukan 10x pada setiap interval. Sehingga dapat mengetahui jarak keberhasilan jarak dan range jarak ideal untuk memberi perintah suara.

ii. Pengujian Sumber Perintah Pengujian ini dilakukan untuk mengetahui apakah perintah yang diucapkan cocok dan diterima sehingga dapat membuka pintu atau tidak.

iii. Pengujian Delay Pemberian Perintah dengan Eksekusi Solenoid

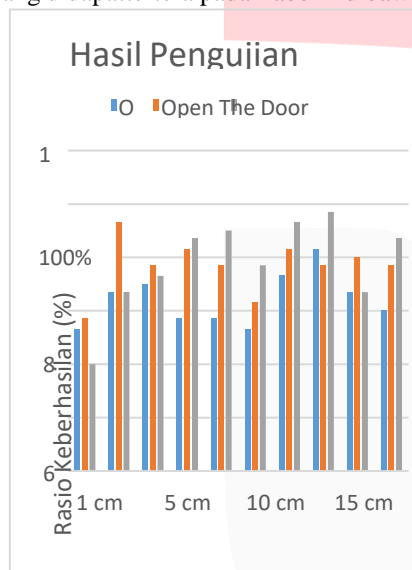
Pengujian ini dilakukan untuk mengetahui berapa lama respon dari *Prototype* perancangan. Memproses data hingga eksekusi *solenoid magnet* buka tutup pintu bekerja dengan memperhatikan kondisi lingkungan, sensitivitas *microphone*, kecepatan koneksi internet dan *natural language processing*.

iv. Pengujian Tingkat Kebisingan pada Microphone

Pengujian ini dilakukan untuk mengetahui tingkat kebisingan ketika melakukan pengujian.

#### b. Data Hasil Pengujian Sistem

Pengujian ini menghasilkan nilai Jarak Ideal dari masing-masing perintah suara mendapatkan hasil yang didapattertera pada Tabel 2 dibawah



Gambar 5. Grafik Hasil Pengujian Jarak Ideal Seluruh Perintah Suara

Berdasarkan Gambar 5 Dari hasil uji coba dari grafik diatas untuk menentukan jarak ideal perintah suara terhadap microphone minim noise dengan perintah “open”, “Open The Door”, dan “Please Open The door” dengan 30 kali uji coba maka diperoleh presentase jarak ideal untuk ketiga variasi pengucapan adalah 50cm dengan presentase nilai rata-rata 86%. Range tingkat keberhasilan sumber suara terhadap microphone berjarak antara 15cm hingga 50cm dengan presentase 75% sampai 86%. Adapun tabel delay pemberian perintah dengan eksekusi selenoid pada Tabel 2.

Berdasarkan Tabel 3 menjelaskan bahwa seluruh perintah suara yang telah ditetapkan sebelumnya yang berupa peritah open, open the door, dan please open the door semuanya memiliki kecenderungan untuk mendapatkan nilai kebisingan

dias 60 dB yang disebabkan oleh factor dari suara knalpot motor, suara percakapan orang dalam rumah serta suara dari beberapa orang yang lewat ruang uji coba. Sebab jika dalam keadaan sunyi tingkat kebisingan sebesar 47,7 dB yang hanya diakibatkan oleh suara yang berasal dari exhaust fan pada Raspberry Pi.

#### V. KESIMPULAN

Sistem *Smartdoor lock* berbasis *Voice Recognition* telah berhasil diimplementasikan pada Tugas Akhir ini dengan menggunakan *Raspberry Pi 4B* dengan akurasi sebesar 83% untuk seluruh jenis perintah suara.

#### REFERENSI

- [1] Herdianto, “Perancangan Smart Home dengan Konsep Internet of Things ( IoT ) Berbasis Smartphone,” *Ilm. Core It*, vol. 6, no. x, pp. 120–130, 2018.
- [2] H. Z. Matondang, D. W. Sudiharto, A. Gautama, and P. Satwiko, “Penggunaan Face Recognition & Voice Recognition sebagai Dua Langkah Verifikasi dan Peningkatan Keamanan pada Smart Door.”
- [3] D. Aryani, D. Iskandar, and F. Indriyani, “Perancangan Smart Door Lock Menggunakan Voice Recognition Berbasis Rapberry Pi 3,” *J. CERITA*, vol. 4, no. 2, pp. 180–189, 2018, doi: 10.33050/cerita.v4i2.641.
- [4] K. Anam, “PENGENALAN SUARA MANUSIA MENGGUNAKAN METODE LINIER PREDICTIVE CODING (LPC).”
- [5] D. Arbaus, D. Arman Prasetya, and A. Puspita Sari, “KECERDASAN BUATAN PADA SISTEM PINTU OTOMATIS MENGGUNAKAN VOICE RECOGNITION BERBASIS RASPBERRY PI.”
- [6] “Perbandingan Spesifikasi Raspberry Pi 4/3B+/3B/2B/1B+/1A+,” 2019. [https://digiwarestore.com/id/digiware-news/50\\_perbandingan-spesifikasi-raspberry-pi](https://digiwarestore.com/id/digiware-news/50_perbandingan-spesifikasi-raspberry-pi).
- [7] RaspberryPi, “Raspberry Pi 4,” *Raspberry Pi Foundation*. 2020.
- [8] “Hidden Markov Model,” pp. 177–194, 2010, doi: 10.1142/9789814287319\_0012.

[9] Jason Brownlee, "Python Environment for Machine Learning with Anaconda," 2017. <https://machinelearningmastery.com/setup-python-environment-machine-learning-deep-learning-anaconda>



