

Design of Application On/Off Electronic Device with Markov Model Using Speech Recognition on Android

Rochman Widiyatmoko^{*1}, Sugeng Purwanto E.S.G.S², Yoanda Alim Syahbana S³

^{1,2,3}Politeknik Caltex Riau

rochmanwidiyatmoko@gmail.com^{*1}, sugeng@pcr.ac.id², yoanda@pcr.ac.id³

Abstract

Electronic devices are supported by a switch that is used to turn the device on and off. Manually pressed switches with distances between remote switches to cause less efficiency in saving human time and manpower. This can be solved by building a system to control electronic devices automatically. The system uses human voice commands to turn on and off electronic devices. The command will be processed into text by the Google Voice Speech Recognition library. The Android app sends human commands that have been processed by Arduino Uno R3 microcontroller. Commands are obtained after the text and data in the database are processed using the Markov Model algorithm. Communication between Android smartphone and microcontroller will be designed through a WIFI network. This system is tested based on noise level with data accuracy level with noise 0-45 dB and obtained 65% result. Based on the test response time obtained that the noise level 0-45 dB obtained results of 5.41 seconds. Based on the test results from the scenario, it can be concluded that the lower the noise generated, the better the system will also respond to commands. From the test suitability get value $X = 1$, meaning that the system is suitability with error rate 0. In testing accuracy to view status function get value 0 with error level 0. Testing of Markov model algorithm yields the calculated 0.125 algorithms manually and code for each command.

Keywords: Speech Recognition, Arduino, Android, Markov Model, Wifi

1. Introduction

For this study, there are some previous research reviews. The first is Research Supriyanto (2013) with the title Design of Light Controller Using ATMEGA8538 Microcontroller based on Android Through Bluetooth and Speech Recognition. The research aimed at building a light control system using android based voice commands. Researchers use switch buttons and speech recognition to enter commands. Data sent from Android will be received by the existing bluetooth module on the microcontroller system. Data read by microcontroller then forwarded to relay, then relay will forward data used to turn on or off lamp [1].

The second study by Liuxinfei (2014) entitled A Chinese Small Vocabulary Offline Speech Recognition System Based on Pocketsphinx In Android Platform. The research aimed to build a vocabulary search system with Chinese language offline. The researchers used the PocketSphinx technique, a technique designed by Carnegie Mellon University (CMU) in the development of an offline vocabulary system with Speech Recognition. Voice data is processed using PocketSphinx technique so that the output will be text format. Inside PocketSphinx the user's voice is calculated with probability with N-gram which is useful to equalize the user's voice with the vocabulary in the database [2].

And in a third study by Hamdani (2015) entitled Smart Office Room Monitoring Based on Arduino and Speech Control. The research aimed to build an electronic device control system with voice command and SMS. Researchers build control applications using the VB.NET programming language. The VB.NET application receives user voice commands via the microphone. The user's voice is converted to text by speech recognition and matched to commands on the database. The computer sends commands to the Arduino and the command is executed by turning on and off the relay connected to the electronic device. Users can also perform remote device control by viewing status and turning off electronic devices via SMS [3].

Electronic devices are categorized based on their level of usefulness. Levels of utility such as in industry, military, medical, household, and other fields. Electronic devices that are often encountered at home fall into the category of home electronic devices (Home Appliance). One

example of home electronic devices is home lights, fans, televisions, and other electronic devices [4]. Electronic devices have a switch button installed on each device. The switch is used for controlling electronic devices such as "turning on" and "turning off" the device [5].

The electronic device can be "on" and "off" by pressing the manual switch. This method is considered inefficient in time and manpower because it requires a person must be near the electrical device to suppress the switch. In addition to pressing the switch, one must be near an electronic device to see electronic devices alive or dead. The device is controlled more than one, with the distance conditions of each equipment, not adjacent [6].

Control of electronic devices will be more practical if controlled through a voice-based electronic device control system. The control system provided is a human voice command to the computer via Microphone [3]. However, with a given solution does not support the mobility of the user because the computer is not always carried by humans. The solution offered to support user mobility is by replacing the computer with the Android-based smartphone. Android Smartphone is open source with a lot of users. This, according to a survey from International Data Corporation (IDC) which states that in 2015 Android smartphones dominate the market with a percentage of 82.8% [7].

This paper describes research on control systems to "turn off" and "turn on" electronic devices using human voice with Markov algorithms on Android Smartphones. Control system provided in the form of voice commands users to Android smartphones. The user's voice is processed using Speech Recognition. Speech Recognition will translate voice input into a text [8]. The text results from speech recognition will be processed with a Markov model algorithm to determine the probability. Commands received by the system will be processed and then sent to the Arduino microcontroller for execution. Microcontroller used is Arduino Uno R3. Arduino Uno allows the use of a shield. Wi-Fi network is used to connect the Android device with Arduino microcontroller.

In this paper consists of 4 parts. The first part explains what is behind the idea of this research, how the research position with existing research and evaluation technique formulation will be done. The second part describes the research method. The third section describes the results and discussion. Then the last section discusses the conclusions and discussion of research development.

2. Research Method

Electronic control system is made into two parts namely the Android and Microcontroller. Android application built based on speech recognition with minimal android version used is android 4.0 ice cream sandwich. Users can perform voice commands by opening the electronic controller application. Google voice will receive voice commands spoken by users and sent to google servers. In this step Android smartphone must be connected to the internet network. The google server will translate the sound into a string that is sent back to the app. Examples of commands to be processed are "turn on red lamp". Markov model is used to search probability value between text result from speech recognition with command data in database. If the markov model has a value greater than 0 then the control command will be sent to Arduino. From the command "turn on red lamp" will generate arduino command 131. Where "13" is the port number on arduino, while "1" is the status for command "on". Access point is used as a liaison between android smartphone with arduino microcontroller. The arduino microcontroller receives commands such as port numbers and status to turn on or turn off electronic devices. The received command example is status = 1 with port = 13 then the microcontroller will turn on port 13 to control the "red" led light. In Arduino microcontroller will read status received, if status = 1 then command to on while if status = 0 then command to off. The indicator light is mounted on the Arduino microcontroller digital port. There are two led lights as indicators that are "red" and "green". The red light is mounted on the digital port 13 while the green light is installed on the digital port 12. The led light is selected because it requires a smaller electric current. The led lamp requires a maximum current of 30 mA, while the Arduino Uno R3 can provide a maximum current of 40 mA on the I / O pin. So that led lights suitable to use without adding current from outside Arduino (See Figure 1).

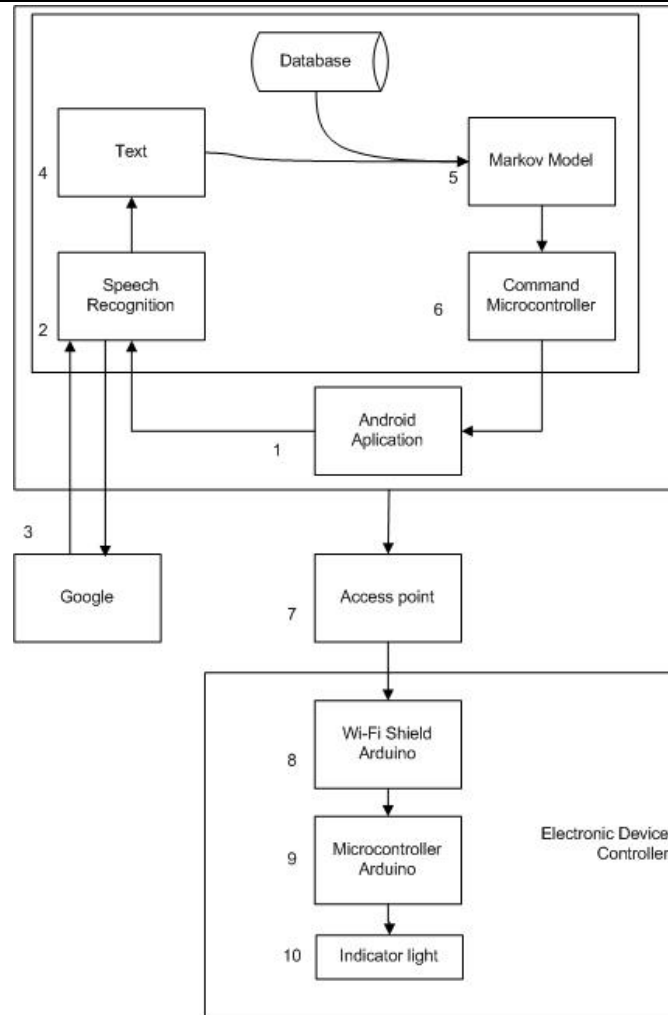


Figure 1. Block Diagram of On/Off Control System

In this study, it is divided into 3 main stages. First, build a control system on/off. This system consists of android application with Arduino microcontroller. The system process flow as described in the previous paragraph.

Second, the testing phase of the on/off control system. At this stage, the test is performed functionally, in accordance with the existing function on the control system on / off. Nested functions are functions of the human voice command (considering the noise and timing levels), electronic device status display function, and Markov model algorithm testing.

Third, perform analysis of test results on/off control system. The method used is the calculation according to the external metric formula ISO / IEC 9126-2 [9]. Where the selected metric is the one that corresponds to the Analysis phase, as well as the input on the metric, is the result of the Testing phase. Markov algorithm model analysis is performed to determine the level of algorithmic function on the on / off control system with voice recognition. From this result will be obtained the conclusion objectively to the control system on / off with voice recognition

3. Results and Discussion

The on / off control system using speech recognition produces two views, namely: application display and hardware display.

3.1 System Interface

The first system view is the android app display. On the main page of the application, there is an input IP Address and port used to access the server on the microcontroller. The microphone logo button is used to enable the speech recognition feature as shown in Figure 2.

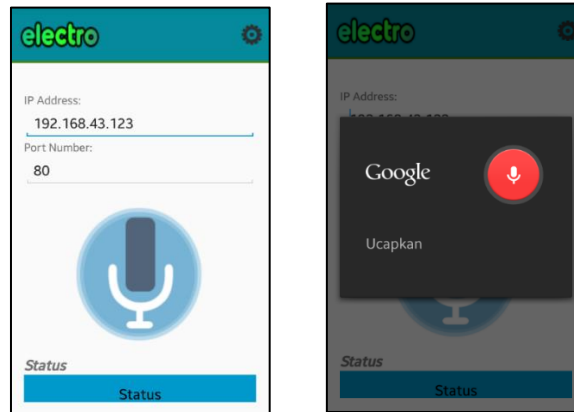


Figure 2. Main Page Interface and Command Input Page

The second system interface is the hardware interface. The hardware circuit in this system consists of a microcontroller, wifi module, and led lamp as an indicator. The entire circuit used can be seen in Figure 3.

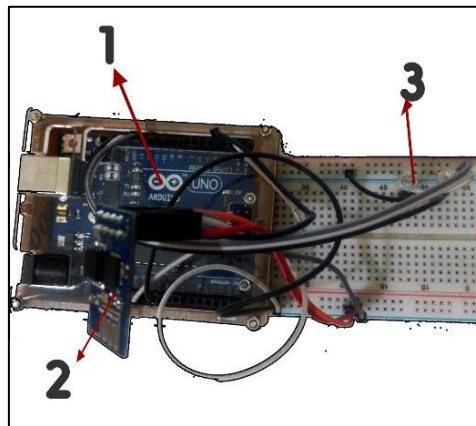


Figure 3. Hardware Device On / Off Control System

Details of the hardware sequence used are as follows:

1. Arduino Uno R3
2. Wifi module
3. led light (indicator)

There is an Arduino microcontroller that acts as a controller of the system that has been created. The wifi module serves as a link between Arduino and Android app. The led light as an indicator is an electronic device controlled by a microcontroller.

Here is a Markov Model process that works on a built-in system that the received user's voice is processed by a speech recognition library with text output. The text will be processed using the Markov Model method with the output of the command to be sent to the Arduino details can be explained as follows: "Kata" is a text variable resulting from speech recognition which is used as a parameter for the algorithm markov model. The number of repetitions corresponds to the number of characters in "Kata". Data "command" in the database stored in the array to facilitate the process of calculating the probability. First check whether the nth "command" character is not the same as "null". If eligible then the next check between the character "Kata" (text results speech recogniton) to-n with the character "command" to -n. If the result is the same then the temPro variable plus one. If the result is not the same then the nth command data is made into "null". Furthermore, the index variable is added with one each time iteration. Repeated as much as "command" data stored in the array. Then calculated probability value with the Equation 1.

$$Probability = \left(\frac{temPro}{index}\right) \times probability \tag{1}$$

Having obtained a temporary probability value, then check whether the loop is not greater than the number of "Kata" characters. If it is smaller than the number of "Kata" characters then the process of searching the temporary probability value like the process. If the value of the recurrence is equal to the number of "Kata" characters then checks whether the probability value is greater than 0. If it is larger then the result is the arduino port and the status corresponds to the "command" data stored in the database. Example command is "turn on red lamp" with probability value 0.125, then the result of markov is port 13 and status = 1. The command sent to arduino is 131. However, if the probability value is equal to 0, then the result " found ".

3.2 Testing and Analysis

3.2.1 Noise Level

The noise testing on this application will be done in three different environmental conditions ie the noise environment is given a range between 0 - 45 dB, for which noise is given a value between 46 - 60 dB, and any noise is given value 60 - 100 dB [10]. In every environmental condition, there will be eight electronic device control commands with ten iterations.

The result of noise level testing, the lower the noise level, the higher the data accuracy. Vice versa, the greater the noise level (noise) then the level of data accuracy is getting smaller. This can be seen from Figure 4. Based on the results obtained analysis, then the application should be used in environmental conditions with noise level (noise) between 0-60 db.

In addition to noise level conditions (noise), the use of the word as a command will affect the accuracy of the data. The test results that on the word turn on the red lamp, turn off the red lamp, turn on the green lamp, and turn off green lamp has a low accuracy value. This is due to the use of the word lamp at the end of the command, where the application cannot translate the user's pronunciation because the user's dialect cannot be understood by the system. Therefore, to make the value of the accuracy of the word to be higher should not use the word lamp.

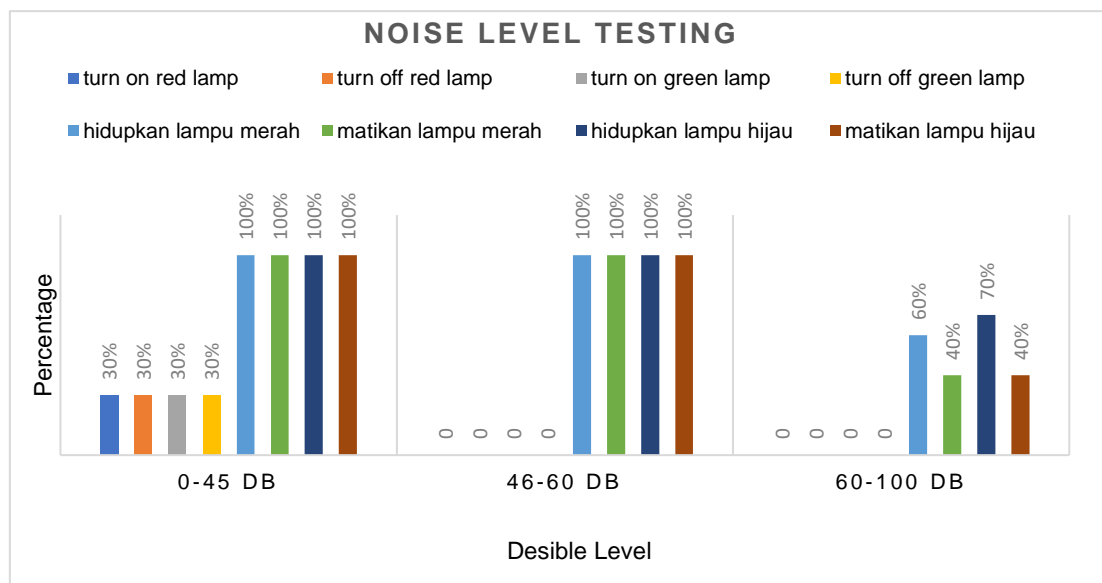


Figure 4. Noise Level Testing

3.2.2 Response Time

In the test response time by using voice commands to the application. Calculates the time required for an electronic device to respond to a command after a command is performed using a stopwatch. Environmental conditions are based on noise testing. The average response time can be seen in Table 1.

Based on Table 1, it can be concluded that the lower the noise level than the time required to process data faster, the higher the noise the longer the time required. The noise level greatly affects the system's response time in receiving user voice commands and executing commands.

Apart from the noise level the network speed and command pronunciation power also affects the time it takes the system to execute commands. The slower the network then the time it takes to translate the user's utterance into text is also longer. This is because Google Voice library directly communicates with Google servers online.

Table 1. Average Response Time

No	Commands	Average Time (Second)		
		0-45 dB	46-60 dB	60-100 dB
1	"turn on red lamp"	5,77	Failed	Failed
2	"turn off red lamp"	6,17	Failed	Failed
3	"turn on green lamp"	5,47	Failed	Failed
4	"turn off green lamp"	5,43	Failed	Failed
5	"hidupkan lampu merah"	4,92	8,73	10,1
6	"matikan lampu merah"	5,08	7,9	10,5
7	"hidupkan lampu hijau"	5,3	7,12	10,1
8	"matikan lampu hijau"	5,12	9,07	10,6
Rata-rata waktu		5,41	8,21	10,31

3.2.3 Suitability Testing

Testability of suitability based on functional adequacy metrics in accordance with ISO 9126-2 [7]. There were eight repeating sounds repeated ten times in control of two objects. The results of suitability testing can be seen in Table 2. The method of measurement using this metric is Equation 2.

$$X = 1 - \frac{\text{Number of system errors}}{\text{Total system testing}} \quad (2)$$

Table 2. The result of Suitability Testing

No	Commands	Total Testing	Total Errors
1	"turn on red lamp"	10	0
2	"turn off red lamp"	10	0
3	"turn on green lamp"	10	0
4	"turn off green lamp"	10	0
5	"hidupkan lampu merah"	10	0
6	"matikan lampu merah"	10	0
7	"hidupkan lampu hijau"	10	0
8	"matikan lampu hijau"	10	0
Total		80	0

The test was conducted 80 times with 10 tests for each command. The result of X value with value 1 with an error level of 0. This is because the system can translate the user's utterance in accordance with the command that has been stored in the database and the system successfully send a command to Arduino.

3.2.4 Accuracy Testing

The accuracy test on this app is used to test the status of all electronic devices. Testing by sending status requests as many as ten requests. The results provided by the system will be compared with the real state of the electronic device. Testing based on accuracy expectation metrics in accordance with ISO 9126-2 [7]. The measurement method using this metric is Equation 3.

$$AE = \frac{\text{Number of functions different from the reasonable result (error rate)}}{\text{Total function testing}} \quad (3)$$

The condition $0 \leq X \leq 1$, where the value close to 0 means the more consistent the function is implemented. Thus, from the above test, it can be concluded that the function to see the status of both accuracies with the focus accuracy expectation metrics.

3.2.5 Markov Model Testing

This testing is by comparing the probability value obtained by manual and by way of coding. This test is done with the aim of equating the concept of manual with coding. Markov Model is a mathematical technique used to perform modeling of various systems and business processes [11]. The Markov chain does not provide recommendation decisions, but only the probability information about decision situations that can aid decision making. Thus, the Markov chain is not an optimization technique, but it is a descriptive technique that produces probability information in the future [12]. Results of probability testing as in Table 3. Calculation of Markov probability model using the following Equation 4.

$$\begin{aligned} P(X_{n+1} = j | X_1 = x_1, X_2 = x_2, X_3 = x_3, \dots, X_n = i) = \\ P(X_{n+1} = j | X_n = i) = P_{ij} \end{aligned} \quad (4)$$

Table 3. Probability Calculation Results

No	Commands	Probabilitas Manual	Probabilitas Code
1	"turn on red lamp"	0,125	0,125
2	"turn off red lamp"	0,125	0,125
3	"turn on green lamp"	0,125	0,125
4	"turn off green lamp"	0,125	0,125
5	"hidupkan lampu merah"	0,125	0,125
6	"matikan lampu merah"	0,125	0,125
7	"hidupkan lampu hijau"	0,125	0,125
8	"matikan lampu hijau"	0,125	0,125
Rata-rata keakuratan		0,125	0,125

The next test compares the system without the algorithm model Markov by using the algorithm. The result of the analysis states that if the command is given in accordance with the existing database then the system will send commands to Arduino. Where the intermediate results using Markov algorithm with without using Markov algorithm is the same. However, if an incomplete command between using the Markov algorithm without using the Markov algorithm is different. This is because the command is truncated (incomplete) so the system without using Markov algorithm cannot find the appropriate database. However, if using Markov algorithm the system can recognize commands as seen from the probability value. Use of the Markov Model in this Final Project as a solution to resolve issues on improper user pronunciation, and pronunciation of truncated commands.

4. Conclusion

Based on the results of testing and analysis conducted it can be concluded that speech recognition technology that is connected to the android application runs well. The average accuracy of converting voice to text reaches 65% at 0 – 65 dB noise level.

Applications can respond to commands quickly. The average time required by the system to respond to orders ± 5.41 seconds with noise levels 0-45 dB and the network in state stable. Applications can run well according to suitability with the focus of functional adequacy metrics. Testability suitability based on ISO 9126-2, with the result value 1 with error rate 0. The function of viewing the status of the application expressed accurately with accuracy expectation metric focus. Testing accuracy based on ISO 9126-2.

Markov method The model runs well based on the probability calculation between manual and code with 100% equality level. Furthermore, the system will run well if using the algorithm Markov model. Because the system can recognize the utterance of user commands even if the command is cut or incomplete.

Although this study shows the application can run well, there are still limitations on the system. Speech recognition technology must be connected to the internet network. So that

research can be developed by using speech recognition technology that can run offline. Control of the system is still limited to turn on and off electronic devices. So research can be developed by controlling the length of time the electronic is turned on and so forth.

References

- [1] A. Supriyanto, "Rancang Bangun Pengendali Lampu Menggunakan Mikrokontroler Atmega8538 Berbasis Android Melalui Bluetooth dan Speech Recognition," Sekolah Tinggi Manajemen Informatika dan Komputer Amikom Yogyakarta, 2013.
- [2] X. Liu, "A Chinese Small Vocabulary Offline Speech Recognition System Based on Pocketsphinx in Android Platform," Switzerland: Trans Tech Publication, 2014.
- [3] R. Hamdani, "Smart Office Room Monitoring Berbasis Arduino dan Speech Control," Politeknik Caltex Riau, 2015.
- [4] A. Pitoyo, "Cara Cepat Memilih Barang Elektronik Rumah Tangga." Solo, 2005.
- [5] H. Ridwanda, "Sistem Kendali Alat Listrik Berbasis Waktu dengan Atmega8535," Universitas Tanjung Pura, 2014.
- [6] N. Sutarmanto, "Sistem Kendali Perangkat Listrik Menggunakan Media Sms (Short Message Service)," 2007.
- [7] D. Ardiansyah, *Arduino Advance – Modul Training Arduino*. Jakarta, 2015.
- [8] R. B. Abuhari, Nurjannah, And M. A. Hairuddin, "Speech Recognition." Makasar, 2014.
- [9] I. Lestari, "Evaluasi Fungsionalitas Learning Management System Berdasarkan Iso/lec 9126-2," *Jurnal Sains, Teknologi dan Industri*, Vol. 13, No. 1, Pp. 123–129, 2015.
- [10] A. Justian, "Analisis Pengaruh Kebisingan Terhadap Performa Siswa Sekolah Dasar di Ruang Kelas," Universitas Indonesia, 2012.
- [11] V. Monika, "Analisis Pengaruh Kebisingan Terhadap Performa Siswa Sekolah Dasar di Ruang Kelas," Binus University, 2012.
- [12] Y. Eni, *Markov Analysis*. Jakarta, 2015.