Voice-Based Monitoring And Control System of Electronic Appliance Using Dialog Flow API Via Google Assistant

Mokh. Sholihul Hadi
Dept. of Electrical Engineering
Faculty of Engineering,
Universitas Negeri Malang
Malang, Indonesia
mokh.sholihul.ft@um.ac.id

Maulana Ahmad As Shidiqi
Dept. of Electrical Engineering
Faculty of Engineering,
Universitas Negeri Malang
Malang, Indonesia
maulana.ahmad.1505366@students.um.ac.id

Ilham Ari Elbaith Zaeni
Dept. of Electrical Engineering
Faculty of Engineering,
Universitas Negeri Malang
Malang, Indonesia
ilham.ari.ft@um.ac.id

Muhammad Alfian Mizar
Dept. of Mechanical Engineering,
Faculty of Engineering,
Universitas Negeri Malang
Malang, Indonesia
alfianmizar@um.ac.id

Mhd Irvan
Graduate School of Information Science and Technology
The University of Tokyo
Tokyo, Japan
irvan@yamagula.ic.i.u-tokyo.ac.jp

***Abstract*—The development of the control system, such as monitoring and control system using voice command, in the electronic device recently increased. One of the biggest and most comprehensive interface platforms was Google Assistant. There was a great opportunity in developing Google Assistant application due to its high support, its minimum size, and the minimal Google-based applications. The Dialog flow API (Application Programming Interface), an API that functioned to process humans' conversation based on artificial intelligent, fully support this possibility. Checking and control appliance through voice required a cell phone to lead the communication. The cell phone conveyed the sound to Google Assistant server through the web. Voice change into content happened in the server at that point proceeded into the Dialog flow API. This API prepared the content as per the discussion characterization or plan. The test outcomes demonstrating that creating electronic gadget application on checking and control dependent on voice had incredible potential. The accomplishment pace of talk affirmation structures is 75% of success rate.**

Keywords— Control, Voice Command, Dialog Flow API, Microcontroller

I. INTRODUCTION

The development of human and machine interaction within the last one to two decades had become more diverse. It started with the massive distribution of internet in the 2000s, the spread of desktop computer in the society, and up to the utilization of handphone that reached the middle-lower groups. Gradually, the usage of technology devices such as desktop computer and handphone became a vital need for the public. Apart from the affordable price, innovations in the tech industry are competing to create profitable products to solve the community's problems in all classes. One field that still widely unexplored and required development is the interaction of human and machine through voice communication. This causes the interface with voice can be the future age of the interfacing technology after the touch screen-based [1][2] Google Assistant is one of many interface providers that uses voice recognition. Moreover, there are additionally Siri by Apple, Amazon Alexa, Cortana from Microsoft and Alice from Yandex [3] [4].

Since the interface is voice based, the outcome must resemble people conversing with individual people. Subsequently, an incorporated framework component was needed. For example, Automated Speech Recognition (ASR), Spoken Language Understanding (SLU) or now and again called Natural Language Processing (NLP), Dialog Manager (DM), Knowledge Database (KDb), Natural Language Generator (NLG), and Text-to-Speech combination (TTS) [5] [6].

Google Assistant has an installed arrangement of ASR and TTS. other than both, third party application designers utilize their own projects or additionally utilize the Dialog flow API [7]. Based on Natural Language Processing (NLP), developers do not have to think of a way to convert human language into information because Google Assistant has API (Application Programming Interface) that can transform it automatically or customized by developers and strongly supports Google Assistant SDK[8]. This advantage helps developers to focus more on the app developing process than language conversing process. Developers also have the edge in storage-saving because users do not have to download an additional application. Related research, Google Assistant is used to control the position of robots such as forward, backward, right, or left orders [9]. Also, there is the use of voice-based virtual assistants to control IoT devices that are developed individually that are not integrated with Google Assistant [10] or only use the Google Speech Recognition API and have their own NLP [11] [12].

To date, not only handphone that spreads in the society but also microcontroller developers that compete to present a cheap and powerful device, such as ESP-32, a chip microcontroller from Espressif System. This microcontroller integrates with Wi-Fi and Bluetooth connection [13][14]. The extensive support from the ESP-32 developer also benefits the developer.

The above advantages make it possible to create a user interaction through voice to control or monitor the electronic device using a microcontroller without having to think about Natural Language Processing (NLP), what just the designer needs to do is to give enough data expected to

978-1-7281-4160-2/19/$31.00 ©2019 IEEE

prepare the framework to perceive what was requested, and give a case of the reaction consequently [15].

II. METHOD

Fig 1, shows the workflow on the end user device side. When the user says a command, the Google Assistant app will send an audio file to the Google Assistant server (aka Action-on-Google) to process and communicate with third-party applications. All processes are in the cloud, the application on the user's device is just an interface.

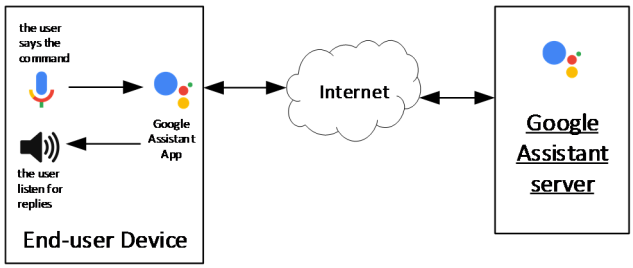


Fig. 1. block diagram on the end-user side

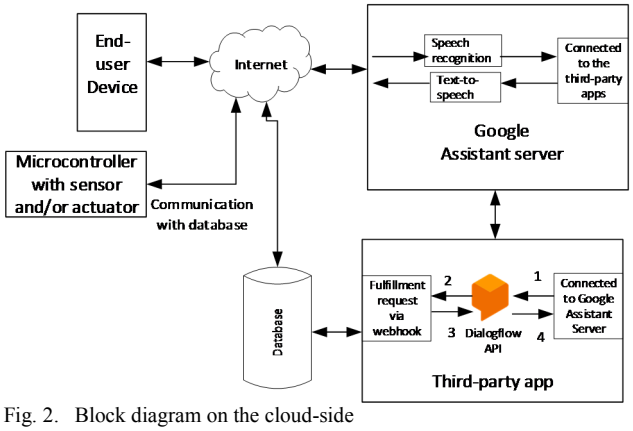


Fig. 2. Block diagram on the cloud-side

In the Google segment there are a few procedures that must be finished in light of the fact that all procedures are done in the cloud. the principal result the server does is that the server gets a sound document from the client (fig 2). Next is the procedure of discourse acknowledgment that is changed over to content and after that the content is sent to an outsider engineer application. in outsider engineer applications there is joining with the API dialog flow which is utilized to process client input discourse. In procedure number 1 in figure 2, the stream that happens is that Dialog flow gets content from the Google Assistant Server. After that the Dialog flow API groups client contribution as per the 'goal' set by the application engineer. If necessary, the following procedure is the satisfaction procedure to synchronize with the database that is spanned by the webhook (number 2, fig 2). From that point forward, Dialog flow will get a reaction from the database (number 3, fig 2) and give a book reaction to be sent to the Google Assistant server with the goal that it turns into a voice (number 4 fig 2). After the content to-discourse process on the Google Assistant server, the client will get a sound document to tune in to in light of the voice direction.

The diagram, as shown in Fig 3, is an example of how Dialog flow and Google Assistant perform, such as communicating with database and hardware, as shown implemented in an aquaponic control system. Connecting the Google Assistant server with microcontroller required database as the bridge between both parties.

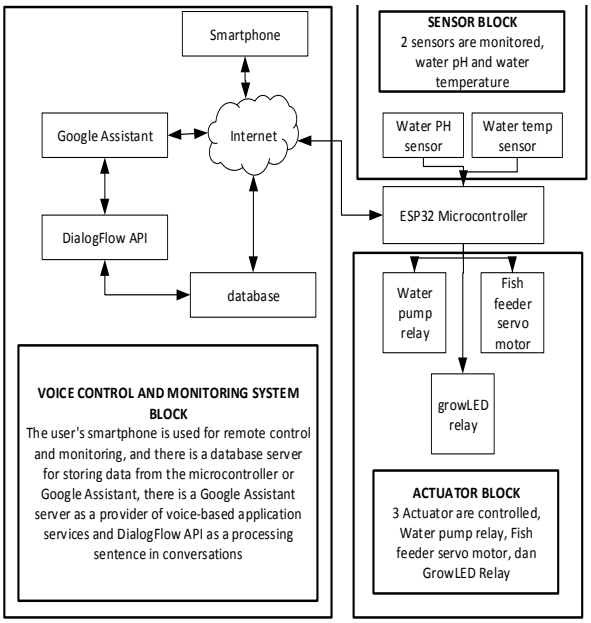


Fig. 3. Block diagram of the voice-controlled aquaponic system

TABLE I. FUNCTION IN INTENT

Function	Information
Training phrase	It is a collection of examples that might be said by the user to triggering an intent. sometimes, this feature is optional. although optional, it is recommended to use this function so that the system can distinguish between intents
Action and parameter (optional)	The action functioned to classify the task in the developer's screen. The parameter functioned to extract data in a conversation.
Response phrase (optional)	As a response from the intended response. If there is more than one response, the system will choose one randomly.
fulfillment	Bridging intent with webhook

Monitoring and control system via voice needed a smartphone to conduct the interaction. The smartphone delivered the sound to Google Assistant server through the internet. Voice conversion into text occurred in the server then continued into the Dialog flow API. This API processed the text according to the conversation classification or intent. When required, webhook could take over the communication according to the database; then webhook delivered the result, which contained the measured

response, to the Dialog flow and into the Google Assistant server. If it were not required, Dialog flow would give the reaction according to intent to the server, and then the server sent the voice data to the user.

There were four intent tests as responses from the inputs. Table 1 explain the function of each setting on the intent.

III. Results and Discussion

The test aimed to provide information from a pre-designed system so that the ability of the success of this system was known. This test used the system to condition the aquaponic.

Tables 2 through 5 exhibit the response of the system if given a voice command. the data presented is only in the form of data communication that occurs between the smartphone and the server using the Dialog flow API. The system will answer using words that have been entered into the database for each condition of a different command sentence. To use voice command system for controlling electronic appliance other than aquaponic system, other words that are more suitable can be applied, depending on the appliance. Because Dialog flow API based on machine learning, in this test we set 0.3 (default value) at the threshold. if the result of matching is below 0.3, it will be no response

TABLE II. Intent and Default Welcome Intent Setting

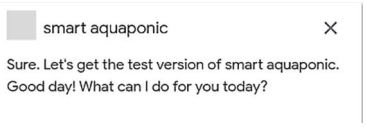
Intent name	Default Welcome Intent
Intent description	The sentence used to trigger the apps by saying "talk to smart aquaponic." In Google Assistant App
Training phrase	-
Response phrase	- Hello, how can I help? - Good day! What can I do for you today? - Greetings! How can I assist?
Fulfillment	No
Response	

TABLE III. Request pH Intent Setting

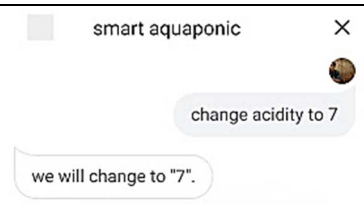
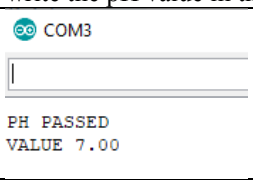
Intent name	Request pH
Intent description	Send order to change pH setpoint in the system
Training phrase	- Change the acidity to 7 - Change the pH to 7 - I want to change the pH - Change pH please to 6
Response phrase	- We will change to "$requestval"
Fulfillment	Yes
Response	
Database response	
	write the pH value in the database
Microcontroller response	

TABLE IV. Ask Temp Intent

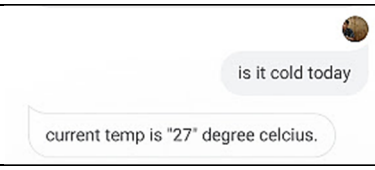
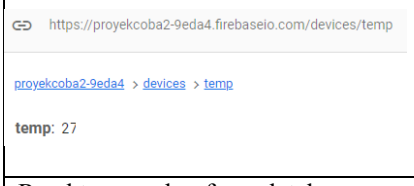
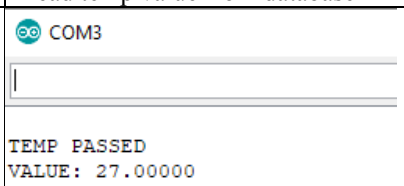
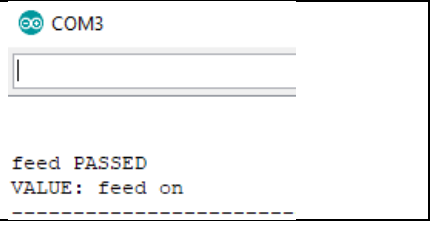
Intent	Ask Temp
Intent Description	Send order to take temperature data
Training phrase	- How about the temp? - Is it cold today? - What's the temperature now?
Response phrase	- Current temp is "${temp}" degree Celsius
Fulfillment	Yes
Response	
Database response	
	Read temp value from database
Microcontroller response	

TABLE V. INTENT REQUEST FEED SETTING

Intent name	Request feed
Intent description	Send order to feed the fish
Training phrase	- Are you hungry? - Time to eat - Feeding frenzy
Response phrase	- Time for a nice meal
Fulfillment	Yes
Response	
Database response	 write the feed value in the database
Microcontroller Response	

Apart from intent tests and responses, there were data on speech to text results as shown in Table 6.

TABLE VI. SUCCESS RATE TEST

No	Speech	Intent target	False	Correct	Success
1	Talk to smart aquaponic	Default welcome intent	4	6	60%
2	Is it cold today	Ask Temp	1	9	90%
3	Change acidity to 7	Request pH	4	6	60%
4	Feeding frenzy	Request feed	1	9	90%
	Success rate				75%

In light of the test results as appeared in table 4, the framework's reaction was tested by saying the words as appeared in the speech section. In the event that the command is correct, the application will execute the expectation that matches the command. then again on the off chance that it isn't right, at that point the application doesn't give any reaction. The achievement pace of discourse acknowledgment frameworks is 75% all things considered.

The test results indicating that developing electronic device application on monitoring and control based on voice had great potential. However, some tests acquired several false detections during speech recognition that made the app unable to recognize or to detect malicious intent. The cause of this problem was because the speaker was not a native English speaker. An error in speech can be lessened by slowly deliver the command or correcting the pronunciation.

IV. CONCLUSIONS

Generally, utilization of monitoring and control electronic device using voice command required more development because of a big gap between conventional and voice-based applications. The voice-based application gave a seamless experience to the users because they no longer needed to move to other applications to interact. They did not need additional memory because the application was inside the Google Assistant server. Thus, it was useful for the low-specification smartphone user.

ACKNOWLEDGMENT

This research is funded by PTUPT research fund from Ministry of Research and Higher Education. And we would like to thanks the university of Tokyo for their kinds support on this research.

REFERENCES

[1] Weigelt and W. F. Tichy, "Poster: ProNat: An Agent-Based System Design for Programming in Spoken Natural Language," 2015 IEEE/ACM 37th IEEE International Conference on Software Engineering, Florence, 2015, pp. 819-820.

[2] E. Kurniawati, L. Celetto, N. Capovilla and S. George, "Personalized voice command systems in multi modal user interface," 2012 IEEE International Conference on Emerging Signal Processing Applications, Las Vegas, NV, 2012, pp. 45-47.

[3] E. V. Polyakov, M. S. Mazhanov, A. Y. Rolich, L. S. Voskov, M. V. Kachalova and S. V. Polyakov, "Investigation and development of the intelligent voice assistant for the Internet of Things using machine learning," 2018 Moscow Workshop on Electronic and Networking Technologies (MWENT), Moscow, 2018, pp. 1-5.

[4] A. Reis et al., "Using intelligent personal assistants to assist the elderlies An evaluation of Amazon Alexa, Google Assistant, Microsoft Cortana, and Apple Siri," 2018 2nd International Conference on Technology and Innovation in Sports, Health and Wellbeing (TISHW), Thessaloniki, 2018, pp. 1-5.

[5] S. Mischie, L. Mâțiu-Iovan and G. GăŞpărescu, "Implementation of Google Assistant on Rasberry Pi," 2018 International Symposium on Electronics and Telecommunications (ISETC), Timisoara, 2018, pp. 1-4.

[6] R. Posevkin and I. Bessmertny, "Translation of natural language queries to structured data sources," 2015 9th International Conference on Application of Information and Communication Technologies (AICT), Rostov on Don, 2015, pp. 57-59.

[7] R. Milton, D. Hay, S. Gray, B. Buyuklieva and A. Hudson-Smith, "Smart IoT and Soft AI," Living in the Internet of Things: Cybersecurity of the IoT - 2018, London, 2018, pp. 1-6.

[8] D. Sheppard, N. Felker and J. Schmalzel, "Development of Voice Commands in Digital Signage for Improved Indoor Navigation Using Google Assistant SDK," 2019 IEEE Sensors Applications Symposium (SAS), Sophia Antipolis, France, 2019, pp. 1-5.

[9] P. Vashistha, J. P. Singh, P. Jain and J. Kumar, "Raspberry Pi based voice-operated personal assistant (Neobot)," 2019 3rd International conference on Electronics, Communication and Aerospace Technology (ICECA), Coimbatore, India, 2019, pp. 974-978.

[10] X. Han and M. A. Rashid, "Gesture and voice control of Internet of Things," 2016 IEEE 11th Conference on Industrial Electronics and Applications (ICIEA), Hefei, 2016, pp. 1791-1795.

[11] S. Khattar, A. Sachdeva, R. Kumar and R. Gupta, "Smart Home With Virtual Assistant Using Raspberry Pi," 2019 9th International Conference on Cloud Computing, Data Science & Engineering (Confluence), Noida, India, 2019, pp. 576-579.

[12] N. Cetic, M. Popovic, S. Furman and M. Krunic, "Cloud based voice interface for Internet of Things devices," 2015 IEEE 1st International

Workshop on Consumer Electronics (CE WS), Novi Sad, 2015, pp. 19-22.

[13] B. Nimmaka and A. Prakasarao, "Implementation of Indoor Pollution Monitoring Using MQTT Protocol," 2018 Second International Conference on Inventive Communication and Computational Technologies (ICICCT), Coimbatore, 2018, pp. 1316-1319.

[14] A. Maier, A. Sharp and Y. Vagapov, "Comparative analysis and practical implementation of the ESP32 microcontroller module for the internet of things," 2017 Internet Technologies and Applications (ITA), Wrexham, 2017, pp. 143-148.

[15] S. Mohapatra, N. Shukla, S. Jain and S. Chachra, "Nsmav-Bot: Intelligent Dual Language Tutor System," 2018 Fourth International Conference on Computing Communication Control and Automation (ICCUBEA), Pune, India, 2018, pp. 1-5.