Home Monitoring and Control Using Smartphone and Speech Processing

Mochamad Mobed Bachtiar[1,*] Bima Sena Bayu Dewantara[1] Dwi Prastyo [1]

[1] *Department of Informatic and Computer Engineering, Politeknik Elektronika Negeri Surabaya, Indonesia*
[*]*Corresponding author. Email: mobed@pens.ac.id*

ABSTRACT

Monitoring and control of home electronic equipment in general is still done manually, this is less efficient if we are not at home but want to monitor the condition of the electronic equipment. With the remote control using a smartphone can be a solution to simplify the control system and monitoring of electronic equipment. In this study, we propose to use voice input to control electronic equipment based on speech commands in a smartphone application. The method used in recognizing sound uses Dynamic Time Wrapping. After several tests, the results of the detection of speech commands in the form of "letters / characters" have a high success rate of 100%, while the "word" command tests have a success rate of 87% and the results of tests of commands in the form of pronouncing "sentences" have a success rate of 57 %. So that from the system that has been made, the more sentences are spoken, the accuracy is low.

Keywords: *Home Monitoring, Smartphone Control, Speech Processing, Dynamic Time Warping*

1. INTRODUCTION

In this modern era, needs that are fast-paced and independent of distance make it easier for humans to do many things. An automatic home can be a practical solution to meet those needs. A smart home is a home where every piece of equipment can be controlled and monitored remotely, without having to press a switch directly. Smart homes are becoming an increasingly popular issue today. The advantage of this system is to help home owners control the electronic equipment in the house. The purpose of making this technology is to make it easier for humans to control and monitor electronic equipment remotely.

Android technology and the Internet of Things were introduced to remotely control home appliances [1] to monitor security conditions and issues. This technology is very helpful for someone to control home and control security using only the android application, but in these applications still need touch a button in the application to control the commands if the user wants to control the equipment. Cloud-based voice recognition is available to enable voice control in smart home automation systems. [2] This speech recognition utilizes existing API services in the cloud. The use of embedded equipment and sensors is also used to build smart home networks so as to reduce energy consumption [3], these devices can control local home monitoring. Applications via the web are also built to be used as smart home monitoring, control commands are inputted via portable as a user interface [4]. The existence of the web will make the conditions of the house easier, but if the web server is not active, monitoring cannot be done.

In this research, we made the control system mobile and use voice as input in giving commands, making it easier to use. The system is designed to recognize patterns of human speech processing with sound frequency and amplitude parameters. The incoming voice signal will be processed by the computer for later use as a command. This process is better known as speech processing, which is voice recognition through processed voice samples. The control system or control can be said to be the relationship between the components that make up a system configuration, which will produce the expected system response. In this study, control was carried out on electronic equipment, namely 2 lights and 1 fan. The given control is stored in binary value conditions, namely 0 and 1. Condition 0 which means the status of electronic equipment is inactive, and condition 1 which means the status of the electronic equipment is active.

2. THEORY

2.1. Fast Fourier Transform

Fast Fourier Transform or FFT is an algorithm that is used to calculate the faster and more efficient than Discrete Fourier Transform (DFT) process. This FFT is used in various needs such as for signal processing and solving differential equations. The comparison of DFT and FFT calculations is that DFT calculations require O (N^2) arithmetic operations, while FFT can perform O (N log N) calculations for the same series calculation operations.

FFT is very suitable to be used to convert signals in the time domain to the frequency domain. For example, voice that has been recorded will be stored in the time domain, then by FFT it will be converted into frequency domain so that it is easier to analyze the voice spectrum. FFT is an algorithm that is optimized for DFT implementations. The signal is sampled over a period of time and divided into its frequency components [8].

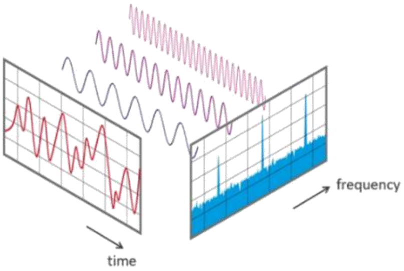


Figure 1. View of a signal in the time and frequency domain [8]

2.2. Dynamic Time Warping

The Dynamic Time Warping method or abbreviated as DTW is a method used to compare the similarities or calculate the distance between two time series from an input with unequal lengths of series. An example of use in this method is to detect a diversity of different sound patterns but the sounds spoken are the same. Just like when we want to recognize a voice of "hey" from someone's pronunciation, then we will find different sound patterns such as "hey (normal)" and "heeeyyyy (long)". This is the result of the voice of the same person pronouncing the sound "what" in different ways. This difference can be caused by the speed of sound or the amplitude of the sound. To be able to recognize the sound as the same sound we need a method that can equate different sound sequences so that they can be recognized as the same sound.

In each series, a measure of distance is placed, comparing the corresponding elements of two sequences. The distance between the two points is calculated using Euclidean Distance.

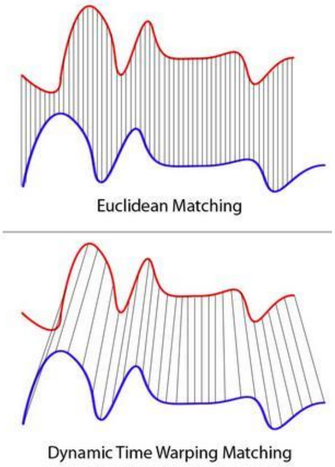


Figure 2. DTW Process using Euclidean Distance [9]

First, confirm that you have the correct template for your paper size. This template has been tailored for output on the A4 paper size.

2.3. Firebase

Firebase is a real time database service provider and backend as a service. An application that allows developers to create APIs to be synchronized to different clients and stored in the Firebase cloud. Firebase has many libraries that make it possible to integrate this service with Android, IOS, Java, Objective-C and Node.JS. Firebase databases are also accessible via the REST API. We use the REST API to post data which is then the Firebase client library that has been applied to the application being built that will retrieve data in real time.

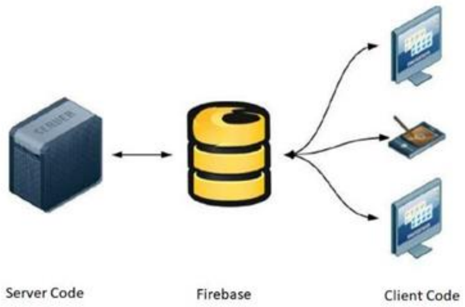


Figure 3. Firebase Workflow

We use the firebase server to secure data. For file hosting, Firebase provides hosting for static files with CND and SNL facilities.

3. METHOD

This system is divided into two parts. The first part is speech processing. Starting from recording voice input, then processing voice data until the system can translate these commands. The sound processing is run on a smartphone application. The second part is the implementation of the internet-of-things system. The application sends the detected voice command to Firebase using internet. Then the ESP8266 and WEMOS based microcontrollers check command data in real time in Firebase. WEMOS then translates the voice commands that have been sent to home electronic equipment. The design system is shown in Figure 4.

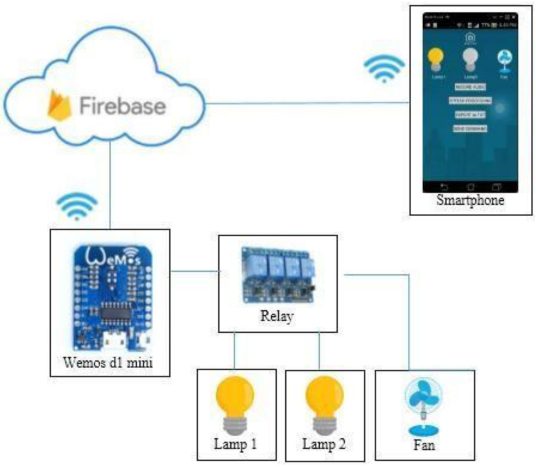
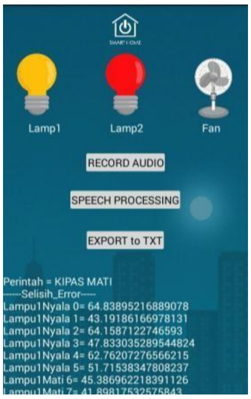


Figure 4. Design system

3.1. Android Application User Interface

All processes from voice recording and speech processing are executed in a Smartphone based application. To record the sound in the Android application using the buttons that have been prepared.



```
Perintah = KIPAS MATI
------Selisih_Error------
Lampu1Nyala 0= 64.83895216889078
Lampu1Nyala 1= 43.19186166978131
Lampu1Nyala 2= 64.1587122746593
Lampu1Nyala 3= 47.833035289544824
Lampu1Nyala 4= 62.76207276566215
Lampu1Nyala 5= 51.71538347808237
Lampu1Mati 6= 45.386962218391126
Lampu1Mati 7= 41.89817532575843
```

Figure 5. User Interface Smartphone Application

For a description of the application user interface is as follows:

1. “RECORD AUDIO” button is used to start voice recording.
2. "SPEECH PROCESSING" button is used to start processing the command recognition of the recorded voice.
3. "EXPORT to TXT" button is used to save each voice signal processing result into text for easier viewing.
4. Two lights and one fan display shows the equipment condition in real time [ON/OFF].

3.2. Hardware Simulation System

To turn on and turn off electronic equipment, several electronic circuits are needed. The main controls are the microcontroller, WEMOS and relays which are used to ON / OFF electronic devices.

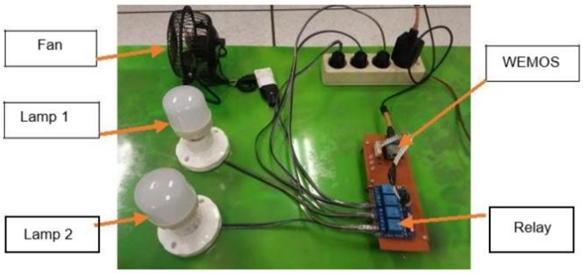


Figure 6. Implementation to the plant

4. RESULTS

The results of each step of voice processing using voice commands "letter / character", "word", and "sentence" are started from the sampling process to the Dynamic Time Warping process.

4.1. Sampling

The purpose of this process is to take a sample point from the voice data, in this case the sound file will be converted into a byte array data so that it can be used for further signal processing. The sampling frequency used is 16000Hz, where in 1 second there are 16000 sampling points.

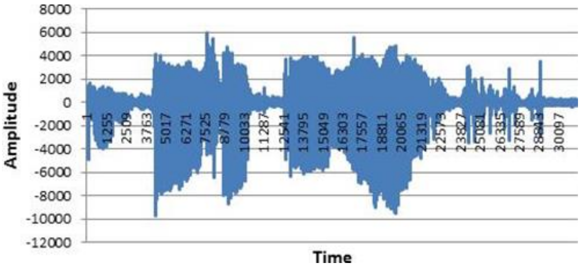


Figure 7. Signal sampling voice “Lampu satu nyala”

4.2. Front End Detection

Sound files that have been taken not only contain the information needed but also there is noise and also data that is not needed, for example the sound recorded outside the sound needed. Then this noise must be removed. The process of eliminating pauses or taking the beginning and end of the sound is done by dividing the sound signal into parts called frames. After that, each boundary value calculation is calculated, where the boundary value is obtained from the powr value, mean (mean), and standard deviation. Then we get the boundary value for each frame. Next is to calculate the average boundary value of all frames. After that the boundary value of each frame is compared with the average boundary value of the entire frame. Frames with a limit value that exceeds the average limit value will be collected and put together which is the beginning and end of the vote.

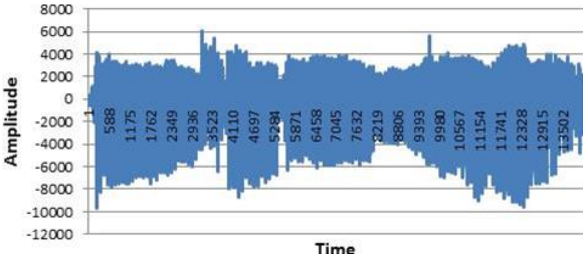


Figure 8. Front End Detection result from sampling

4.3. Frame Blocking

In the application that has been made, the resulting sound signal has a sampling frequency of 16000Hz, so:

- 1 second = 16000 sample points
- 10 ms = 16000/100 = 160 sample points
- so one frame consists of 160 sample points.

One window = 2 * frame = 2 * 160 = 320 sample points.

If M = frame, N = Window, and the amount of overlap is 10 milliseconds (one frame), then the frame distribution can be done with the Figure 9.

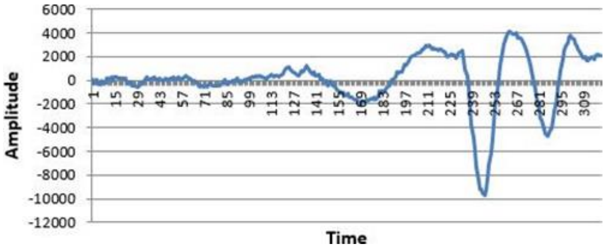


Figure 9. First window signal of voice "Lampu satu nyala"

The result of the frame blocking process is that the signal is divided into several frames and windows, without changing the value of the signal.

4.4. Windowing

The purpose of the windowing process is to change the discontinue signal multiplied by the window function to become a continuous signal. The windowing function used in this final project is window hamming.

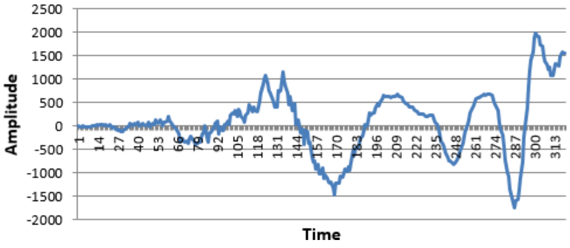


Figure 10 First window signal after windowing

4.5. Fast Fourier Transform

The FFT process requires the amount of data which is the number 2n. The data for each window used is 320. The number is not the power of 2, so the amount of data is changed to 512. Because the available data is 320, the rest contains number 0.

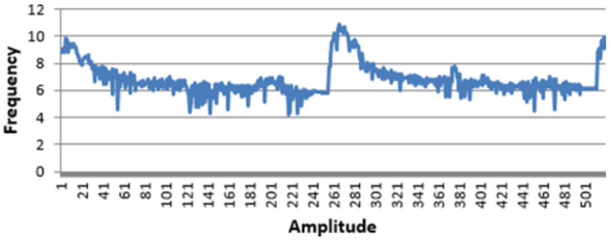


Figure 11 Spectrum signal

4.6. Autocorrelation

The result of the log processing is a spectrum. The spectrum signal is not used for the speech processing process in the feature extraction stage, but is only used for comparison with cepstrum.

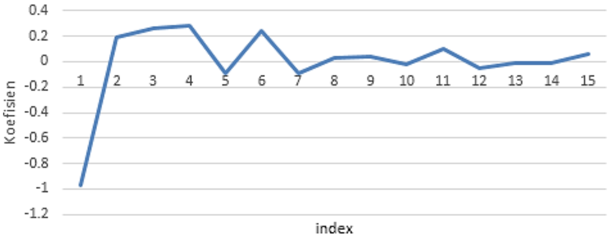


Figure 12 Autocorrelation result first window

4.7. LPC Analysis

The results of the LPC calculation autocorrelation analysis. This process is for voice feature extraction. The result of this process is the cepstrum of the sound signal. This cepstrum is used as a feature which is a feature or signal that can represent the total sound signal.

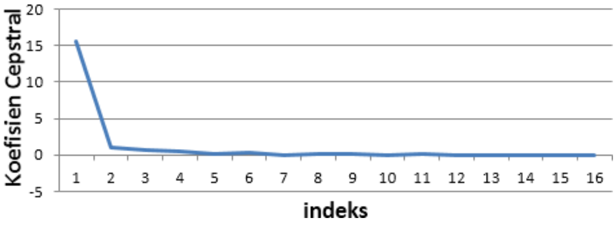


Figure 13 LPC Analysis result first window

4.8. LPC Analysis to cepstral coefficient

To ascertain whether the LPC analysis process is functioning properly, it can be checked by performing a Fast Fourier Transform calculation on cepstrum. Because cepstrums only amount according to the order specified (16), then to do FFT input cepstrum must be given zero padding so that the amount of data is 512. If the LPC analysis process is appropriate then the FFT results are cepstrum signals from the FFT signal windowing results. If it is the same, the next process can be carried out, namely the classification of features using the Dynamic Time Warping method.

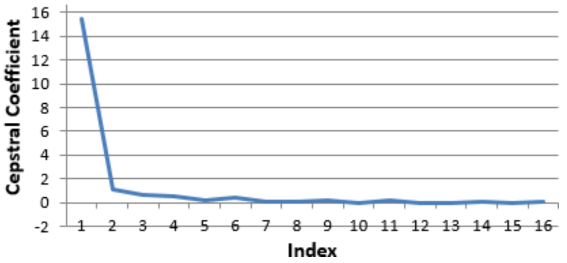


Figure 14 Feature extraction result first window

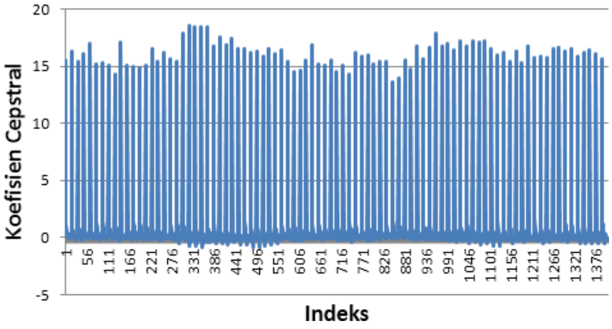


Figure 15 Feature extraction result full signal

4.9. Fast Fourier Transform LPC Analysis

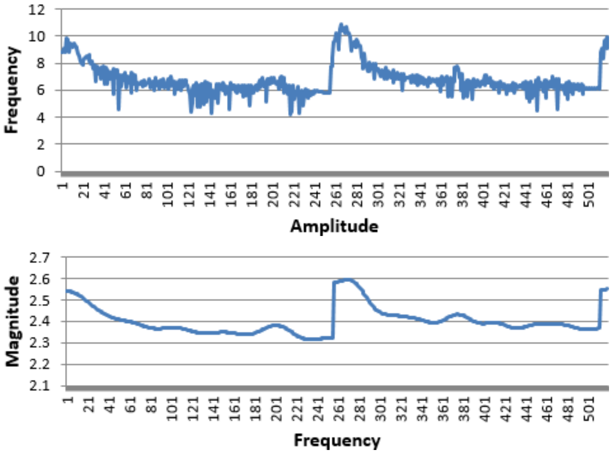
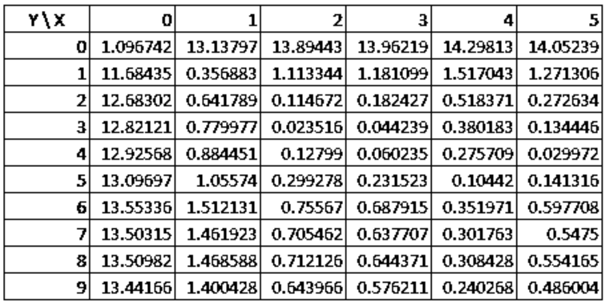


Figure 16 Comparing spectrum and spectrum first window

4.10. Dynamic Time Wraping

Step 1 – The feature is from LPC Analysis to cepstral coefficient result. First Feature x is the testing voice and the second feature Y is standard voice in database. Calculate the error value from each window of the two features.

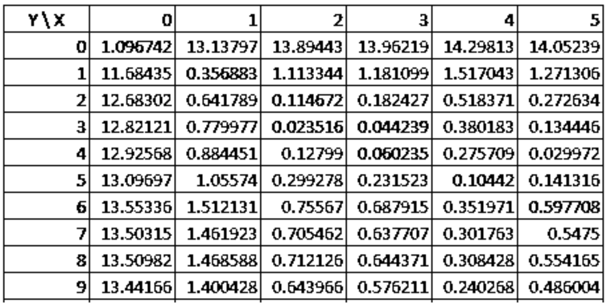
Table 1. Comparing Two Features



Y\X	0	1	2	3	4	5
0	1.096742	13.13797	13.89443	13.96219	14.29813	14.05239
1	11.68435	0.356883	1.113344	1.181099	1.517043	1.271306
2	12.68302	0.641789	0.114672	0.182427	0.518371	0.272634
3	12.82121	0.779977	0.023516	0.044239	0.380183	0.134446
4	12.92568	0.884451	0.12799	0.060235	0.275709	0.029972
5	13.09697	1.05574	0.299278	0.231523	0.10442	0.141316
6	13.55336	1.512131	0.75567	0.687915	0.351971	0.597708
7	13.50315	1.461923	0.705462	0.637707	0.301763	0.5475
8	13.50982	1.468588	0.712126	0.644371	0.308428	0.554165
9	13.44166	1.400428	0.643966	0.576211	0.240268	0.486004

Step 2 - Determine the wrapping path

Table 2. Dynamic time wrapping



Y\X	0	1	2	3	4	5
0	1.096742	13.13797	13.89443	13.96219	14.29813	14.05239
1	11.68435	0.356883	1.113344	1.181099	1.517043	1.271306
2	12.68302	0.641789	0.114672	0.182427	0.518371	0.272634
3	12.82121	0.779977	0.023516	0.044239	0.380183	0.134446
4	12.92568	0.884451	0.12799	0.060235	0.275709	0.029972
5	13.09697	1.05574	0.299278	0.231523	0.10442	0.141316
6	13.55336	1.512131	0.75567	0.687915	0.351971	0.597708
7	13.50315	1.461923	0.705462	0.637707	0.301763	0.5475
8	13.50982	1.468588	0.712126	0.644371	0.308428	0.554165
9	13.44166	1.400428	0.643966	0.576211	0.240268	0.486004

Step 3 - Calculate the total difference

After finding the path with the smallest total error value, then the values for each column on the path are added up. Starting from column 0.0 to the last column, namely according to the number of feature windows x, the number of feature windows y. This total error value will be compared with the other total error results according to the number of standard signals or the database being compared. The smallest total error value is the feature that is considered to have the highest similarity.

The first test is the standard data in the form of orders of A, I, U, E, O and S.

Table 3. Result using letter commands

Input Voice	Standard Voice					
	A	I	U	E	O	S
A	**25.2**	56.1	49.5	39.4	43.2	35.9
A	**23.4**	52.0	45.2	34.6	37.6	33.5
A	**30.9**	49.3	47.9	36.9	37.3	47.6
A	**25.5**	52.6	47.6	39.0	41.5	39.1
A	**26.7**	53.1	40.8	44.4	27.0	36.4
I	36.5	**21.0**	28.6	23.1	28.1	38.9
I	37.2	**22.4**	29.2	24.4	29.8	29.2
I	45.7	**22.2**	28.1	30.7	30.4	36.2
I	42.0	**26.6**	34.4	31.4	34.8	39.2
I	44.0	**25.3**	29.4	30.6	27.7	38.8
U	36.2	39.5	**22.4**	38.0	32.4	36.9
U	36.2	25.2	**15.9**	31.0	27.0	31.9
U	38.9	28.0	**20.1**	26.8	26.9	39.4
U	34.4	37.7	**18.5**	36.3	27.0	34.2
U	44.1	51.8	**28.6**	47.8	45.7	55.6
E	41.0	36.2	31.6	**27.7**	35.4	29.1

Input Voice	Standard Voice					
	A	I	U	E	O	S
E	39.7	39.4	48.4	**18.7**	28.3	26.2
E	47.3	34.2	39.9	**21.8**	28.4	26.3
E	30.7	41.6	31.0	**22.5**	23.1	33.3
E	38.5	39.3	43.4	**23.8**	31.7	28.1
O	31.6	41.5	36.5	43.8	**22.1**	56.0
O	33.9	41.9	34.3	38.8	**20.2**	35.9
O	38.1	35.1	32.3	40.7	**15.5**	40.7
O	35.9	44.7	30.6	33.3	**20.3**	43.1
O	35.7	53.5	32.5	41.3	**30.9**	41.0
S	28.8	45.6	33.6	35.4	39.9	**17.3**
S	41.3	34.8	38.8	30.0	35.9	**18.0**
S	31.9	36.0	40.9	30.6	34.9	**22.1**
S	35.8	38.9	40.2	33.3	37.8	**19.0**
S	31.9	40.8	31.1	23.9	34.8	**17.4**
Success rate	100%	100%	100%	100%	100%	100%

The second test used standard voice and commands in the form of words as follows: satu, dua, tiga, empat, lima, enam.

Table 4. Result using words commands

Input Voice	Standard Voice					
	Satu	Dua	Tiga	Empat	Lima	Enam
Satu	**21.8**	27.3	31.8	25.1	34.7	31.5
Satu	**20.5**	21.3	34.0	28.2	32.4	29.0
Satu	**22.6**	26.8	35.3	26.4	33.6	26.5
Satu	**24.1**	38.3	42.8	50.2	49.1	47.4
Satu	**18.8**	23.3	32.5	23.0	32.7	21.3
Dua	29.2	**20.6**	34.8	27.2	30.6	31.9
Dua	32.4	**25.3**	27.8	25.9	28.9	28.5
Dua	35.8	**12.2**	32.5	20.1	21.3	17.3
Dua	28.6	**14.5**	26.8	31.2	25.7	28.1
Dua	34.6	**20.3**	35.8	26.2	29.8	22.7
Tiga	32.4	24.6	**22.0**	30.7	24.7	30.0
Tiga	45.1	24.1	**18.3**	30.3	21.3	29.1
Tiga	35.2	26.9	**21.5**	28.0	23.3	30.8
Tiga	34.5	27.6	**20.0**	30.5	20.5	27.0
Tiga	39.8	28.6	32.1	27.5	**27.0**	28.6
Empat	26.1	28.9	30.6	**24.3**	31.0	35.5
Empat	30.1	**20.4**	29.0	21.1	22.7	21.5
Empat	33.3	25.2	26.7	**21.3**	27.2	24.6
Empat	34.3	23.8	30.0	20.4	25.6	**18.0**
Empat	37.2	27.5	44.6	**22.8**	23.3	23.3
Lima	41.4	23.3	24.5	36.4	**17.6**	29.5
Lima	31.6	25.5	29.5	21.8	**18.2**	21.9
Lima	42.6	23.2	29.3	23.1	**21.4**	24.4
Lima	46.1	28.0	29.2	24.6	22.5	**20.6**
Lima	48.8	22.1	38.2	29.7	**19.0**	25.2
Enam	30.1	19.1	27.1	20.2	17.9	**16.7**
Enam	35.3	25.6	23.1	18.6	23.0	**15.9**
Enam	33.2	24.8	31.5	27.9	26.6	**14.5**
Enam	37.9	21.7	33.7	19.3	23.2	**17.3**
Enam	30.8	23.0	29.5	18.5	21.9	**17.1**
Success rate	100%	100%	80%	60%	80%	100%

The third test used standard voice and commands in the form of sentences as follows:

- C1 = Lampu satu nyala (Lamp 1 ON)
- C2 = Lampu satu mati (Lamp 1 OFF)
- C3 = Lampu dua nyala (Lamp 2 ON)
- C4 = Lampu dua mati (Lamp 2 OFF)
- C5 = Kipas nyala (Fan ON)
- C6 = Kipas mati (Fan OFF)

Table 5. Result using sentence commands

No	Input Voice	Standard Voice					
		C1	C2	C3	C4	C5	C6
1	C1	**21.1**	25.6	27.0	24.9	33.7	24.3
2	C1	**26.3**	29.8	31.2	32.7	44.1	30.3
3	C1	38.0	**27.6**	32.1	29.0	31.6	29.8
4	C1	33.4	33.9	**25.8**	27.3	37.2	31.9
5	C1	**20.3**	25.6	26.5	22.3	29.2	22.2
6	C2	**21.9**	26.8	22.5	25.1	35.9	42.5
7	C2	**20.2**	24.7	26.8	24.7	38.1	32.2
8	C2	31.9	**26.1**	26.3	28.0	36.7	30.9
9	C2	47.0	27.5	34.8	**23.8**	32.2	32.5
10	C2	26.8	27.6	36.2	**26.2**	40.0	35.8
11	C3	33.5	33.9	**27.1**	32.3	38.0	34.2
12	C3	45.9	38.7	**27.2**	33.5	37.7	37.9
13	C3	36.9	31.9	**30.3**	32.4	39.0	35.9
14	C3	45.4	44.3	35.1	**26.4**	45.7	43.4
15	C3	41.4	42.7	**30.5**	31.8	47.6	40.3
16	C4	35.2	32.8	27.2	**26.4**	40.8	36.0
17	C4	24.8	23.8	**21.3**	28.3	32.6	33.0
18	C4	36.6	31.9	**29.3**	30.8	43.5	42.6
19	C4	24.9	27.8	23.4	**18.2**	32.0	26.1
20	C4	27.5	25.7	**20.8**	24.3	33.8	28.8
21	C5	52.1	42.1	46.4	39.8	**28.2**	35.8
22	C5	39.8	39.3	32.3	31.2	**27.1**	28.1
23	C5	43.4	31.6	47.7	27.4	**19.0**	23.1
24	C5	41.3	27.5	41.4	22.7	28.8	**19.8**
25	C5	31.0	32.2	33.1	24.3	24.3	**22.9**
26	C6	33.4	36.5	44.2	41.0	40.8	**24.7**
27	C6	55.5	36.6	34.7	36.4	28.2	**24.5**
28	C6	35.4	35.6	30.8	31.4	31.6	**24.9**
29	C6	41.2	38.3	35.9	32.1	**26.2**	28.2
30	C6	37.9	39.1	39.4	30.8	30.5	**30.4**
Success rate		60%	20%	80%	40%	60%	80%

The command detection process uses the DTW method by finding the smallest total Euclidian distance value. Commands that are considered to be the closest to the standard signal are in blocks with bold values.

Table 6. Classification result

Input Signal	Standard Signal						Average Success Rate
	Command 1	Command 2	Command 3	Command 4	Command 5	Command 6	
Testing orders in the form of letters	100%	100%	100%	100%	100%	100%	100%
Testing orders in the form of words	100%	100%	80%	60%	80%	100%	87%
Testing orders in the form of sentences	60%	20%	80%	40%	60%	80%	57%

5. CONCLUSION

The more standard references or data, the better the results of the classification process using DTW, but the disadvantage is the longer the classification time will be. The test result in the form of "letters / characters" has the highest success rate at 100%. The letter includes the letters "A, I, U, E, O, S". Whereas in testing the commands in the form of "words" have a success rate of 87%, the words entered are "satu, dua, tiga, empat, lima, enam". In testing the command in the form of a sentence has a success rate of 57%. The sentence pronounced is "lampu satu nyala, lampu satu mati, lampu dua nyala, lampu dua mati, kipas nyala, kipas mati". The more command characters used, the lower the speech recognition success rate.

REFERENCES

[1] X. Wang and J. Li, "Design of Intelligent Home Security Monitoring System Based on Android," 2018 2nd IEEE Advanced Information Management,Communicates,Electronic and Automation Control Conference (IMCEC), Xi'an, 2018, pp. 2621-2624, doi: 10.1109/IMCEC.2018.8469543.

[2] M. Matić, I. Stefanović, U. Radosavac and M. Vidaković, "Challenges of integrating smart home automation with cloud based voice recognition systems," 2017 IEEE 7th International Conference on Consumer Electronics - Berlin (ICCE-Berlin), Berlin, 2017, pp. 248-249, doi: 10.1109/ICCE-Berlin.2017.8210640.

[3] E. Irmak, A. Köse and G. Göçmen, "Simulation and ZigBee based wireless monitoring of the amount of consumed energy at smart homes," 2016 IEEE International Conference on Renewable Energy Research and Applications (ICRERA), Birmingham, 2016, pp. 1019-1023, doi: 10.1109/ICRERA.2016.7884488.

[4] D. Pavithra and R. Balakrishnan, "IoT based monitoring and control system for home automation," 2015 Global Conference on Communication Technologies (GCCT), Thuckalay, 2015, pp. 169-173, doi: 10.1109/GCCT.2015.7342646.

[5] S. Milivojša, S. Ivanović, T. Erić, M. Antić and N. Smiljković, "Implementation of voice control interface for smart home automation system," 2017 IEEE 7th International Conference on Consumer Electronics - Berlin (ICCE-Berlin), Berlin, 2017, pp. 263-264, doi: 10.1109/ICCE-Berlin.2017.8210646.

[6] Manikandan J., "Design and evaluation of wireless home automation systems," 2016 IEEE 1st International Conference on Power Electronics, Intelligent Control and Energy Systems (ICPEICES), Delhi, 2016, pp. 1-5, doi: 10.1109/ICPEICES.2016.7853323.

[7] T. B. Amin and I. Mahmood, "Speech Recognition using Dynamic Time Warping," 2008 2nd International Conference on Advances in Space Technologies, Islamabad, 2008, pp. 74-79, doi: 10.1109/ICAST.2008.4747690.

[8] https://www.nti-audio.com/en/support/know-how/fast-fourier-transform-fft

[9] https://towardsdatascience.com/dynamic-time-warping-3933f25fcdd