Design of Voice Command In Smart Wheelchair Using Hmm Method

1st Achmad Hidayatno[1], 2nd Sumardi[2], 3rd David Kristian Adi Putra[3], 4th M Hilal Bayu Aji[4], 5th Arum Patmadani[5]
achmad.hidayatno@gmail.com[1], davidkristian17@gmail.com[1]

Department of Electrical Engineering, Diponegoro University Semarang, Indonesia

Abstract: A wheelchair is a tool used for people with disabilities to be able to move from one place to another. Wheelchairs are used not only in a horizontal place but can be used in a higher place. Wheelchairs are also not only used for people with foot disabilities but are used by hospital patients, the elderly, and people who are at high risk of injury when walking alone. In general, a wheelchair that is often used is a standard wheelchair that is used in hospitals with operations using human labor or assisted by others. However, the use of a standard wheelchair is considered quite difficult because it requires enough energy to be able to move the wheelchair if no one else is helping. Along with the times, motorized wheelchairs have been built and operated using joysticks. The addition of a motor to the wheel helps the user so that the user does not need to use power anymore in moving the wheelchair. In this paper, we develops the ability of a wheelchair that can move using voice instructions. We use the Hidden Markov Model (HMM) method which is utilized in the Pocketsphinx library. This library supports the speech recognition feature to recognize spoken words that implemented at Raspberry Pi mini-computer with the Raspbian operating system. Based on the test results, the average success the word of "maju" 97%, the word of "mundur" 88.5%, the word of "kanan" 94%, the word of "kiri" 92%, the word of "stop" 98.5%, the word of "tamu" 97.5%, the word of "tidur" 91.5%, the word of "makan" 94%, the word of "toilet" 89%.

Keywords: wheelchair, voice instructions, hmm, pocketsphinx, raspberry pi

1. Introduction

A wheelchair is a tool used for people with leg disabilities to be able to move from one place to another [1]. Wheelchairs are used not only in horizontal Place but can be used at a higher place. Based on Susenas 2012, people with disabilities in Indonesia amounted to 2.45% [2]. In 2012 Susenas released data that represent more than one type of disability was 39.97% [2], then to the percentage of seeing and walking or climbing up the stairs was 10.26% [2].

In general a wheelchair that is often used is a standard wheelchair that is used in hospitals with operations using human labor or assisted by others. However, the use of a standard wheelchair feels quite difficult because it requires enough energy to be able to move the wheelchair if no one else to help.

Along with the times, has been made a wheelchair that has a better ability than a standard wheelchair. Types of wheelchairs that are sold today are numerous, ranging from manual wheelchairs to automatic wheelchairs using motorcycles and operated with joysticks [3]. The addition of the motor on the wheels helps the user so that the user does not need to use more power in moving the wheelchair. To be able to move as desired, users can simply use the joystick

ICITID 2021, August 30, Yogyakarta, Indonesia

DOI 10.4108/eai.30-8-2021.2311520

and the wheelchair can move. Wheelchairs that use joysticks for wheelchair movements do not fulfill user needs [4].

In this design, we develop the ability of a wheelchair that can move using voice instructions. Voice recognition is later as input that has previously been determined any word and put into a collection of words so that the words that are spoken will later be matched with files that have been made and the wheelchair can move according to the commands spoken.

Hidden Markov Model is a speech recognition method that is used as a method in this design. This method is a statistical model that can be considered a simple dynamic Bayesian network [5]. Implementation of this method is utilized in the PocketSphinx library [6]. Speech recognition features in this library is used to detect human voice [7]. To use this library, the author uses a Raspberry Pi 3 [8] minicomputer with the Raspbian Operating System.

2. Software Design

2.1. Building PocketSphinx

PocketSphinx is a lightweight version of Sphinx using the C programming language. This library implements HMM in its application. To be able to use the PocketSphinx application, you must first install the Raspbian operating system with the following steps:

1. cd ~/
2. wget http://sourceforge.net/projects/cmusphinx/files/pocketsphinx/5prealpha/pocketsphinx-5prealpha.tar.gz.
3. tar -zxvf pocketsphinx-5prealpha.tar.gz
4. cd ./pocketsphinx-5prealpha
5. ./configure
6. make clean all
7. make check
8. sudo make install

On the Raspberry terminal screen then run one by one command to install PocketSphinx. Each command must be installed successfully and if an error occurs then repeat the installation.

2.2. Building SphinxBase

SphinxBase is needed to keep a log of all states and parameters that have been used by the system in the previous process to speed up the process if the state and parameters are accessed again. To be able to use SphinxBase you must first install the Raspbian operating system with the following steps:

1. cd ~/
2. wget http://sourceforge.net/projects/cmusphinx/files/sphinxbase/5prealpha/sphinxbase-5prealpha.tar.gz
3. tar -zxvf ./sphinxbase-5prealpha.tar.gz
4. cd ./sphinxbase-5prealpha
5. ./configure --enable-fixed
6. make clean all
7. make check
8. sudo make install

On the Raspberry terminal screen then run one by one the command to install SphinxBase. Each command must be installed successfully and if an error occurs then repeat the installation.

2.3. Creating a Word File

The list of words that you want to recognize as input from the system will first be made in a file *.txt to be uploaded to http://www.speech.cs.cmu.edu/tools/lmtool-new.html to compile. The commands to be used are ‘maju’, ‘mundur’, ‘kanan’, ‘kiri’, ‘stop’, ‘tamu’, ‘tidur’, ‘makan’, and ‘toilet’. This file is the command word data that is used to recognize the voice.

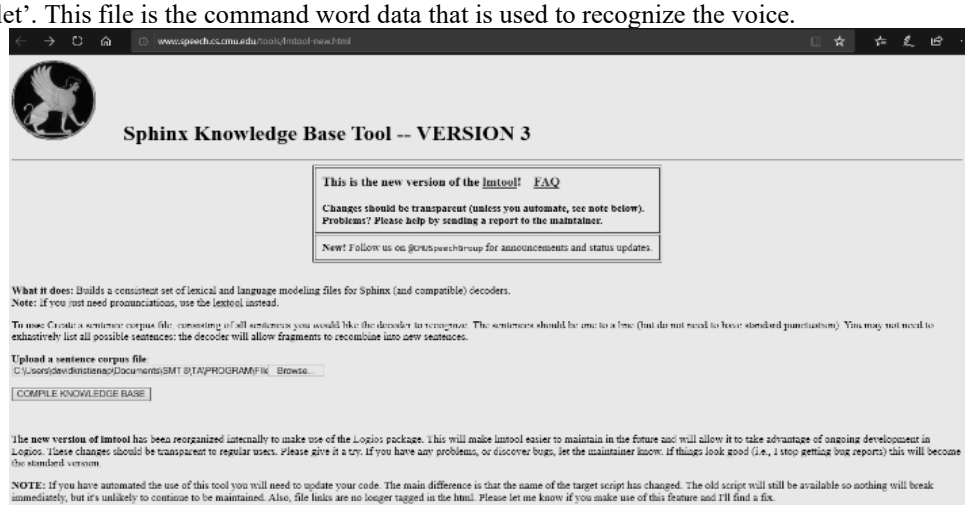

Fig. 1. Initial Display of the Sphinx Knowledge Base Tool

Figure 1 show the initial display of the Sphinx Knowledge Base Tool. This tool can be used to get dictionary files and language models as matching words spoken. After the compilation is complete, the dictionary file and language model can be downloaded in a format *.dic for dictionary files and *.lm for the language model.

Table 1. Words and Their Pronunciation

Word	Pronunciation
KANAN	K EY N AH N
KIRI	K IH R IY
MAJU	M AE JH UW
MAKAN	M EY K AH N
MUNDUR	M AH N D AH R
STOP	S T AA P
TAMU	T AE M UW
TIDUR	T IH D AH R
TOILE	T OY L AH T

Dictionary file contains the pronunciation of words, shown on Table 1, that have been downloaded on the Sphinx Knowledge Base Tool. Dictionary files are used to match spoken words so they can produce word recognition according to the spoken words. Because PocketSphinx doesn't support Indonesian in pronunciation, then uses English pronunciation.

```
Language model created by QuickLM on Wed Aug 21 01:35:43 EDT 2019
Copyright (c) 1996-2010 Carnegie Mellon University and Alexander I. Rudnicky

The model is in standard ARPA format, designed by Doug Paul while he was at MITRE.

The code that was used to produce this language model is available in Open Source.
Please visit http://www.speech.cs.cmu.edu/tools/ for more information

The (fixed) discount mass is 0.5. The backoffs are computed using the ratio method.
This model based on a corpus of 9 sentences and 11 words

\data\
ngram 1=11
ngram 2=18
ngram 3=9

\1-grams:
-0.7782 </s> -0.3010
-0.7782 <s> -0.2218
-1.7324 KANAN -0.2218
-1.7324 KIRI -0.2218
-1.7324 MAJU -0.2218
-1.7324 MAKAN -0.2218
-1.7324 MUNDUR -0.2218
-1.7324 STOP -0.2218
-1.7324 TAMU -0.2218
-1.7324 TIDUR -0.2218
-1.7324 TOILET -0.2218

\2-grams:
-1.2553 <s> KANAN 0.0000
-1.2553 <s> KIRI 0.0000
-1.2553 <s> MAJU 0.0000
-1.2553 <s> MAKAN 0.0000
-1.2553 <s> MUNDUR 0.0000
-1.2553 <s> STOP 0.0000
```

Fig. 2. Language Model

In Figure 2 is a language model that has been downloaded on the Sphinx Knowledge Base Tool that is used to generate vocabulary grammar in voice recognition applications. The language model toolkit is based on the uni-gram modeling, namely n-gram size 1, bi-gram, n-gram size 2, and tri-gram, n-gram size 3 of the language to be recognized. The n-gram model is a probabilistic language model for predicting the next word.

3. Testing and Results

3.1. Testing the Word of "Maju"

Table 2. Testing the Word of "Maju"

Percobaan	Pengujian ke-										Total	%
	1	2	3	4	5	6	7	8	9	10		
1	a	a	a	a	a	a	a	a	a	a	10	100
2	g	i	a	a	a	a	a	a	a	a	8	80
3	a	a	a	a	a	a	a	a	a	a	10	100
4	a	a	a	a	a	a	a	a	a	a	10	100
5	a	a	a	a	a	a	a	a	a	a	10	100
6	a	a	a	a	a	a	a	a	a	a	10	100
7	a	a	a	a	a	a	a	a	a	a	10	100
8	a	a	a	a	a	a	a	a	a	a	10	100
9	a	a	b	a	a	a	a	a	a	a	9	90
10	a	a	a	a	a	a	a	f	f	a	8	80
11	a	a	a	a	a	a	a	a	a	a	10	100
12	a	a	a	a	a	a	a	a	a	a	10	100
13	a	a	a	a	a	a	a	a	a	a	10	100
14	a	a	a	a	a	a	a	a	a	a	10	100
15	a	a	a	a	a	a	a	a	a	a	10	100
16	a	a	a	a	a	a	a	a	a	a	10	100
17	a	a	a	a	a	a	a	a	a	a	10	100
18	a	a	a	a	a	a	a	a	a	a	10	100
19	a	a	a	a	a	a	a	a	a	a	10	100
20	b	a	a	a	a	a	a	a	a	a	9	90
Rata-rata												97

Based on Table 2, from 20 experiments with each saying 10 times the word of "maju" can be seen the average success of the system in recognizing the word of "maju" and sending the character 'a' is 97%. The system that not recognizes the word of "maju" and displays characters instead of 'a' can be caused by noise, differences in the pronunciation of each person's words because the library uses English pronunciation, and the similarity of words so the system can recognize words other than the word of "maju" .

3.2. Testing the Word of "Mundur"

Table 3. Testing the Word of "Mundur"

Percobaan	Pengujian ke-										Total	%
	1	2	3	4	5	6	7	8	9	10		
1	i	b	i	b	b	b	b	b	b	b	8	80
2	b	b	b	i	b	b	b	b	b	b	9	90
3	b	b	b	b	b	f	f	b	b	b	8	80
4	b	b	b	i	b	b	i	b	b	b	8	80
5	b	b	b	b	b	b	b	b	b	i	9	90
6	b	i	b	b	b	b	b	b	b	b	9	90
7	b	i	b	b	b	i	b	b	b	b	8	80
8	b	b	b	b	b	b	b	b	b	b	10	100
9	b	b	b	b	b	b	i	b	b	b	9	90
10	b	b	b	b	i	i	b	b	b	b	8	80
11	b	b	b	b	b	b	b	b	b	b	10	100
12	b	b	b	b	b	b	b	b	b	b	10	100
13	b	b	b	b	b	b	b	b	b	b	10	100
14	b	i	b	b	b	b	b	b	b	b	9	90
15	b	b	b	b	i	i	b	b	b	b	8	80
16	b	b	b	b	b	b	b	b	b	b	10	100
17	b	b	d	b	g	b	b	b	b	b	8	80
18	b	b	i	b	b	b	b	b	c	b	8	80
19	b	b	b	b	b	b	b	b	b	b	10	100
20	b	b	b	b	b	i	b	b	b	i	8	80
Rata-rata												88.5

Table 3 shows the test results of 20 experiments with each saying 10 times the word of “mundur”. It can be seen that the average success of the system in recognizing the word of “mundur” and sending the 'b' character is 88.5%. Systems that not recognize the word of “mundur” and display characters instead of 'b' can be caused by noise, differences in the pronunciation of each person's words because the library uses English pronunciation, and similarity of words so the system can recognize words other than the word of “mundur”.

3.3. Testing the Word of “Kanan”

Table 4. Testing the Word of “Kanan”

Percobaan	Pengujian ke-										Total	%
	1	2	3	4	5	6	7	8	9	10		
1	c	c	c	c	c	c	c	c	c	c	10	100
2	c	g	c	c	c	c	b	c	c	c	8	80
3	c	f	c	c	f	c	c	c	c	c	8	80
4	c	b	c	c	c	b	c	c	c	c	8	80
5	c	c	c	c	a	f	c	c	c	c	8	80
6	c	c	c	c	c	c	c	c	c	c	10	100
7	c	c	c	c	c	c	c	c	c	c	10	100
8	c	c	c	c	c	c	c	c	c	c	10	100
9	c	c	c	c	c	c	c	c	c	c	10	100
10	c	c	c	c	c	c	c	c	c	c	10	100
11	c	c	c	c	c	c	c	c	c	c	10	100
12	c	c	c	c	c	c	c	c	c	c	10	100
13	c	c	c	c	c	c	c	c	c	c	10	100
14	c	c	c	c	c	c	c	c	c	c	10	100
15	c	c	c	c	c	c	c	c	c	c	10	100
16	c	c	c	c	c	c	c	c	c	c	10	100
17	c	c	c	c	c	c	c	c	c	c	10	100
18	c	c	c	c	f	f	c	c	c	c	8	80
19	c	c	c	c	c	c	c	c	c	c	10	100
20	c	c	c	c	c	f	c	c	f	c	8	80
Rata-rata												94

Table 4 shows the test results of 20 experiments with each saying 10 times the word of “kanan” can be seen the average success of the system in recognizing the word of “kanan” and sending the 'c' character is 94%. The system that not recognizes the word of “kanan” and displays characters instead of 'c' can be caused by noise, differences in the pronunciation of each person's words because the library uses English pronunciation, and the similarity of words so the system can recognize words other than the word of “kanan”.

3.4. Testing the Word of “Kiri”

Table 5. Testing the Word of “Kiri”

Percobaan	Pengujian ke-										Total	%
	1	2	3	4	5	6	7	8	9	10		
1	d	d	d	d	d	d	d	d	d	d	10	100
2	d	d	d	d	d	d	f	f	d	d	8	80
3	f	d	d	d	d	d	d	d	d	d	9	90
4	d	d	d	d	d	d	d	d	d	d	10	100
5	d	d	d	f	d	d	g	d	d	d	8	80
6	d	d	d	d	d	d	d	g	d	g	8	80
7	d	d	d	d	d	d	d	d	d	d	10	100
8	d	d	d	d	d	d	d	d	f	d	9	90
9	d	d	d	d	d	d	d	d	d	d	10	100
10	d	d	d	c	c	d	d	d	d	d	8	80
11	d	d	d	d	d	d	d	d	d	d	10	100
12	d	d	d	d	d	d	d	d	d	c	9	90
13	d	c	d	d	d	d	d	d	c	d	8	80
14	d	d	d	d	d	d	d	d	d	d	10	100
15	d	d	d	d	d	c	c	d	d	d	8	80
16	d	d	d	d	d	d	d	d	d	d	10	100
17	d	d	d	d	d	d	d	d	d	d	10	100
18	d	d	d	d	d	d	d	c	d	d	9	90
19	d	d	d	d	d	d	d	d	d	d	10	100
20	d	d	d	d	d	d	d	d	d	d	10	100
Rata-rata												92

Based on the test results in Table 5, 20 experiments with each saying 10 times the word of “kiri” can be seen the average success of the system in recognizing the word of ”kiri” and sending the 'd' character is 92%. The system that not recognizes the word of “kiri” and displays instead of ‘d’ characters can be caused by noise, differences in the pronunciation of each person's words because the library uses English pronunciation, and similarity of words so the system can recognize words other than the word of “kiri”.

3.5. Testing the Word of “Stop”

Table 6. Testing the Word of “Stop”

Percobaan	Pengujian ke-										Total	%
	1	2	3	4	5	6	7	8	9	10		
1	e	e	e	e	e	e	e	e	e	e	10	100
2	e	e	e	e	e	e	e	e	e	e	10	100
3	e	e	e	e	e	e	e	e	e	e	10	100
4	e	e	e	i	e	c	e	e	e	e	8	80
5	e	e	e	e	e	e	e	e	e	e	10	100
6	e	e	e	e	e	e	e	e	e	e	10	100
7	e	e	e	e	e	e	e	e	e	e	10	100
8	e	e	e	e	e	e	e	e	e	e	10	100
9	e	e	e	e	e	e	e	e	e	e	10	100
10	e	e	g	e	e	e	e	e	e	e	9	90
11	e	e	e	e	e	e	e	e	e	e	10	100
12	e	e	e	e	e	e	e	e	e	e	10	100
13	e	e	e	e	e	e	e	e	e	e	10	100
14	e	e	e	e	e	e	e	e	e	e	10	100
15	e	e	e	e	e	e	e	e	e	e	10	100
16	e	e	e	e	e	e	e	e	e	e	10	100
17	e	e	e	e	e	e	e	e	e	e	10	100
18	e	e	e	e	e	e	e	e	e	e	10	100
19	e	e	e	e	e	e	e	e	e	e	10	100
20	e	e	e	e	e	e	e	e	e	e	10	100
Rata-rata												98.5

Based on Table 6, from 20 experiments with each saying 10 times the word of “stop” can be seen the average success of the system in recognizing the word of “stop” and sending the character 'e' is 98.5%. Systems that recognize words instead of “stop” and display characters instead of 'e' can be caused by noise.

3.6. Testing the Word of “Tamu”

Table 7. Testing the Word of “Tamu”

Percobaan	Pengujian ke-										Total	%
	1	2	3	4	5	6	7	8	9	10		
1	f	f	f	f	f	f	f	f	f	f	10	100
2	f	f	f	f	f	f	f	f	f	f	10	100
3	f	f	f	f	f	f	f	f	f	f	10	100
4	f	f	f	f	f	f	f	f	f	f	10	100
5	f	f	f	f	f	f	f	f	f	f	10	100
6	f	f	f	f	f	f	f	f	f	f	10	100
7	f	f	f	f	f	f	f	f	f	f	10	100
8	f	f	f	e	f	f	f	f	f	e	8	80
9	g	f	f	i	f	f	f	f	f	f	8	80
10	f	f	f	f	f	f	f	f	f	f	10	100
11	f	f	f	f	f	f	f	f	f	f	10	100
12	f	f	f	f	f	f	f	f	f	f	10	100
13	f	f	e	f	f	f	f	f	f	f	9	90
14	f	f	f	f	f	f	f	f	f	f	10	100
15	f	f	f	f	f	f	f	f	f	f	10	100
16	f	f	f	f	f	f	f	f	f	f	10	100
17	f	f	f	f	f	f	f	f	f	f	10	100
18	f	f	f	f	f	f	f	f	f	f	10	100
19	f	f	f	f	f	f	f	f	f	f	10	100
20	f	f	f	f	f	f	f	f	f	f	10	100
Rata-rata												97.5

Table 7 shows the test results from 20 experiments with each saying 10 times the word of “tamu”can be seen the average success of the system in recognizing the word of “tamu” and sending the 'f' character is 97.5%. The system that can’t recognizes word “tamu” and displays not-f characters can be caused by noise, differences in the pronunciation of each person's words because the library uses English pronunciation, and the similarity of words so the system can recognize words other than the word of “tamu”.

3.7. Testing the Word of “Tidur”

Table 8. Testing the Word of "Tidur"

Percobaan	Pengujian ke-										Total	%
	1	2	3	4	5	6	7	8	9	10		
1	h	g	g	g	g	g	h	g	g	g	8	80
2	g	g	g	g	g	g	g	g	c	g	9	90
3	g	g	g	g	h	g	h	g	g	g	8	80
4	g	g	g	g	g	g	g	g	g	g	10	100
5	i	g	g	g	g	g	g	g	g	a	8	80
6	g	g	g	g	i	g	g	g	g	g	9	90
7	g	g	g	d	g	g	g	g	g	g	9	90
8	g	g	g	g	g	g	g	g	g	g	10	100
9	g	g	g	i	g	g	g	g	g	g	9	90
10	g	c	g	g	g	g	g	g	g	g	9	90
11	g	g	g	g	g	g	g	d	g	d	8	80
12	g	g	g	g	g	g	g	g	g	g	10	100
13	g	g	g	g	g	g	g	g	g	g	10	100
14	g	g	g	g	g	g	g	g	g	g	10	100
15	g	g	g	g	g	g	g	g	g	g	10	100
16	g	g	g	g	g	g	d	g	g	g	9	90
17	g	g	g	g	g	g	g	g	g	g	10	100
18	g	g	g	g	g	c	g	g	g	g	9	90
19	g	g	g	g	g	g	g	g	g	g	10	100
20	g	g	g	g	g	g	d	g	g	h	8	80
Rata-rata												91.5

The word of "tidur" test results in Table 8 show that from 20 experiments with each saying 10 times the word of "tidur" can be seen the average success of the system in recognizing the word of "tidur" and sending the 'g' character is 91.5%. The system that can't recognizes word "tidur" and displays characters instead of 'g' can be caused by noise, differences in the pronunciation of each person's words because the library uses English pronunciation, and the similarity of words so the system can recognize words other than the word of "tidur".

3.8. Testing the Word of "Makan"

Table 9. Testing the Word of "Makan"

Percobaan	Pengujian ke-										Total	%
	1	2	3	4	5	6	7	8	9	10		
1	h	h	h	h	h	h	h	h	h	h	10	100
2	a	h	h	h	h	h	h	h	h	h	9	90
3	h	h	h	h	h	h	h	h	h	h	10	100
4	h	h	h	b	h	h	h	b	h	h	8	80
5	h	h	h	h	h	h	h	h	h	h	10	100
6	h	b	h	h	h	h	h	h	h	h	9	90
7	h	h	h	h	h	h	h	h	h	h	10	100
8	h	h	h	h	h	h	h	h	h	h	10	100
9	h	h	h	h	h	h	h	h	h	h	10	100
10	h	h	h	h	h	h	h	h	h	h	10	100
11	h	h	h	h	h	h	h	h	h	h	10	100
12	h	h	h	h	h	h	h	h	h	h	10	100
13	h	h	h	h	h	h	g	h	h	h	9	90
14	h	h	h	h	h	g	h	h	h	h	9	90
15	h	h	h	h	b	h	c	h	h	h	8	80
16	h	h	h	b	h	h	h	h	b	h	8	80
17	h	h	h	h	h	h	h	h	h	h	10	100
18	h	a	h	h	h	h	h	h	h	h	9	90
19	h	h	h	h	h	h	h	h	h	h	10	100
20	h	h	h	h	h	h	h	h	b	h	9	90
Rata-rata												94

Based on the test of the word of “makan” in Table 9, from the 20 experiments with each saying 10 times the word of “makan” can be seen the average success of the system in recognizing words and sending the character 'h' is 94%. The system that can’t recognizes word “makan” and displays characters instead of 'h' can be caused by noise, differences in the pronunciation of each person's words because the library uses English pronunciation, and the similarity of words so that the system can recognize words other than the word of “makan”.

3.9. Testing the Word of “Toilet”

Table 10. Testing the Word of “Toilet”

Percobaan	Pengujian ke-										Total	%
	1	2	3	4	5	6	7	8	9	10		
1	i	i	i	i	i	i	i	i	i	i	10	100
2	d	i	i	i	i	i	i	i	i	i	9	90
3	b	i	i	i	i	i	i	i	i	i	9	90
4	i	i	i	i	i	i	i	i	i	i	10	100
5	i	i	i	i	i	i	i	i	i	c	9	90
6	i	d	i	i	i	i	i	i	i	c	8	80
7	i	i	i	i	i	i	i	i	i	i	10	100
8	i	b	i	i	i	d	i	i	i	i	8	80
9	i	i	i	i	i	i	i	i	i	i	10	100
10	i	i	i	i	i	i	i	i	i	i	10	100
11	i	i	b	i	i	i	i	i	i	i	9	90
12	i	i	i	i	c	i	i	b	i	i	8	80
13	i	i	h	i	i	i	i	c	i	i	8	80
14	i	i	i	i	i	i	i	i	i	i	10	100
15	i	i	i	i	i	i	i	b	b	i	8	80
16	i	i	b	i	i	i	i	b	i	i	8	80
17	i	i	i	i	i	i	i	b	i	i	9	90
18	i	i	i	i	i	i	i	i	i	c	9	90
19	i	i	i	i	i	c	i	c	i	i	8	80
20	i	i	i	i	i	i	b	i	h	i	8	80
Rata-rata												89

Based on Table 10, from 20 experiments with each saying 10 times the word of “toilet” can be seen the average success of the system in recognizing the word of “toilet” and sending the character 'i' is 89%. The system that recognizes non-the word of “toilet” and displays characters instead of 'i' can be caused by noise, differences in the pronunciation of each person's words because the library used uses English pronunciation, and similarity of words so the system can recognize words other than the word of “toilet”.

3.10. Overall Testing Tool

Testing the entire tool is done by testing the system recognize the spoken word and send the character assosiate with the recognized word to the microcontroller to command the movement of the wheelchair. Tests carried out by a person sitting on a wheelchair say 5 times the word of “toilet”. The results of overall testing of the tool can be shown in Table 10.

Table 11. Overall Testing Tool

Testing	The spoken word	Succeed / Fail
1	toilet	succeed
2	toilet	succeed
3	toilet	succeed
4	toilet	succeed
5	toilet	succeed

Based on the results of the overall testing of the tools in Table 10, it can be seen that from 5 tests of uttering the word of “toilet”, the system can recognize the spoken word and successfully send word recognition results to the microcontroller. Microcontroller which has received data in the form of characters will instruct the actuator to drive the Direct Current (DC) motor . Thus the wheelchair can move according to the trajectory to the toilet with the speed that has been adjusted.

IV. Conclusion

The results of the word of “maju” testing show the average success of the system in recognizing the word of “maju” and sending the 'a' character is 97%. The average success of the system in recognizing the word of “mundur” and sending the 'b' character is 88.5%. The average percentage of success of the system in recognizing the word of “kanan” and sending the 'c' character is 94%. In the word of “kiri” test, the average system success in recognizing the word of “kiri”and sending the 'd' character is 92%. The average success of the system test results in recognizing the word of “stop” and sending the 'e' character is 98.5%. In the word of “tamu” testing, the average success of the system in recognizing the word of “tamu” and sending 'f' characters is 97.5%. The word of “tidur” test results show that the average success of the system in recognizing the word of “tidur” and sending the 'g' character is 91.5%. The average success of the system in recognizing the word of “makan” and sending the 'h' character is 94%. The average

percentage of success of the system in recognizing the word of “toilet” and sending the character 'i' is 89%. In overall testing the tool, 5 times the test uttered the word of “toilet”, the system can recognize the word of “toilet” and successfully send word recognition results to the microcontroller. Microcontroller which has received data in the form of characters will instruct the actuator to drive the DC motor. Thus the wheelchair can move according to the trajectory to the toilet with the speed that has been adjusted.

References

[1]. I. M. L. Batan, “Pengembangan Kursi Roda Sebagai Upaya Peningkatan Ruang Gerak Penderita Cacat Kaki”, Jurnal Teknik Industri, Vol. 8, No. 2, Desember 2006: 97-105.

[2]. Kemenkes. 2014. “Infodatin Penyandang Disabilitas Pada Anak”. Jakarta: Kemenkes.

[3]. Liem, Yuliana Kathina Hatta, Pujiono, dan Tasrippan, “Rancang Bangun Kursi Roda Elektrik Menggunakan Perintah Suara Berbasis Aplikasi Android”, JURNAL TEKNIK POMITS,Vol. 1, No. 1, 2012, 1-6.

[4]. A. K. Ridia, A. Hidayat, Derisma, “Penerapan Metode Fuzzy Logic Pada Kursi Roda Elektrik Dengan Kendali Suara”, PROSIDING SEMNASTEK 2017, Pp. 1-8, 2017.

[5]. M. E. B. Prasetyo, “Teori Dasar Hidden Markov Model”, Makalah II2092 Probabilitas dan Statistik-Sem. I, 2010/2011.

[6]. Pocketsphinx, Carnegie Mellon University. [Online]. Available: http://cmusphinx.sourceforge.net/wiki/tutorialPocketSphinx [Accessed: 22-Agustus-2019].

[7]. M. W. Alauddin, W. Kurniawan, B. D. Setiawan, “Rancang Bangun Alat Pendeteksi Suara Panggilan Manusia Berbahasa Indonesia Untuk Tunarungu Menggunakan Library PocketSphinx Berbasis Embedded system”, Repositori Jurnal Mahasiswa PTIIK UB, Volume 8, Pages 16, 2016.

[8]. Raspberry Pi, “Raspberry Pi,” Raspberry Pi Foundation. [Online]. Available: https://www.raspberrypi.org [Accessed: 22-Agus