Speaker Recognition for Device Controlling using MFCC and GMM Algorithm

1st Ridwan Abdul Malik
School of Electrical Engineering
Telkom University, Indonesia
ridwanamdt@student.telkomuniversity.ac.id

2nd Casi Setianingsih.
School of Electrical Engineering
Telkom University, Indonesia
setiacasi@telkomuniversity.ac.id

3rd Muhammad Nasrun
School of Electrical Engineering
Telkom University, Indonesia
nasrun@telkomuniversity.ac.id

Abstract*—Biometric technology is widely used to identify a smart home device controller with access control to the system. Sound Abstract is one of the biometric technologies used because human speech is different and unique. Generally, a smart home device controller based on sound can be controlled by everyone so that a speaker who should not have access rights to the system will still execute his voice command. The solution to this problem is a sound control system that can identify one speaker's voice with other speakers registered on the system to control smart home devices and reject commands from foreign speakers who are not registered on the system to secure a voice control system is formed. The Mel-Frequency Cepstrum Coefficient (MFCC) method, capable of capturing the characteristics of different human voices and is unique; the output of the MFCC is modeled and classified using GMM (Gaussian Mixture Model) on each cepstrum subject so that the modeling results can identify the voice of the speaker registered on the system listed or the voice of foreign speakers not registered with the system. The accuracy of the system built can identify the voice of the speaker registered on the system by 98.1% and reject the voice of the speaker who is not registered on the system by 91.6%.

Keywords: Speaker Recognition, MFCC, GMM, Raspberry Pi.

I. INTRODUCTION

The smart home is an intelligent system that can control electronic devices at home, the development of intelligent home is increasingly rapid with the existence of the Internet of Things (IoT) that can connect all electronic devices to the internet network.

Biometric technology is widely used to identify a smart home device controller with access control to the system. Sound is one of the biometric technologies that can be used because human speech is different and unique. Generally, a smart home device controller based on sound can be controlled by everyone so that a speaker who should not have access rights to the system will still execute his voice command. The solution to this problem is how to make a voice control system that can identify the voice of one speaker with other speakers registered on the system to control smart home devices and reject commands from foreign speakers who are not registered on the system so that a secure voice control system is formed. The Mel-Frequency Cepstrum Coefficient (MFCC) method,

can capture the characteristics of different and unique human voices [1], so that special characteristics can be taken as distinguishing speaker sounds, the output of the MFCC is a feature vector called cepstrum, then modeling using Gaussian Mixture Model (GMM) on each cepstrum subject so that the modeling results can be classified and compared with the voice of one speaker with other speakers who have access control.

In this research, a secure speaker recognition system is designed to identify one speaker's voice with another speaker who has access rights to the system and rejects the foreign voice

commands that do not have access rights to the system to control electronic devices connected to the smart home system.

II. RELATED WORK

A. User Identification System Using Biometrics Speaker Recognition by MFCC and DTW along with signal processing package (Tazwar Muttaqi; S. Hossein Mousavinezhad)

The purpose of this study is to identify users by MFCC and DTW techniques. In the paper, verify a user name initially to find a sound recording that matches the user's name. Each voice recording has a length of three seconds of speech. So, at the beginning of the verification process, a user must enter his name correctly, and then the user will be asked to speak into the microphone. A user must speak the exact sentence or words he utters when the database sounds recorded. Voice signals are recorded last by the user, and voice recordings in the database are sent for digital processing. After digital processing, Mel-Scale Frequency Cepstral Coefficient (MFCC) extracts unique features from the sound signal. The Mel scale connects the sound frequency or tone to the actual measurement frequency.

B. Speech and Speaker Recognition for HomeAutomation: Preliminary Results (Michel Vacher; Benjamin Lecouteux; Javier Serrano Romero; Moez Ajili, Franyois Portet; Solange Rossato)

This paper presents an approach to provide voice commands in multi-room smart homes for seniors and people with visual impairments. The purpose of this paper is to recognize home automation commands and to identify speakers. Speaker identification provides supplementary information for context inference (location, activity, the speaker's identity who utters a command). The voice training system uses large data sets modeled using the Universal Background Model (UBM) algorithm. UBM training uses an iterative EM algorithm. The Expectation-Maximization (EM) algorithm is divided into two steps:

- Expectation Evaluation (E), where the expected log function value is likely calculated concerning the observed values.
- The maximization step (M), where the parameter maximizes the maximum likelihood expected.

978-1-7281-9877-4/20/$31.00 ©2020 IEEE

The parameter estimated in the Maximization Step (M) is used as a checkpoint to start a new Expectation Evaluation (E).

III. Research Method

A. Speaker Recognition

Speaker recognition is the process of identifying a person's speaker voice through his voice characteristics [2]. Naturally, the human brain can do speaker recognition to distinguish someone's different voice without seeing his face. Speaker Recognition can be applied to software to identify human voices [3]. In general, Speaker Recognition has several stages: feature extraction, modeling, and classification [4].

Speaker recognition can be divided into text-dependent and text-independent. In the dependent text, the word spoken by someone has already been specified [5]. Whereas in the independent text, there is no predetermined word, so the system must model the general feature of a speaker's voice [5][6].

B. Mel Frequency Cepstrum Coefficient (MFCC)

MFCC is a method of extracting features whose output is a feature vector called cepstrum [3]. MFCC performs calculations ranging from wavelength, noise, and other things so that a sound's characteristics can be adequately extracted. The process stage in the MFCC is preprocessing, framing, Mel filter bank, FFT, windowing, cepstrum [7].

The first step taken in the speaker recognition process is preprocessing, normalizing, or leveling the magnitude of the sound signal; the next process is framing by changing the analog signal unit time t to become a function of x(n). Then the framing process is carried out by dividing the signal into several frames to facilitate analysis, to reduce noise at both ends of the frame, the windowing process is carried out. To make the process easier, a Fast Fourier Transform (FFT) is done which is to change the time domain signal to the frequency domain; after changing the signal domain, a Mel Filter Bank calculation is performed, which is to connect the perceived frequency with the actual frequency measured [8], the process of converting frequency to the mel scale, the function of the mel scale to map the linear frequency scale of the speech signal to a logarithmic scale for frequencies higher than 1 kHz [9]. The final process is to change the signal from the mel scale to the time domain and get a feature extraction called cepstrum.

C. Gaussian Mixture Model (GMM)[3]

GMM is an algorithm that functions to model some data into a gaussian distribution using the mean parameter (μ) and variance (σ2) from a data distribution. Mean (μ) is a midpoint of a gaussian distribution, and variance (σ2) is a measure of the data's distribution value. GMM has the advantage of modeling more than one Gaussian distribution. Fig. 2 is a plotting model of a data set using GMM.

The first step that must be done is to model data used into a probabilistic function using equation (1).

$$p(x|\theta) = \sum_z p(x, z|\theta) \quad (1)$$

X is a data, θ is parameter mean (μ) and variance (σ2) some data, and z is a certain gaussian member. After probabilistic modeling, the next steps are as follows:

1) Lower Bound

The lower bound is a function to lookup values $p(x|\theta)$ less than or equivalent to $p(x|\theta)$. The lower bound function uses an equation Jensen's inequality to $\log p(x|\theta)$, like equation (2).

$$\log \sum_n \lambda_n x_n \geq \sum_n \lambda_n \log x_n \quad (2)$$

After the lower bound process is carried out, the parameter maximization process is carried out using the EM algorithm.

2) Expectation-Maximization (EM)[10]

EM is an algorithm for calculating Maximum Likelihood (ML) estimates that appear in hidden data iteratively [9]. EM algorithm has two processes, namely E-step and M-step, which is done iteratively. The E-step process, the current model estimation data, is used to estimate hidden data. In the M-step, the hidden data estimation results were previously used to maximize the low-bound function. The results of the E-step and M-step processes produce the model with the greatest likelihood of the entire data being observed. Equation (3) is used to find the right parameters to calculate EM by considering the lower bound equation.

$$Q(\theta, \theta^{old}) = E_{z\sim P(z|x,\theta)^{[\log p(x,z|\theta)]}} = E_{zP(z|x,\theta^{old})[\log p(x,z|\theta)]} \quad (3)$$

IV. System Design and Overview

A. Overview of The System

The built system outline can identify the speaker's voice who has access and not, to control the smart home device. In Fig. 2, the black line is a system built in the research. The workings of the system built-in Fig. 2, a user inputs by saying a command to the microphone, then the microphone receives an input in the form of a speaker's voice to be trained or predicted in his voice, to be able to access control the smart home system. Raspberry Pi acts as an incoming sound processor and processes the sound using a predetermined algorithm, to do a match between the sound that has access control and not, to the smart home device control system, if the system detects a sound match with the speaker who has access, then the system will allow access to control electronic devices connected to a microcontroller such as NodeMCU, so that the microcontroller as a switch can turn on a smart home electronic device that is connected to the microcontroller, based on the sound of the speaker who has access.

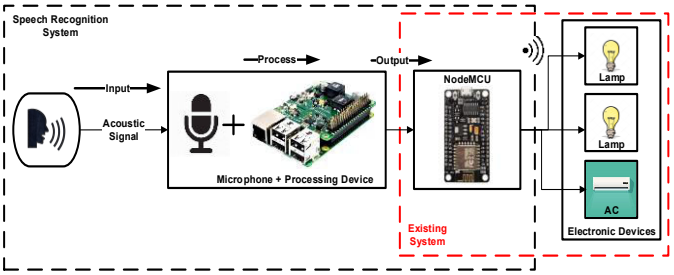

Fig 2. Overview of the General System Built

B. System Planning

At this stage, the authors design the system that has been built. In Fig. 3, the first receives input in the form of sound processed with a predetermined algorithm. To differentiate the

978-1-7281-9877-4/20/$31.00 ©2020 IEEE

input of training sounds from predicted sounds, an option is provided on the menu, which functions as a menu differentiator and the user interface. In the voice training process, the speaker will be trained and extracted using MFCC characteristics, namely Preprocessing, Framing, Windowing, FFT, Mel Filter Bank, so that the vector feature is called cepstrum, then some cepstrum data sets will be modeled using GMM, so we get some mean and variance set of voice data of speakers trained, to be stored as sounds that have access to control the smart home system. In voice prediction, it will be extracted using the MFCC algorithm and modeled using GMM as the training process. The results of modeling the predictive sound and training sound will be compared; if there is a match between the predictive sound and training sound, then the speaker's voice is given access to control the device connected to the microcontroller and vice versa if you do not have access then the system will reject the voice command given by the speaker.

1) Voice Training Menu and Voice Prediction

The "Voice Training" menu is used to record and train the speaker's voice input, and the "Voice Prediction" menu is used to identify the user's voice match with the database of the voice training model that has been created.

2) Training Speaker Using Voice Recording Process

Following the steps in the speaker training process:

- On the speaker recognition system, select the voice training menu.
- Enter the speaker's name to be trained and a password to enter the voice training menu. The password is a security measure that is installed on the system. If the password is correct, the login process is successful.
- Record the speaker's voice according to the voice command in Table III, each command record three times. The recording is a wav format file.
- Do the feature extraction process using the MFCC algorithm. The result of this process is cepstrum features in the form of an array with 13 coefficients.
- Do the modeling process and use the GMM algorithm on the feature cepstrum.
- The speaker sound model produced in this training process will be used in the sound streaming process to detect the speaker's voice who provides voice command input.

3) Sound Streaming Process

On the "Sound Prediction" menu, a continuous sound stream is performed to detect the speaker's command. Following the steps in the sound streaming process:

- The speaker recognition system listens to the input sound continuously.
- If an input sound is detected with sound intensity> = 35 dB, the system will record the input sound until the sound intensity returns below 35 dB. The resulting record is saved in a wav format file.
- Calculate the duration of the recording; if the duration is 1.3 seconds to 2 seconds, it will proceed to the next process. If the recording duration is under 1.3 seconds or above 2 seconds, the process returns to listening to the input sound.
- The next process is feature extraction on the voice of the speaker input using the MFCC algorithm. The feature extraction results will be used to recognize the speaker's voice from the existing sound model.
- In the speaker recognition process, if the speaker's voice is detected already in the training data, the voice command will be forwarded to the speech recognition process. Meanwhile, if the speaker's voice is not detected in the training data, the system will refuse the next process and return to listening to the input sound.

4) Preprocessing [5]

This process uses preemphasis filtering to maintain high frequencies in a spectrum, which is generally eliminated during sound production and used to reduce the noise ratio in the signal to improve the signal quality. A sample of preprocessing results when saying command 1 to turning on device one is shown in Fig 4.

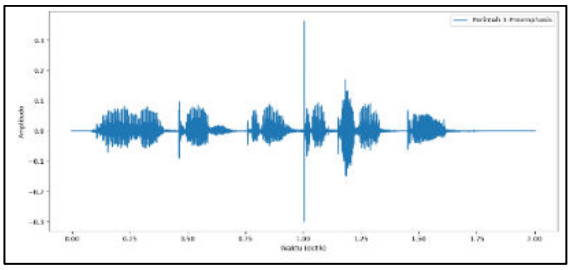

Fig 4. Preemphasis when Saying Command 1

5) Frame Blocking[5]

At this stage, a short segment process is performed on the input sound signal. The frame length of 10-30 milliseconds is often used for signal processing. In this study, the signal is cut into frames with a value of 25 ms, with the overlap size of each slot is half of the length value of one frame. In Fig. 5, the speaker utters the command word 1, carried out framing and windowing.

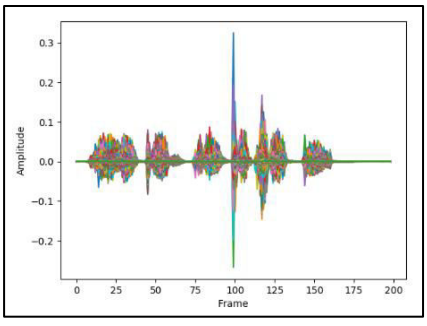

Fig 5. Framing and Applying Hamming Window

6) Windowing[5][11]

To reduce leakage due to the framing process that causes aliasing, a new signal has a different frequency from the original. So the Window process is carried out because a good Window function must narrow in the main lobe and widen in the sidelobe [10]. The

978-1-7281-9877-4/20/$31.00 ©2020 IEEE

following is the formula used in the Window function for the input signal.

$$x(n) = x_i(n) \times w(n) \quad (4)$$

$$w(n) = 0.54 - 0.46 \cos\left(\frac{2\pi n}{N-1}\right) \quad (5)$$

Where, n = 0, 1, 2, ..., N

x(n) = signal sample value

xi(n) = sample value of the i-th frame of the signal

w(n) = window function in the hamming window.

7) FFT[5]

At this stage, the FFT process is carried out to facilitate the next process calculation (Mel-filterbank); then, the signal is converted into the frequency domain. Modifications made are by grouping the odd n and even n boundaries so that the N point DFT becomes (N / 2) point [10].

$$D_k = \sum_{m=0}^{N_{m-1}} D_m \times e^{\frac{-j2\pi km}{N_m}} \quad (6)$$

Where,

k = 0, 1, 2, 3, …, N_{m-1}

Xi(k) = signal sample value in the frequency domain

xi(n) = signal sample value in the time domain

w(n) = windowing function N = number of frames

K = FFT size

In the FFT process, the K value used is 2048 because in the sound recording stream process, to do the FFT process with a sound duration of 1.3 to 2 seconds, a K value of more than 1024 is required.

8) Mel-Frequency Wrapping[5][12][13][14]

This stage is carried out filtering on the frame frequency obtained at the previous stage, using several M filter banks. This filter follows the human ear's anatomy, where the 'Mel' scale (derived from Melody) has no linear relationship with the frequency of the sound. This filter's results are applied to the triangle filter function as a multiplier factor in the input signal frames in the FFT process.

$$\text{mel}(f) = 2595 \times \ln\left(1 + \frac{f}{700}\right) \quad (7)$$

Where,

mel (f) = mel frequency scale

f = linear frequency.

9) Cepstrum[5][14]

This stage is the final stage of the MFCC algorithm for feature extraction. The Mel-frequency scale will be converted from the mel spectrum to the time domain using the Discrete Cosine Transform (DCT) formula.

$$C_n = \sum_{k=1}^{K} (\log S_k) cos\left[n\left(k - \frac{1}{2}\right)\frac{\pi}{K}\right] \quad (8)$$

Where,

n = 1, 2, 3, …, K

Cn = cepstrum mel frequency coefficient

Sk = the mel power coefficient.

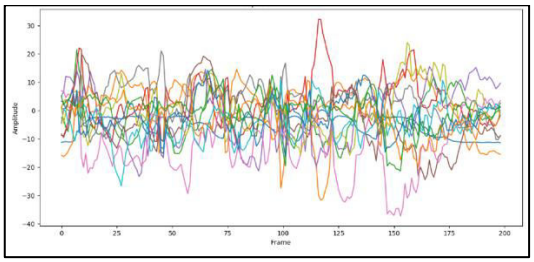

Fig 6. Cepstrum in Spectrum

```
   6.50328857e-02   8.72959951e-01   3.87341988e-01  -5.30685536e-01
  -2.97278196e-01]
[ -1.72707421e+00  -1.54031985e+00   2.20016365e-01  -9.91131021e-01
   9.83201222e-01  -5.31244268e-01   1.22764289e+00  -1.88782162e+00
  -5.42002094e-02   1.14301197e+00   3.52758649e-01  -6.29110116e-01
   1.46438411e-01]
[ -1.70703695e+00  -1.58071413e+00   3.06902835e-01  -7.80428696e-01
   1.06278137e+00  -6.59284288e-01   1.36918010e+00  -1.91185110e+00
  -7.83855116e-02   1.05820970e+00   4.72298332e-01  -3.23945683e-01
   1.81237097e-01]
[ -1.75647200e+00  -1.60767066e+00   4.88493751e-01  -5.73720645e-01
   1.17646663e+00  -6.17335883e-01   1.47463978e+00  -1.67302572e+00
  -1.37142584e-01   4.54635978e-01   5.05385759e-01   3.16751805e-01
   5.21013063e-01]]

('Number of Frame : ', 199)
```

Fig 7. Cepstrum in Array 13 Form of Coefficient with 199 Many Frames

10) Gaussian Mixture Model (GMM)

The characteristic extract results will be compared to the modeling between the training sound sample with the sound data test. Fig. 8 shown the results of modeling using GMM; there are similarities in the mean and variance, the test sound model with the model that has been trained.

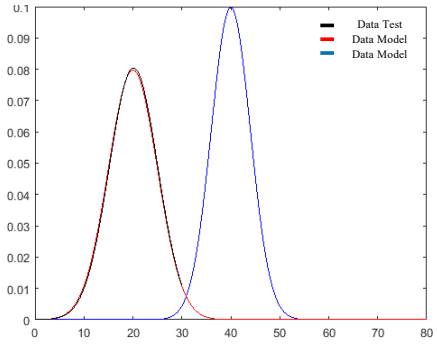

Fig 8. Modeling results using GMM

GMM configuration uses a library of sklearn with threshold parameters set according to the trials and analyses conducted.

11) Identification and Validation

The identification stage is compared between the log-likelihood value obtained from the estimated parameter mean and the variance between the predicted speaker voice data and the voice data that has been done in previous training.

978-1-7281-9877-4/20/$31.00 ©2020 IEEE

C. Hardware Design

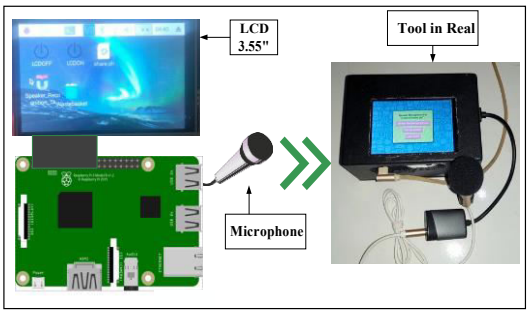

Fig 9. Hardware Design

Hardware design is a form and arrangement of several parts of the device, assembled into a whole machine to carry out the speaker recognition process. Fig. hardware abstracts made like Fig. 9, the Fig. has used three pieces of equipment, including LCD that uses a portion of the Raspberry Pi GPIO pin to communicate and the power supply of the Raspberry Pi, then the microphone is connected via the USB port as a voice signal receiver in the speaker recognition.

V. Testing Methode

A. Data Training

The number of people used as training data amounts to 6 speakers, for each speaker will say six words with one word trained three times. The total sample of voice data in training was 108 data sets.

The number of voice data of unregistered speakers is 24 voice data sets, consisting of 4 different speakers saying the same words as the speaker in training. In Table I, the command is used on the training sound sample spoken by the speaker. Command words were spoken in Indonesian.

TABLE I. SPEAKER COMMANDS

Comannd	Pronounced Words	Pronounced Words In English
1	Nyalakan perangkat satu	Turn on device one
2	Nyalakan Perangkat dua	Turn on device two
3	Nyalakan perangkat tiga	Turn on device three
4	Matikan perangkat satu	Turn off device one
5	Matikan perangkat dua	Turn off device two
6	Matikan perangkat tiga	Turn off device three

B. Speaker Detection Analysis

TABLE II. LOG LIKELIHOOD VALUE OF THE SPEAKER DETECTION

No	Speaker	Log Likelihod Value	Speaker Sound Test
1	1	3958	Speaker 2
	2	**-2786**	
	3	-3316	
	4	-3624	
	5	-3765	
	6	-3789	

Table II is one example of testing the log-likelihood value of 10 different speakers consisting of 6 speakers registered on the system, and four speakers not registered on the system. GMM output values when identifying sounds tested with training sound samples. The "Log-Likelihood" value results from the sum of the similarity values between the votes tested and the training sound sample data. From the trials that match some training sound data samples with different speaker sound data, the log-likelihood value results vary; the more the value approaches the positive number, the greater the similarity of the test sound with the training sound sample. From the data in Table IV, log-likelihood values below -3000 are the values generated if the voice data tested with training sound sample data is not the same speaker, while values above -3000 are generated from test sounds with the same training speaker sound sample data. The log-likelihood value above -3000 can be used as a threshold value to detect speaker sound that is the same as the sound registered or already in training about the system being built.

C. Testing Identification of Voice Samples in Training Data with Test Data

TABLE III. VOICE MATCH VALIDATION

Speaker	(Word) Pronounced Command						Total Detected
	1	2	3	4	5	6	
1	1	1	0	1	1	1	5
2	1	0	1	1	1	1	5
3	1	1	1	1	1	1	6
4	1	1	1	1	1	1	6
5	1	1	1	1	1	1	6
6	1	1	1	1	1	1	6

Information :

1: Match test data with training data

0: Mismatch of test data with training data

In Fig. 11, the speaker's best identification is on speakers 3-6 as many as six of the six different words. Based on the identification results above, the graph of the accuracy is as follows:

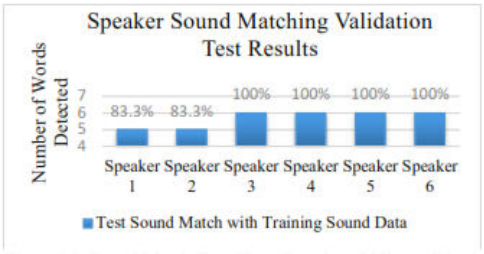

Fig 11. Sound Match Test Chart Based on Different Words

Based on experiments using training data for 108 speaker voice datasets consisting of six different speakers with six other words and each word in training three times, the system built managed to identify speaker sounds for different people by 98.1%.

D. Rejection Testing on Test Sound Data Not Training

In Table IV, an identification test of speaker sound samples not registered in the training data is tested.

TABLE IV. LOG LIKELIHOOD VALUE

Speaker	(Word) Pronounced Command						Total Detected
	1	2	3	4	5	6	
7	0	1	1	1	1	1	5
8	1	1	0	1	1	1	5
9	1	1	1	1	1	1	6
10	1	1	1	1	1	1	6

Information :

978-1-7281-9877-4/20/$31.00 ©2020 IEEE

1: The sound is not recognized

0: Voice recognized

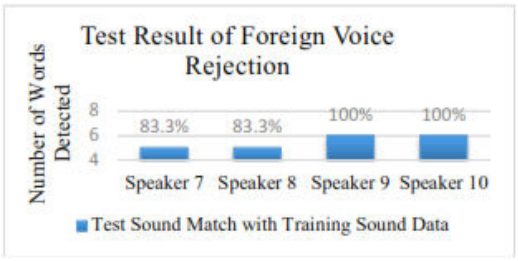

Fig 12. Rejection of a Foreign Voice Not Training

Based on the graph in Fig. 12, speakers 7 and 8 are correctly turned down in several five of six different words, while speakers 9 and 10 can be turned down in several six out of six other words.

Based on experiments using training data, 24 speaker sound datasets consisting of 4 speakers with 6 different words, the system that produces sound on unregistered speakers was 91.6%.

E. Testing the Identification of Training Data with Test Data Based on Sound Intensity

This experiment tests the accuracy of matching training data with test data based on sound intensity. The test sound used is the word "Turn on Device One" in each test data, and the sound intensity is calculated for each word in decibels. The test results can be seen in the graph below.

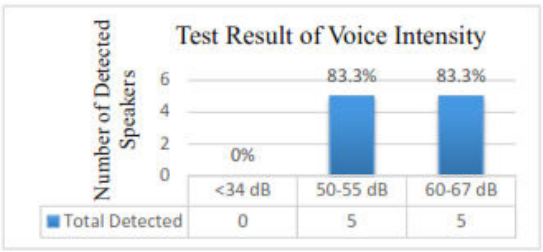

Fig 13. Test Result Graph Based on Sound Intensity

In Fig. 13, the test sound intensity is below 34 dB, so the sound cannot be received by a system that produces small sounds such as noise waves, while the sound with an intensity of 50-55 dB and 60-67 dB Requires 5 speakers from 6 different speakers, the results of detection two speakers issued the wrong person because there are similarities between the two sounds.

F. Testing Response Time Matching Voice Samples in Training Data with Test Data

In Fig. 14, speaker 1 to speaker 6 generated response time is less than 0.2 seconds from 6 times the test for different speakers with the amount of training data for 108 data sets, so that the processing time is relatively fast. The response time results will be different depending on the amount of training data.

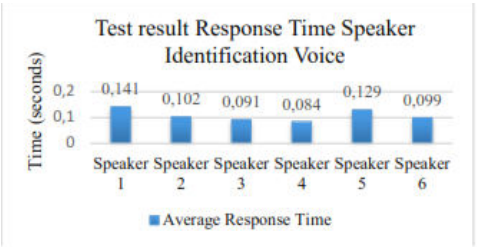

Fig 14. Graph of Average Word Detection Response Time for Each Test

VI. Conclusion

From the results of implementation and testing can be concluded that the speaker recognition system can identify the speaker sounds that have been registered or in previous training with an accuracy of 98.1%, using training data of 108 sound datasets, consisting of 6 speakers with 6 different voice commands performed 3 times training on each command the sound.

References

[1] N. P. Trilok, S. Cha, and C. C. Tappert, "Establishing the Uniqueness of the Human Voice for Security Applications," *Dimens. Contemp. Ger. Arts Lett.*, pp. 6–11, 2004.

[2] R. D. Peacocke and D. H. Graf, "An Introduction to Speech and Speaker Recognition," *Computer (Long. Beach. Calif).*, vol. 23, no. 8, pp. 26–33, 1990.

[3] D. K. Putra, I. Iwut, and R. D. Atmaja, "Simulasi Dan Analisis Speaker Recognition Menggunakan Metode Mel Frequency Cepstrum Coefficient (mfcc) Dan Gaussian Mixture Model (gmm)," *eProceedings Eng.*, vol. 4, no. 2, 2017.

[4] T. Mahboob, M. Khanum, M. Sikandar, H. Khiyal, and R. Bibi, "Speaker Identification Using GMM with MFCC," *IJCSI Int. J. Comput. Sci. Issues ISSN ISSN*, vol. 12, no. 2, pp. 1694–814, 2015.

[5] Y. N. Utami, N. Anbaranti, F. Teknik, and U. Telkom, "Perancangan Speaker Recognition Pada Sistem Kendali Lampu Berbasis Mikrokontroler Speaker Recognition Design Based on Lamp Control System," in *Proceedings of Engineering (E-PROCEEDING)*, 2015, vol. 2, no. 2, pp. 3332–3346.

[6] B. Plannerer, *An Introduction to Speech Recognition*. 2005.

[7] B. J. Mohan, R. B. N, and A. R. Module, "Speech Recognition using MFCC and DTW."

[8] B. J. Mohan and N. Ramesh Babu, "Speech recognition using MFCC and DTW," *2014 Int. Conf. Adv. Electr. Eng. ICAEE 2014*, 2014.

[9] M. Vacher, B. Lecouteux, J. S. Romero, M. Ajili, F. Portet, and S. Rossato, "Speech and Speaker Recognition for Home Automation : Preliminary Results," 2015.

[10] S. Borman, "The Expectation Maximization Algorithm A short tutorial," *Tutor. from emtut@seanborman .com*, vol. 25, no. x, pp. 1– 9, 2009.

[11] D. B. Manurung, B. Dirgantoro, and C. Setianingsih, "Speaker Recognition For Digital Forensic Audio Analysis Using Learning Vector Quantization Method," in *2018 IEEE International Conference on Internet of Things and Intelligence System (IOTAIS)*, 2018, pp. 221–226.

[12] A. Awad, H. Omar, Y. Ahmed, and Y. Farghaly, "Speech Recognition System Using MFCC and DTW," *Speech Recognit. Syst. Using MFCC DTW*, no. December 2016, p. 4, 2016.

[13] C. Ittichaichareon, "Speech recognition using MFCC," *Int. Conf. Comput. Graph. Simul. Model.*, no. September, pp. 135–138, 2012.

[14] H. Z. Muhammad, M. Nasrun, C. Setianingsih, and M. A. Murti, "Speech Recognition for English to Indonesian Translator Using Hidden Markov Model," in *International Conference on Signals and Systems*, 2018, pp. 255–260.

978-1-7281-9877-4/20/$31.00 ©2020 IEEE