JURNAL INFOTEL
Informatics - Telecommunication - Electronics
Website: http://ejournal.st3telkom.ac.id/index.php/infotel
ISSN: 2085-3688; e-ISSN: 2460-0997

Concurrently wireless sensor network using microcontroller for home monitoring against fire

Putu Wiweka Prasetyananda[1], Sunny Arief Sudiro[2,*], Bheta Agus Wardijono[3]
[1]Magister Teknik Elektro, Universitas Gunadarma
[2,3]STMIK Jakarta STI&K
[1]Jl. Margonda Raya 100, Depok 16424, Indonesia
[2,3]Jl. BRI, No. 17, Jakarta 12140, Indonesia
[*]Corresponding email: sasudiro@staff.jak-stik.ac.id

Received 1 September 2022, Revised 14 October 2022, Accepted 24 November 2022

Abstract — This paper proposes to make a prototype that can detect fires automatically and can be monitored via a website with a Wireless Sensor Network (WSN) to detect fires in several places at once and facilitate the placement of the detectors. WSN is a technology in which multiple sensors work together to establish communication over a wireless network. This prototype fire monitoring system can be monitored through the website in real time and the detection data is stored on the cloud server. This prototype fire monitoring system uses 4 sensor nodes, each of which has 1 KY-026 fire sensor and 1 MQ2 smoke sensor and uses the NodeMCU ESP8266 as a microcontroller and can be placed in several places to detect fires and send detection results to the master node wirelessly. The master node processes data concurrently, that is sent by the 4 sensor nodes into 16 conditions, in the event of a fire it will send a telegram message and condition data to the cloud server. Several attempts to transmit data from the sensor node to the master node server were completely successful and required an average delivery time of 890 milliseconds. Sending data from the master node detection results to the cloud server requires an average data delivery time of 345 milliseconds. When a fire occurs, the master node sends a telegram notification message with an average message delivery time to the homeowner of 972 milliseconds. Website monitoring can be accessed from anywhere and with any device that supports internet and web browser usage.

Keywords – NodeMCU ESP8266, ESPNOW, KY-026, MQ2, wireless sensor network

I. Introduction

Wireless Sensor Network or WSN is a technology where several sensors work together to establish communication over a wireless network using transceiver and receiver modules [1]. This network can be formed by many sensor nodes. WSNs can be used for monitoring or controlling environmental areas. Fire is a problem that occurs suddenly so it will be very difficult to predict and overcome. The causes of fires are varied and generally occur due to electrical short circuits, negligence due to forgetting to turn off the stove while cooking, gas leaks, and many other causes. Fire is of course very detrimental and can cause loss, both material, and life. The absence of a system that can detect fires in people's homes causes fire handling to be delayed; as a result, it can cause fires to spread everywhere. With this problem, it is necessary to have a system that can detect fires in the house automatically. WSN can be used as a fire detection system.

Several previous studies on fire detection, and research on system and circuit design for a fire early warning devices using heat sensors (LM35), gas sensors (QM-6), and smoke sensors (TGS 2600). If there is a fire will send a warning via Short Message Service (SMS) [2]. In this study there is a problem because it cannot monitor fires directly, only sending short messages when a fire occurs. Research on the design and manufacture of a fire early warning device with a video camera and smoke sensor that is connected to a microcontroller and then uses a Zigbee module to connect to a computer was proposed in [3]. In this study, there are drawbacks because it can only monitor in several places at once and the monitoring results can only be accessed on a local network. Research on the peat-land fire detection system in Riau

Province had been conducted. This tool uses an LM35 sensor, a fire sensor, and a humidity sensor. Data from the sensor is processed by the microcontroller and then sent via the HC12 module and the data is stored in databases to be displayed on a computer application, if a fire is detected, the application on the computer will send an SMS via the SMS gateway [4]. In this study, there is a lack of detection that can only be done on the local network. Research on overcoming fires in the home area, this tool uses temperature, humidity, CO sensors and smoke sensors on the sensor node which is connected to the Xbee device to the server (Raspberry Pi 2) and can be accessed in real-time on the cellphone with the application [5]. In this study, there is a lack of monitoring applications that can only be accessed through an Android device. Research on the development of IoT for detecting and tackling fires can be monitored via the web [6]. In this study, there are drawbacks in that this tool still uses an Ethernet cable to connect to servers for the web and applications.

This prototype fire detection system contributes as the simulation model to overcome problems in detecting fires to be more accurately. In this prototype, the implementation of detection data can be simulated, and it is easy to monitor through the website. This prototype can send a message to homeowners when the event of a fire occurred. Differences in the contribution of the research compared with previous studies are in Table 1.

Table 1. Delay Specification

	a	b	c	d
P	Serial	Serial	Serial	Concurrently
T	Wireless	Wireless	Wired	Wireless
S	Flexible	Flexible	Not Flexible	Flexible
C	Y_1	Y_2	Y_3	Y_4

where P = Processing Data, T = Transmitting data, S = Sensor Node Placement, C = Can be monitored, a = Article 1 [4], b = Article 2 [5], c = Article 3 [6], d = This paper, Y_1 = Yes, in local network with pc application, Y_2 = Yes, with android application anywhere, Y_3 = Yes, with website anywhere, and Y_4 = Yes, with website anywhere.

II. Research Method

Concurrency is the ability to process a task or process simultaneously. To be able to work concurrently, we need a scheduling algorithm so that the task or process can be processed based on the scheduling algorithm such as batch scheduling algorithm, interactive scheduling algorithm, real-time scheduling, and thread scheduling. This scheduler serves to determine the task or process that will be executed when faced with a choice of several tasks or processes to produce faster processing and in accordance with what we want [7].

This prototype fire detection system is intended for homes with a length of 12 meters and a width of 10 meters. To detect fires quickly, and precisely, and when a fire occurs, you can immediately notify the occupants of the house either directly while at home or through notification messages when outside the house and can be monitored in real-time through the website. Fig. 1 shows the placement of four sensor nodes with a red dot and one master node in the house with a yellow dot.

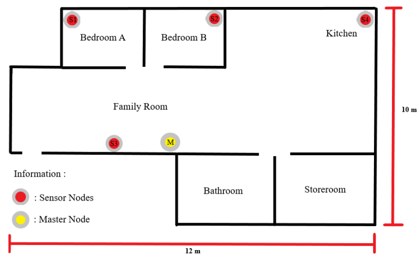

Fig. 1. Sensor node and master node placement plan.

Fig. 2 is a block diagram of the overall system, the Wireless Sensor Network system consists of four sensor nodes, each of which is connected to the master node using a Wi-Fi module with the ESP NOW protocol. After the master node receives data from the sensor node, the master node processes the data into several conditions. Then the data from the sensor node detection and the results of the conditioning process are sent to the server.

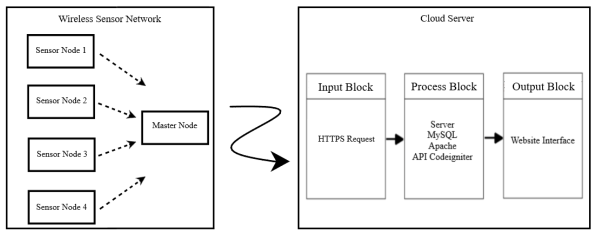

Fig. 2. System block diagram.

This prototype house fire detection system was built using the concept of the WSN which consists of sensor nodes in charge of retrieving data (fire and smoke detection). The master node collects data that has been taken by several sensor nodes and processes the condition of the data results and sends the results of the data processing to the cloud server, the data that is already on the cloud server is then processed and displayed in real-time on the website.

Fig. 3 is the architecture and workings of the wireless sensor network. Sensor node and master node using NodeMCU ESP8266 were developed. Several sensor nodes are connected to the master node wirelessly using a Wi-Fi network with ESP NOW protocol then the master node processes the results from the sensor nodes and sends them to the cloud server so that users can monitor the detection results through the monitoring website [1], [8]–[10]. This

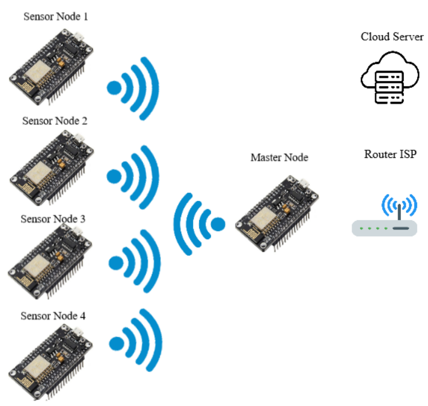

Fig. 3. Wireless sensor network architecture.

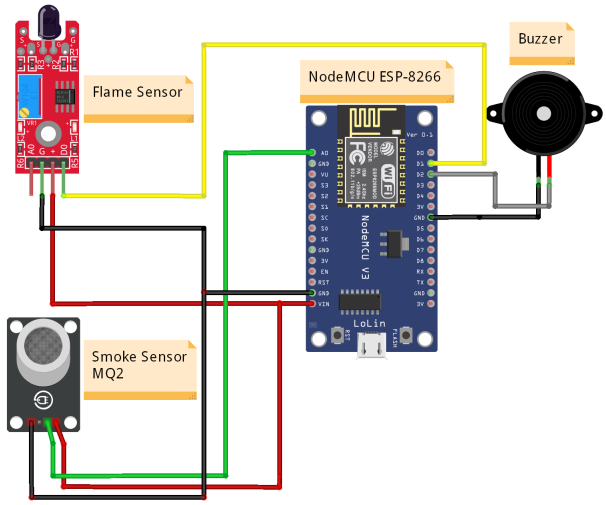

Fig. 4. Sensor Node Circuit Diagram

architecture is simple, when it is necessary to add sensor nodes, it can be done easily if the area is still in coverage without having to completely overhaul the system that is already running.

A. *Sensor Node*

The sensor node uses the NodeMCU ESP8266 as its microcontroller which already has a Wi-Fi module for communication [11]–[13]. The step-down module is used to lower the adapter voltage from 12V to 5V so that it can be used by microcontrollers and sensors. KY-026 fire sensor and MQ2 smoke sensor to detect the presence of fire and smoke in the event of a fire. Buzzer functions for immediate notification when a fire occurs.

The sensor node detects the presence of fire and smoke and then sends the results to the master node. The sensor node is supplied with voltage by a 12VDC adapter via the dc jack on the back. The switch serves to disconnect and connect the voltage from the dc jack to the step-down module. The step-down module functions as a voltage reducer from 12VDC to 5VDC and supplies it to the KY-026 microcontroller and fire sensor and MQ2 smoke sensor.

The reasons for using the KY-26 fire sensor and MQ2 smoke sensor are:

1) KY-026 can detect fire quickly and accurately with a maximum distance of 60cm and can provide fairly accurate fire detection data [14].
2) MQ2 can detect smoke in the range of 300ppm to 10,000 ppm and can provide fairly accurate smoke readings [13]–[15].

The circuit diagram of sensor nodes can be seen in Fig. 4. To detect a large area, you can add a few more sensor nodes to detect fires in the entire area you want to detect.

B. *Master Nodes*

The master node uses the NodeMCU ESP8266 as its microcontroller because the wireless sensor network system used to communicate uses the ESP NOW protocol, which is a protocol developed by Espressif. ESP NOW can only be used on ESP type microcontrollers [1], [11], [16]–[19]. The ESP8266 NodeMCU will receive data from the 4 sensor nodes and process the data into conditions and then send the condition results to the cloud server via the REST API. When one of the sensor nodes detects a fire, the master node will send a message via the telegram bot and send condition data to the cloud server. When the sensor nodes do not detect a fire, the master node will send condition data every 5 minutes to the server. The master node collects data from all sensor nodes and processes it then the results of the process will be sent to the cloud server via the REST API. The master node circuit can be seen in Fig. 5.

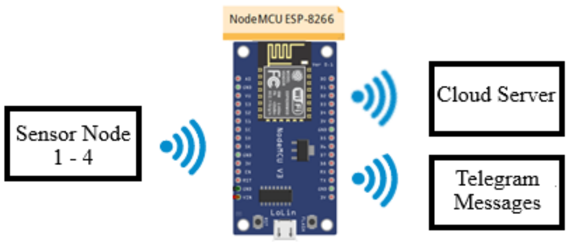

Fig. 5. Master node circuit diagram.

C. *Cloud Server*

The cloud server uses a Virtual Private Server (VPS) with specifications of 3 CPU cores, 3GB RAM, 40GB storage, and unlimited internet bandwidth. The operating system used is Ubuntu 20.04 LTS and installed several supporting software for REST APIs and Websites such as Apache, MySQL, and PHP 7.4.3. The REST API and Website are made using the CodeIgniter 4 framework and several tools such as bootstrap CSS, jQuery, and JavaScript.

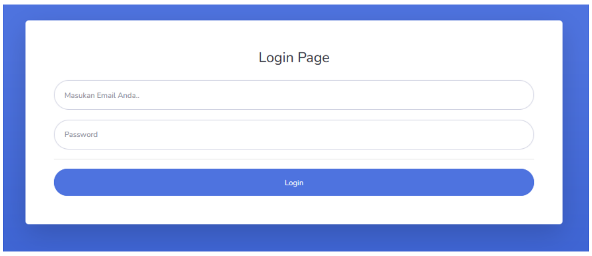

Fig. 6. Website login page

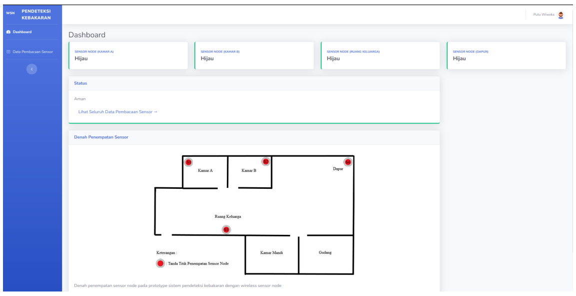

Fig. 7. Website dashboard.

The monitoring dashboard and all sensor reading data can only be accessed by logged-in users. This Dashboard menu present in Fig. 7 is useful for displaying real-time fire detection data from sensor nodes and their fire status. The Sensor Reading Data menu functions to view all detection data that has been sent to the server. The website can be accessed from anywhere using the internet and makes it easier to monitor the house in the event of a fire.

D. System Flowchart

This prototype fire detection system has several flowchart parts, namely flowchart sensor nodes, flowchart master nodes, flowchart cloud servers.

1) Sensor Node Flowchart: Fig. 8 is a sensor node flowchart where the process begins with the pin and variable initialization stage, followed by reading fire and smoke. If smoke or fire is detected, the buzzer will light up and send the board id and status = 1 data to the master node. If smoke or fire is not detected, the buzzer will turn off and send the data board id and status = 0 to the master node.
2) Master Node Flowchart: Fig. 9 is a master node flowchart, where the process begins with the pin and variable initialization stage. The reading of the sensor node results (S1 – S4) where conducted concurrently. There are 15 conditions (K 1-15 conditions) and 15 condition outcomes (M 1-15 Results). Example K-1 conditions where S1 = 0, S2 = 0, S3 = 0, and S4 = 1 produces M-1 where the result is "Fire Detected in the Kitchen". And so on up to 16 conditions. The placement of S1, S2, S3, and S4 can be seen in Fig. 1 If the results of reading S1 - S4 one of them gets input 1, then it checks the 15 conditions and sends a Telegram message based on the results of the conditions that are run and sends the status of S1 - S4 and the condition results to the server via the REST API. If all S1 – S4 get input 0 then it will check the SendTime condition, if the SendTime is not 5 minutes then it will read S1 – S4 again and if the SendTime is 5 minutes it will send S1 – S4 status and "safe conditions" to the server via REST API, and the SendTime will be set to 0 again.
3) Cloud Server Flowchart: Fig. 10 is a cloud server flowchart where the process begins with an https get request. When the master node sends data via the REST API, it will check the data transmission format and passcode to be able to input data. If you access the cloud server domain, you will be directed to the login page. If the login fails it will be directed to log back in, if the login is successful, it will display the monitoring dashboard.

III. Result and Discussion

The results of experiments carried out show that this tool has the potential to be adapted to everyday life to speed up countermeasures and to know for sure when a fire occurs in a certain area.

The sensor node detection data by the KY-026 fire sensor and the MQ2 smoke sensor will be sent to the master node. The delivery time needed to send data from the master node to the sensor node can be seen in Table 2.

Table 2. Delivery time of Sensor Node to Master Node

Delivery Time	Receipt Time	Delay	Delivery Status
10:52:31.070	10:52:31.940	870 ms	Success
10:52:32.052	10:52:32.972	920 ms	Success
10:52:33.079	10:52:33.955	876 ms	Success
10:52:34.064	10:52:34.986	922 ms	Success
10:52:35.111	10:52:35.974	863 ms	Success
Average delay		**890 ms**	

The test results in Table 2 are obtained by connecting the sensor node and master node to the laptop and then opening the Arduino IDE serial monitor to read the process of sending data on the sensor node to the master node and recording the time of sending and receiving data to find out the delivery delay. From 5 attempts to send data, all data can be sent perfectly, and it doesn't take too long to send data, even fast.

The data from the sensor nodes received by the master nodes are then processed and conditioned and then sent the data from the process to the cloud server via the REST API provided for display on the website. The time required for sending data from the master node to the server can be seen in Table 3.

The test results in Table 3 are obtained by connecting the master node to a laptop and then opening the serial monitor to read the delivery process

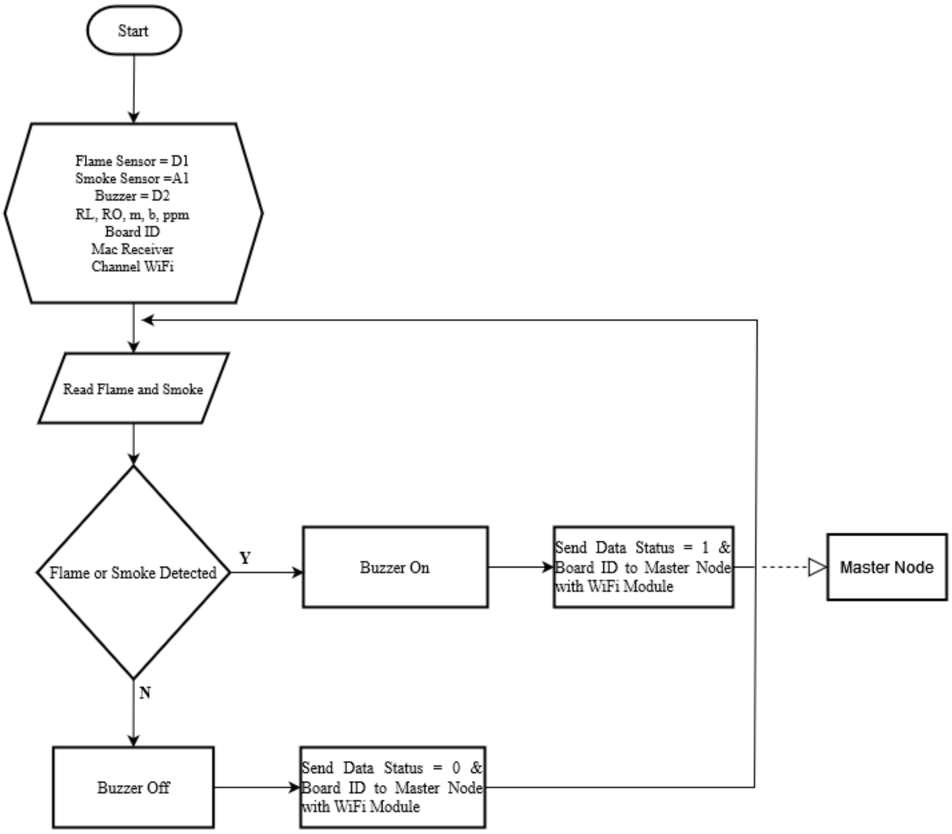

Fig. 8. Sensor node flowchart.

and then recording the delivery time, opening the monitoring website, and displaying the master node input data to see the time of receipt on the cloud server. From 5 attempts to send data based on Table 3, all data can be sent perfectly, and it doesn't take too long to send data, even fast.

Table 3. Delivery time of Master Node to Cloud Server

Delivery Time	**Receipt Time**	**Delay**	**Delivery Status**
14:08:19.415	14:08:19.756	341 ms	Success
14:09:56.893	14:09:57.245	352 ms	Success
14:10:53.345	14:10:53.703	349 ms	Success
14:12:39.180	14:12:39.554	374 ms	Success
14:13:30.744	14:13:31.100	356 ms	Success
Average delay		**345 ms**	

When the sensor node detects a fire and sends data to the master node, the master node will send a message via telegram to the homeowner and the master node will immediately send the condition processing data to the cloud server. When all sensor nodes do not detect fire or smoke, the master node will send data to the cloud server every 5 minutes. The sending of fire notification messages by the master node via the telegram bot API uses an internet connection therefore there is a delay in delivery. Delay in delivery can be seen in Table 4.

The test results in Table 4 are obtained by connecting the master node to the laptop and then opening the serial monitor to read the process of sending telegram messages when a fire condition occurs and then recording the delivery time, the reception time is obtained through the cellphone receiving the telegram message. From 5 attempts to send data based on Table 3, all telegram messages can be sent perfectly, and it doesn't take too long to send messages, even fast.

Table 4. Delay Notification Message Delivery

Delivery Time	**Receipt Time**	**Delay**	**Delivery Status**
14:08:18.777	14:08:19.784	1.007 ms	Success
14:09:56.261	14:09:57.389	1.128 ms	Success
14:10:52.711	14:10:53.849	1.138 ms	Success
14:12:38.545	14:12:39.709	1.164 ms	Success
14:13:30.744	14:13:31.167	423 ms	Success
Average delay		**972 ms**	

IV. Conclusion

This prototype house fire detection system consists of 4 sensor nodes and 1 master node. The master node concurrently processes the 4 input data from the sensor node into 16 conditions, so the system is more responsive. Based on testing the data transmission from the sensor node to the master node has an average delay of 870 milliseconds. The test of sending data from the master node to the cloud server has an average delay of 345 milliseconds. The test of sending telegram messages from the master node to the recipient's cellphone has an average delay of 972 milliseconds.

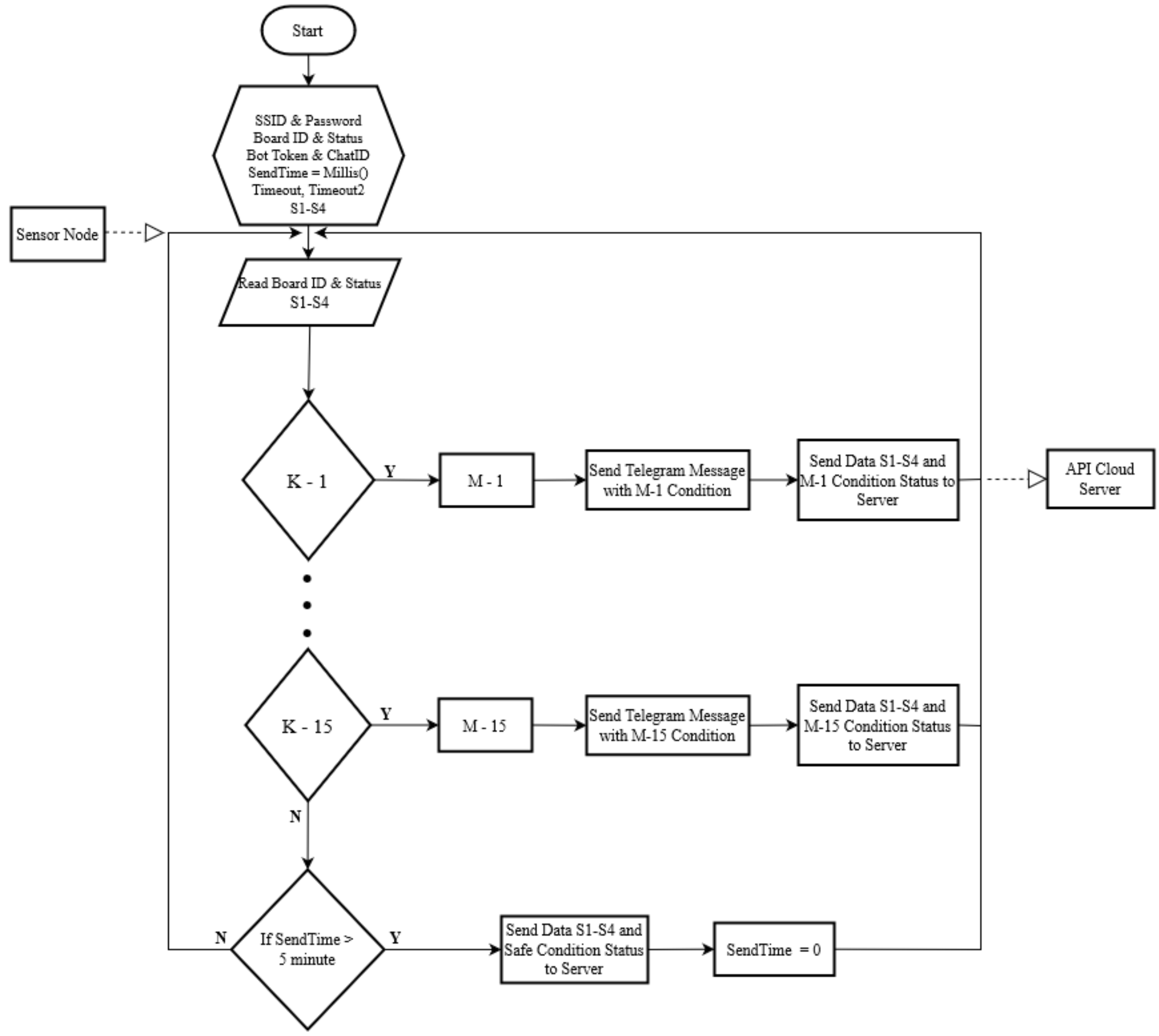

Fig. 9. Master node flowchart.

REFERENCES

[1] A. A. Putra and H. Nurwasito, "Implementasi middleware ESP32 menggunakan protokol ESP-NOW dan REST HTTP pada sistem monitoring microclimate," *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, vol. 6, no. 4, pp. 1576–1584, Apr. 2022.

[2] R. S. Kharisma and A. Setiyansah, "Fire early warning system using fire sensors, microcontroller, and SMS gateway," *Journal of Robotics and Control (JRC)*, vol. 2, no. 3, pp. 165–169, May 2021.

[3] M. Zhang, Y. Wan-Jun, N. -M. Cang, Y. Tian, J. -Y. Tang, and M. Zhang, "Research on indoor fire early warning system based on video image and smoke sensor," in *2020 5th International Conference on Intelligent Informatics and Biomedical Sciences (ICIIBMS)*, 2020, pp. 163–167, doi: 10.1109/ICIIBMS50712.2020.9336407.

[4] A. Sandhyavitri, R. Amri, M. Yusa, D. Fermana, and N. D. Ali, "Early warning systems using fire sensors, wireless, and SMS technology," in *2018 2nd International Conference on Electrical Engineering and Informatics (ICon EEI)*, 2018, pp. 108–113, doi: 10.1109/ICon-EEI.2018.8784315.

[5] M. U. Harun Al Rasyid, D. Enda, and F. A. Saputra, "Smart home system for fire detection monitoring based on wireless sensor network," in *2019 International Electronics Symposium (IES)*, 2019, pp. 189–194, doi: 10.1109/ELECSYM.2019.8901528.

[6] R. H. Eltom, E. A. Hamood, A. A. Mohammed, and A. A. Osman, "Early warning firefighting system using internet of things," in *2018 International Conference on Computer, Control, Electrical, and Electronics Engineering (ICCCEEE)*, 2018, pp. 1–7, doi: 10.1109/ICCCEEE.2018.8515815.

[7] E. A. Lee and S. A. Seshia, Introduction to embedded systems : a cyber-physical systems approach., Mit Press, 2016..

[8] M. S. Koushik, M. Srinivasan, R. Lavanya, S. Alfred, and S. Setty, "Design and development of wireless sensor network based data logger with ESP-NOW protocol," in *2021 6th International Conference for Convergence in Technology (I2CT)*, 2021, pp. 1–5, doi: 10.1109/I2CT51068.2021.9417914.

[9] R. R. Isnanto, Y. E. Windarto, J. I. D. Gloriawan, and F. N. Cesara, "Design of a robot to control agricultural soil conditions using ESP-NOW protocol," in *2020 Fifth International Conference on Informatics and Computing (ICIC)*, 2020, pp. 1–6, doi: 10.1109/ICIC50835.2020.9288575.

[10] D. Yukhimets, A. Sych, and A. Sakhnenko, "Designing a method for constructing distributed open ACS based on the ESP-NOW wireless protocol," in *2020 International Russian Automation Conference (RusAutoCon)*, 2020, pp. 642–647, doi: 10.1109/RusAutoCon49822.2020.9208135.

[11] Y. S. Parihar, "Internet of things and nodemcu: A review of use of nodemcu ESP8266 in IoT products," *J. of Emerging Technologies and Innovative Research*, vol. 6, no. 6, pp. 1085–1088, 2019.

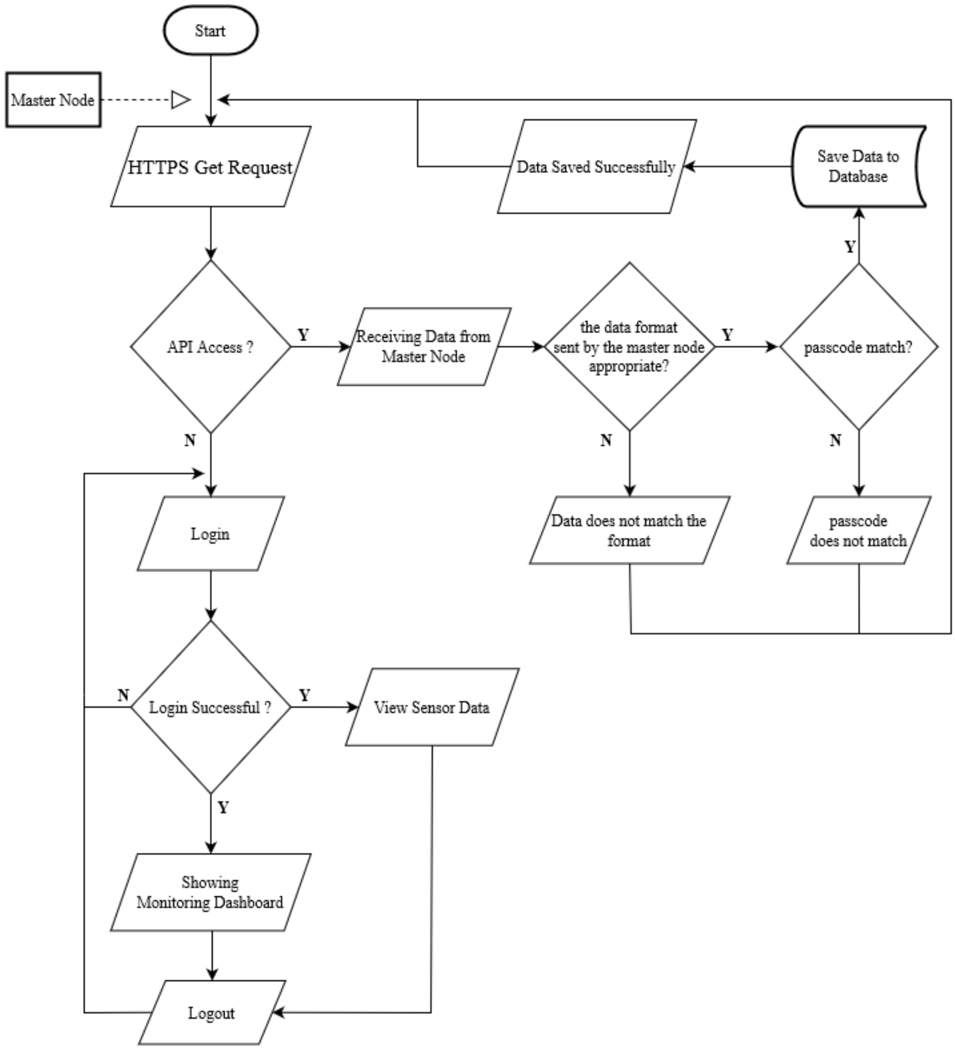

Fig. 10. Cloud server flowchart.

[12] A. V. Zinkevich, ”ESP8266 microcontroller application in wireless synchronization tasks,” in *2021 International Conference on Industrial Engineering, Applications and Manufacturing (ICIEAM)*, 2021, pp. 670–674, doi: 10.1109/ICIEAM51226.2021.9446411.

[13] R. K. Kodali, T. Devi B., and S. C. Rajanarayanan, ”IOT based automatic LPG gas booking and leakage detection system,” in *2019 11th International Conference on Advanced Computing (ICoAC)*, 2019, pp. 338-341, doi: 10.1109/ICoAC48765.2019.246863.

[14] J. Mulyono and E. Apriaskar, ”Simulasi alarm kebakaran menggunakan sensor Mq-2, falme sensor berbasis mikrokontroler arduino,” *Journal Ilmiah Elektronika dan Komputer*, vol. 14, no. 1, pp. 16–25, 2021.

[15] R. Inggi and J. Pangala, ”Perancangan alat pendeteksi kebocoran Gas LPG menggunakan sensor MQ-2 berbasis arduino,” *SIMKOM*, vol. 6, no. 1, pp. 12–22, 2021.

[16] A. Siswanto, ”Evaluasi kinerja wireless 802.11N untuk e-learning,” *IT Journal Research and Development*, vol. 1, no. 2, pp. 13–25, 2017.

[17] T. N. Hoang, S. -T. Van, and B. D. Nguyen, ”ESP-NOW based decentralized low cost voice communication systems for buildings,” in *2019 International Symposium on Electrical and Electronics Engineering (ISEE)*, 2019, pp. 108–112, doi: 10.1109/ISEE2.2019.8921062.

[18] K. Khanchuea and R. Siripokarpirom, ”A multi-protocol IoT gateway and WiFi/BLE sensor nodes for smart home and building automation: Design and implementation,” in *2019 10th International Conference of Information and Communication Technology for Embedded Systems (IC-ICTES)*, 2019, pp. 1–6, doi: 10.1109/ICTEmSys.2019.8695968.

[19] R. K. Kodali and V. S. K. Gorantla, ”RESTful motion detection and notification using IoT,” in *2018 International Conference on Computer Communication and Informatics (ICCCI)*, 2018, pp. 1–5, doi: 10.1109/ICCCI.2018.8441423.