IoT for Residential Monitoring Using ESP8266 and ESP-NOW Protocol

2 authors:

Mochamad Fajar Wicaksono
Universitas Komputer Indonesia
32 PUBLICATIONS   **75** CITATIONS

Myrna Dwi Rahmatya
Universitas Komputer Indonesia
27 PUBLICATIONS   **66** CITATIONS

IoT for Residential Monitoring Using ESP8266 and ESP-NOW Protocol

Mochamad Fajar Wicaksono [1], Myrna Dwi Rahmatya [2]

[1] Teknik Komputer, Universitas Komputer Indonesia, Jl. Dipati Ukur No.102-116, Bandung 40132, Indonesia
[2] Manajemen Informatika, Universitas Komputer Indonesia, Jl. Dipati Ukur No.102-116, Bandung 40132, Indonesia

ARTICLE INFO

ABSTRACT

This research aims to create a cluster housing monitoring system. This system combines IoT with the ESP-NOW protocol. In this system, the head of security and homeowners can unify the state of the house through a web application and an android application. The system will send notifications via the web and the LINE messaging application if there is an intrusion or gas leak or no fire is detected. The contribution of this research is to shorten the path and time of data transmission and make it easier for homeowners and heads of security to monitor and take action as soon as possible when something goes wrong in the residential area. The method used in this research is experimental. There are 4 main parts to this system, namely ESP-NOW sender, ESP-NOW receiver, gateway module, and server. All microcontrollers in this system use the NodeMCU ESP8266 because the NodeMCU, apart from being a controller, also functions as a WiFi module. By using the ESP-NOW protocol, it is possible for every microcontroller, both the sender and the receiver, to communicate without using WiFi WAP. In this study, the ESP-NOW sender collected 10 installed in each house to unify the conditions of the house. The ESP-NOW receiver worked to receive data from all senders. Any data received by the receiver will be sent to the gateway serially. The gateway module functions to send all messages to the server online. Application on the server-side built using PHP and MySQL. The display of the system can be opened through the android application or web application. The results of this test indicate that the system is running well with a 100% success proportion where every message from the sender is stored on the server-side, and the application can provide notifications according to the scenario of the system.

Corresponding Author:

Mochamad Fajar Wicaksono, Teknik Komputer, Universitas Komputer Indonesia, Jl. Dipati Ukur No.102-116, Bandung 40132, Indonesia
Email: mfajarw@email.unikom.ac.id

1. INTRODUCTION

Currently, many people are busy and active outside the home, so they are rarely at home. Meanwhile, the condition of the house is monitored by security. To assist security in monitoring the condition of the house, many have created a house and housing monitoring system where data related to the condition of the house is sent online to the server. This data is usually monitored by the admin or head of security as well as the owner of the house. In general, the process of sending sensor data is sent online by the controller used. This indicates that every controlled house must have an internet connection. This is a limitation of this method because not every house is willing and able to provide a continuous internet connection every day. On the other hand, there is another way to avoid this, namely by using the ESP-Mesh protocol, which will send data in a relay from each house or node to the server. However, this will result in a large delay related to sending the data [1]. Meanwhile, in terms of monitoring, the issue of data transmission time becomes

important. Therefore, it is necessary for new innovations to be carried out to anticipate these things. The innovation is the implementation of the ESP-NOW and IoT protocols. The use of the ESP-NOW protocol is necessary because not all homes have internet access that continues to be active every day. By using ESP-NOW, every home does not need to have an internet connection. Then, using the ESP-NOW protocol can shorten the data transmission path where each client directly sends data to the ESP-NOW server node directly, which in turn will reduce data transmission delays. As a comparison example, in the ESP-Mesh Protocol that sends data in a relay, where there are 4 MESH client nodes and 1 MESH server node assuming data transmission is carried out through each MESH client node and the delivery time from each MESH client node is 2 seconds, then the total time required for the farthest node is 8 seconds. Meanwhile, using the ESP-NOW protocol and in the case of a 2-second delivery time, it only takes an average of 2 seconds for each node to arrive at the ESP-NOW server. Furthermore, using the ESP-NOW protocol will save costs because every home does not need to have an internet connection.

This study describes the monitoring of cluster housing by combining IoT with the ESP-NOW protocol. In this system, there are two main actors related to monitoring, namely the head of housing security and the owner of the house. The number of houses monitored in this study amounted to 10. Both the head of security and the owner of the house can monitor the condition of the house through a web application or an Android application that has been built on this system. In particular, the head of security can see the entire condition of the house in the monitored housing. The system will provide notifications in the form of messages through the web application and the LINE messaging application if something goes wrong in the houses being monitored. The notification is given to the head of security and homeowners so that they can take action on the current situation.

Research related to monitoring home conditions by utilizing IoT has been very much done. One of the studies that discuss housing conditions is a study conducted in 2018 [2]. This study discusses home monitoring, and it has been running according to its purpose. However, the study only focused on one house.

The next research is research in 2019 [3]. This study discusses large-scale monitoring of areas using the Mesh protocol and has succeeded in achieving its goals. However, because it uses the Mesh protocol, the delay in sending data will be very large because each node functions as a relay to convey messages to the Mesh server.

The third research related to this research is research in 2021. This research makes housing monitoring by utilizing IoT and ESP-Mesh. The system built in this study has been running well, but in this system, there is a fairly large delay because it uses the ESP-Mesh protocol [4]. This ESP-Mesh protocol has the advantage of self-healing so that when one node is not connected, the other nodes will build their connection to get to the Mesh server [5], [6]. Because each node is responsible for delivering messages, the resulting delay will be very large, especially for the most distant nodes [1]. This is a drawback of the Mesh protocol.

In general, previous studies have not used other protocols that can overcome the problem of data transmission delay. The time of data transmission is very important because it is related to monitoring problems. The protocol that can solve this problem is the ESP-NOW protocol. The ESP-NOW protocol itself is a protocol created by Espressif, where this protocol allows multiple devices to communicate one-way or two-way without using Wi-Fi [7]–[11]. In this protocol, at least there is a controller that will act as master and slave. This protocol will be combined with the Internet of Things (IoT). By using IoT that combines this electronic equipment, we can get data from various devices such as microcontrollers and sensors [12], [13]. In addition, we can also exchange data with any equipment [14], [15].

The system created in this study was built using NodeMCU ESP8266, ESP-NOW protocol, PHP and MySQL, and App Inventor. App Inventor is used to create Android applications. This application directly points to the main URL address of this system so that what appears in the web application will automatically appear on the Android application. On the other hand, all controllers used in this study are ESP8266. The reason behind choosing this controller is that this controller is very compatible with the ESP-NOW protocol, whereas Espressif as the manufacturer, made this ESP-NOW so that it can be used by ESP boards, especially the NodeMCU ESP8266. In addition to the Wi-Fi function, the NodeMCU is equipped with several peripherals to be able to communicate with the outside world, such as 10 GPIO pins, an analog pin, and a pin for communication [16] so it is very suitable for the needs of this research.

In its implementation, 10 houses are implemented and monitored in this study. Furthermore, each ESP8266 stored in each house will be referred to as a sender node. Each sender will send data continuously to an ESP8266 which acts as a receiver node. All data received will be sent to the server via ESP8266, which acts as a gateway. Data from the ESP8266 gateway to the server is sent online. In the end, the final data will be received and stored in the database on the server. The stored data will be presented in tabular form in PHP web applications and android applications. The head of security can see all the data from each house, while the homeowners only have access to see the condition of their house. Each value sent and stored in the

database will determine how this system behaves. It is necessary to send a notification to the head of security and the owner of the house regarding the actual condition or not. These notifications can be received by the head of security and homeowners via the web app, android app, and LINE messaging app.

From all the explanations above, it can be concluded that the advantages and disadvantages of the system made in this study can be concluded. The advantage of this research is the relatively fast data reception time compared to the use of other protocols, where the time spent on sending data from each node is approximately the same. Another advantage is that each house does not need to install the internet for the controller node that is stored at home because, in this study, each node is connected to the central node using the ESP-NOW protocol. The disadvantage of this research is that when the receiver from esp-now is damaged, the data from all nodes will never reach the server. But despite its weaknesses, this research still has a contribution to housing monitoring that makes it easier for homeowners and heads of housing security to take action as soon as possible when something goes wrong in a residential area.

2. METHOD

The method used in building the system in this research is the experimental method. In its implementation, various experiments were carried out, which of course, we're following the basic theories that supported this research which was ultimately aimed at obtaining results following the initial objectives [17]. The process in this method starts with literature study, design, and simulation, and the last is to carry out implementation and testing to test the reliability of the system [18]. These processes are depicted in Fig. 1.

Fig. 1. The Flow of *the* Experimental Method of This Research

2.1. Study of Literature

In this section, we conduct various studies related to previous studies related to this research to obtain comparisons, research gaps, and contributions from the research. In addition, searches and studies related to the basic theories and supporting theories are carried out in this research. The results obtained are in the form of supporting theories to implement this research as well as several previous studies. Where in general the previous research, controlled houses must have an internet connection [2], and others still use the ESP-Mesh protocol, which results in large data transmission delays [3], [4].

2.2. System Design

In this section, the design process related to software hardware for this system is carried out. In the hardware section, block diagrams and circuit designs are made. The software design made includes algorithms for programs on hardware and programs for applications in this research. The block diagram design of the system made in this study is shown in Fig. 2.

From the block diagram, it can be seen that all the sender nodes, then the receiver and gateway nodes used, are NodeMCU ESP8266. It can be said that the main controller module in the system made in this research is NodeMCU ESP8266. This NodeMCU is a System on Chip (SoC) that is equipped with WiFi. [19]–[22]. The NodeMCU is built from the ESP-12E [23]. This NodeMCU is equipped with an L106 processor with a frequency of 80MHz and 224KB of RAM. To communicate with the outside world, the NodeMCU is equipped with an SPI pin, one analog pin, and 10 GPIO pins [16].

Each sender node will read the various sensors connected to it. The sensors that are read are two PIR (Passive Infrared Receiver) sensors, ultrasonic sensors, LDR sensors, gas sensors, and fire sensors. All values that are read describe the state of the house or sender. In this study, the HC-SR501 PIR sensor was used. A sensor is placed on the top right of the front of the house, while the other sensor is placed on the left of the front of the house. The PIR HC-SR501 sensor is used to detect human presence [23]–[25]. This sensor requires a voltage of 5V to work [26]. In this case, both sensors are used to detect an intruder. This sensor will give a HIGH-value output to the output pin when it detects human movement [27].

Furthermore, this system also uses ultrasonic sensors. This sensor is generally used to measure distance. In this study, the sensor used is the HC-SR04 module. This sensor module can be used to measure distances with a range from 2cm to 400cm [28]–[31]. The accuracy value of HC-SR04 is 3mm. This sensor works at a frequency of 40KHz [32]. In this system, the ultrasonic sensor has the function of detecting intruders. This sensor is placed in the center of the top in front of the house. This is intended so that when the PIR1 sensor or PIR2 sensor fails to work, there is still one sensor that can detect an intruder. This HC-SR04 requires a

voltage of 5V to work [33], [34]. When one of the PIR sensors is HIGH or the value received is less than the lowest limit, the NodeMCU, which acts as the sender node, will immediately take action, namely activating the buzzer.
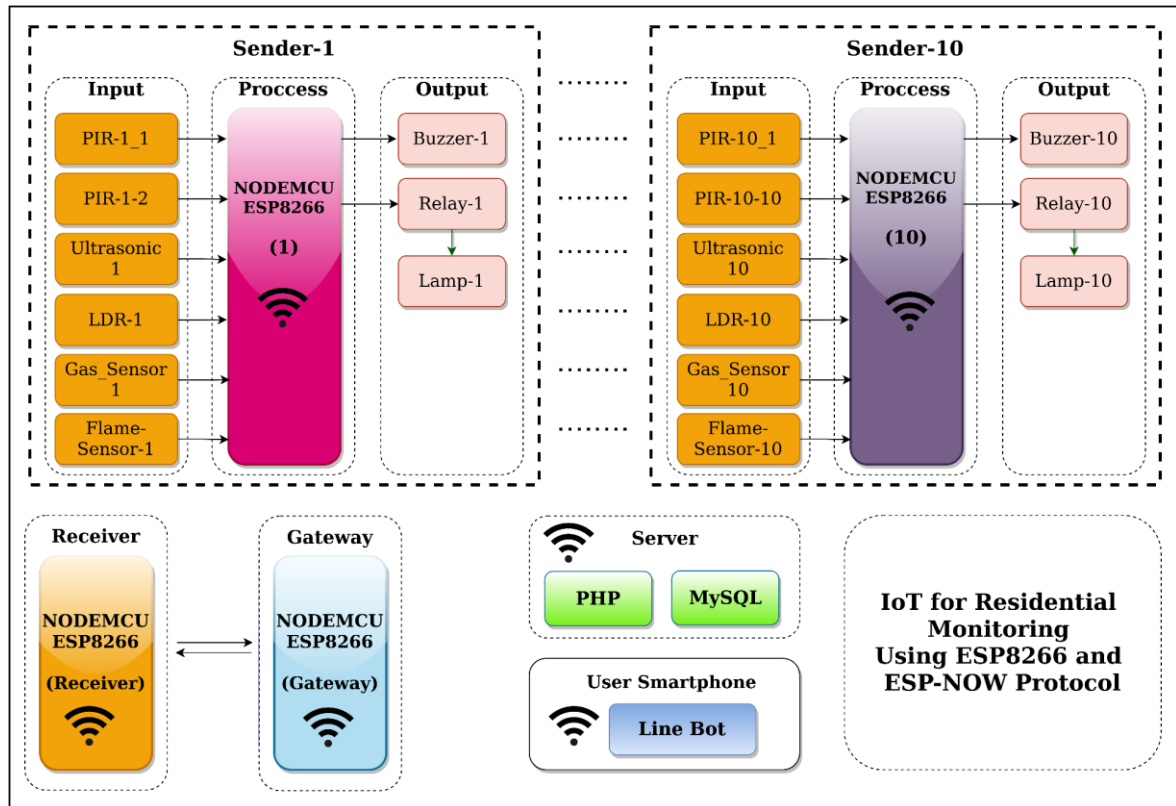

Fig. 2. IoT and ESP-NOW House Monitoring Block Diagram

Gas sensors are used to detect gas leaks in monitored homes. The gas sensor used is the MQ-2 module. This module is directly connected to the NodeMCU ESP8266. MQ-2 can be used to detect LPG gas leaks [35], [36]. The output used in this sensor is a digital value. This sensor works on a voltage of 5V. This sensor is placed in the kitchen. The next sensor is the fire sensor. In this study, it is used to detect the presence of fire as an early warning related to fires. The sensor module used is sensitive to the wavelength of fire with a range of 760nm to 1100nm [37]. This fire sensor module has a detection distance of up to 3 feet and a detection coverage of 60 degrees [38]. The output read from this sensor is a digital value. This sensor is placed in the kitchen. Furthermore, the LDR sensor in this system is used to measure the light in front of the house. If the light is dark, which means the received value has exceeded the specified limit, the sender node will automatically take action regarding this condition. The sender node will activate the relay so that the lights in the front of the house turn on. This relay itself is an electronic switch that will work when the coil is given an electric current [39].

In terms of sending data, each sender node will first read data from all sensors. After all the data is read, the sender node will send all the data to the receiver node wirelessly using the ESP-NOW protocol. Any data received at this receiver node will be sent first to the NodeMCU, which acts as a gateway. Sending data between the receiver node and the gateway using serial communication. Furthermore, every data received will be sent directly to the server.

On the server, the PHP and MySQL programming languages are used. PHP is included in server-side scripting. PHP command lines can be inserted into HTML lines and vice versa [40], [41]. PHP can be used as a backend system [42]. Any data received will be retrieved using the POST method and then saved to the database using the MySQL command. MySQL itself belongs to the type of RDBMS [43] and is open source and relatively easy to integrate with PHP [44].

Any sensor data received and stored on this server determines how the web application will act. The action taken by the server is in the form of giving notifications to the web application and sending it to the LINE messaging application using the bot LINE Notify. This notification is related to the violation of access

rights to the user's house, gas leaks, and detected fires that can cause fires. All data that states the condition of the house can be accessed through a web application by users, while specifically, the head of security can access and monitor the entire condition of the house or node.

In addition to the web application, this system also provides an Android application so that homeowners have other options regarding the monitoring application they will use. This android application is for security chiefs and homeowners. For the head of security, this application is optional only. Android applications are built using App Inventor. This application directly points to the main URL page of the monitoring website. After designing a block diagram, the next step is hardware-related design. The design of this hardware circuit is shown in Fig. 3.
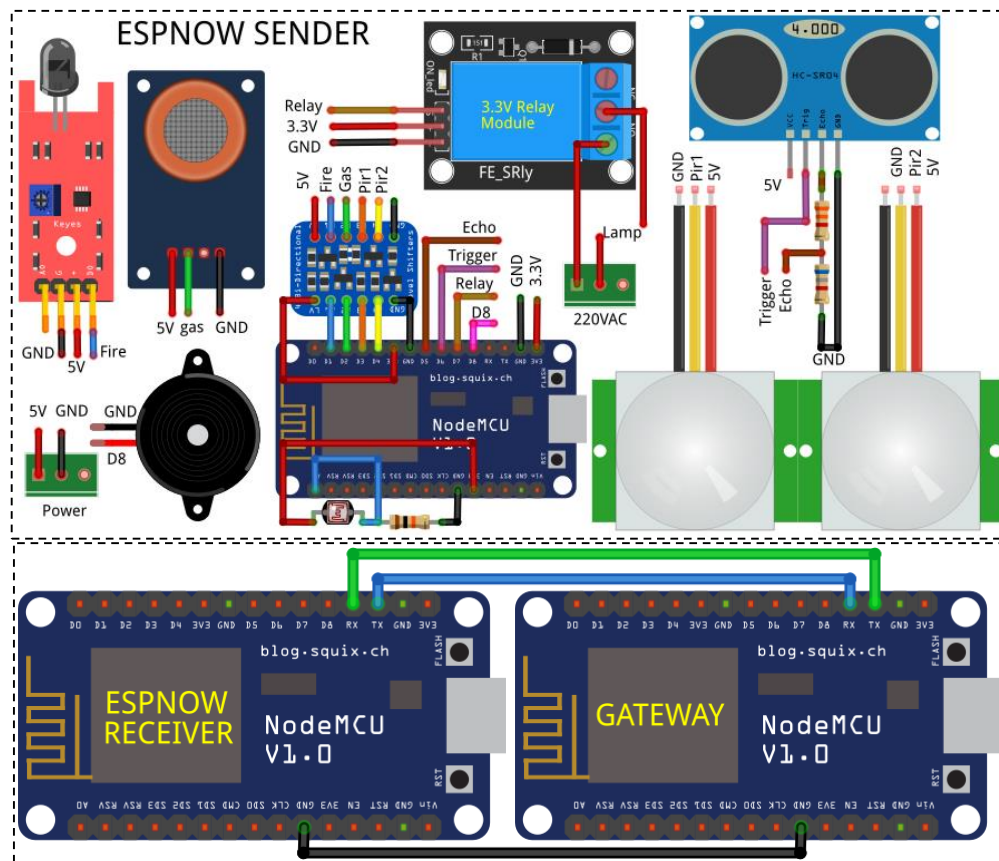

Fig. 3. IoT and ESP-NOW House Monitoring Electronic Design

Fig. 3 shows the circuit of this system. The top of the figure shows the circuit for all the sender nodes, which contains the NodeMCU, sensors, and actuators. Fire sensor, gas sensor, pir1 sensor, pir2 sensor, and ultrasonic sensor use digital I/O mode. In the circuit, it can be seen that the fire sensor module, all PIR sensors, and gas sensors are inserted into the bidirectional level converter module. This is because the output of all these sensors has an output level of 5V, while the HIGH-value limit for the I/O NodeMCU ESP8266 is 3.3V. The output of the ultrasonic sensor or echo pin is given a voltage divider to lower its output level to 3.3V. The LDR sensor in this circuit is connected to the NodeMCU ESP8266 analog pin. The lower part of the circuit shows the NodeMCU, which acts as a receiver from ESP-NOW, and the NodeMCU, which acts as a gateway to send data to the online server. After making a circuit design, the next step is to make a software design for this system. This design includes software design for all hardware and software design for the server-side. The software design for all hardware is divided into two parts, namely the design for the sender node and the design for the receiver and gateway nodes. The following is a software flow design for each sender node presented in the form of a text algorithm:

1. Start
2. I/O initialization
3. Variable initialization
4. Initialize ESP-NOW as the sender
5. Read the PIR1 sensor and save the result to a variable

6.   Is the value of the PIR1 sensor HIGH value?
7.   If yes, then:
  - activate the buzzer
  - go to point 9
8.   If not, then go to point 9
9.   Read the PIR2 sensor and save the result to a variable
10. Is the value of the PIR1 sensor HIGH value?
11. If yes, then:
  - activate the buzzer
  - go to point 13
12. If not, then go to point 13
13. Read the Ultrasonic sensor and save the result to a variable
14. Is the distance value less than 300cm?
15. If yes, then:
  - activate the buzzer
  - go to point 17
16. If not, then go to point 17
17. Read the gas sensor and save the result to a variable
18. Is the output value that reads LOW?
19. If yes, then:
  - activate the buzzer
  - go to point 21
20. If not, then continue to point 21
21. Read fire sensor and save the result to a variable
22. Is the read value LOW?
23. If yes, then:
  - activate the buzzer
  - go to point 25
24. If not, then go to point 25
25. Read the LDR sensor and save the result to a variable
26. Convert analog value data to the digital value
27. Is the read value less than or equal to 0?
28. If yes, then:
  - activate the relay to turn on the light
  - go to point 30
29. If not, then go to point 30
30. Save all sensor values to a variable according to the format.
31. Send to receiver node with ESP-NOW protocol.
32. Back to point 5

The next section is the design of the software flow for the receiver node. The following flow is presented in the form of a text algorithm:

1.   Start
2.   Serial and variable communication initialization
3.   Initialize ESP-NOW as a receiver
4.   Initialize data reception function
5.   Has any data been received from the sender?
6.   If yes, then:
  - Save data to a variable
  - Send serially to the gateway
  - Back to point 5.
7.   If not, then return to point 5.

The next design is the flow design for NodeMCU, which acts as a gateway. The following plot is presented in the form of a text algorithm:

1.   Start
2.   Serial and variable communication initialization
3.   Initialize WiFi connection

4. Is there serial data received?
5. If yes, then:
    - Send the data to the server with the POST method
    - Continue to point 4.
6. If not, then continue to point 4.

Next is the design of the software flow on the server, which includes data reception, data storage, and actions to be carried out. The following flow is presented in the form of a text algorithm:
1. Start
2. Was any data received (using the POST method)?
3. If yes, then:
    - Parse sensor data
    - Save data to the database
    - Process sensor data
    - If one of the values is outside the specified safe limit, then send the appropriate notification to the front end and send a LINE notification
    - If it is safe, go to point 5.
4. If not, then continue to point 5.
5. Was the "recover" button pressed by the user?
6. If yes, then:
    - Update the status related to the sensor to be "Recovered" in the database according to the corresponding id.
    - Continue to point 2.
7. If not, then continue to point 2.

2.3. Simulation
At this stage, various experiments are carried out to simulate the program that has been made. The simulation process for the hardware is carried out on the Arduino IDE software. The simulation process of the application program is carried out on the browser, which involves the backend of the application program and the display. In addition, the application simulation process is also carried out on smartphones regarding notifications to the LINE messaging application and the display on the Android application.

2.4. Implementation and testing
In this section, the entire system is put together for testing. All sender nodes, receiver nodes, gateways, and servers are connected. The test is carried out according to the program scenario. It is intended to get results following the initial goal. The testing process carried out includes testing per part and overall testing.

3. RESULTS AND DISCUSSION
In the system that has been created, the head of security can monitor housing conditions, and homeowners can monitor the condition of their respective homes. In addition, the head of security and homeowners have been notified via the LINE messaging application when something goes wrong in the monitored house, so they can immediately take the necessary action. In previous studies, they only monitored one house and directly used online connections [1]. So that when it is reproduced, it will be difficult for homeowners because they have to provide a continuous internet connection. Another study monitored many houses and objects using the mesh protocol and was able to send data without an internet connection [2], [3]. However, because it has to relay data to each node, it causes a greater delay for nodes that are farther away [6]. Meanwhile, this research monitors many houses using the ESP-NOW protocol, where each sender node sends data directly to the receiver node so that large data reception delays can be avoided. This is evidenced by the results of the tests that have been carried out, where the delay in sending data from each ESP-NOW node to the server is the fastest at 2 seconds and the longest is 4 seconds, as shown in Fig. 6 and Fig. 8.

This is evidenced by the tests that have been carried out. Testing of this system is carried out by testing each part of the system block and testing as a whole according to the scenario and system objectives. Testing of the block section includes testing the functionality of the sender node, testing the sending of data from the sender node to the receiver node, and testing the serial data transmission from the receiver node to the gateway. The testing section of the sender node functionality is shown in Table 1.

Table 1 shows that the test involved 10 sender nodes. The test results show that each sender node can read all sensors, activate the buzzer, and turn on the lights by activating the relay. The test for each sender

node is carried out 10 times, but to shorten the writing on the table, it is only written once. The success percentage of this test is 100%. After successfully ensuring that each sender node can carry out its function, the next step is to test the sending of data from the sender node to the receiver node using the ESP-NOW protocol. The test results of sending data from the sender node to the receiver node are shown in Table 2.

Table 1. Sender System Test Functionality

Sender	PIR1	PIR2	Ultrasonic	LDR	Gas	Flame	Buzzer	Relay
1	100%	100%	100%	100%	100%	100%	100%	100%
2	100%	100%	100%	100%	100%	100%	100%	100%
3	100%	100%	100%	100%	100%	100%	100%	100%
4	100%	100%	100%	100%	100%	100%	100%	100%
5	100%	100%	100%	100%	100%	100%	100%	100%
6	100%	100%	100%	100%	100%	100%	100%	100%
7	100%	100%	100%	100%	100%	100%	100%	100%
8	100%	100%	100%	100%	100%	100%	100%	100%
9	100%	100%	100%	100%	100%	100%	100%	100%
10	100%	100%	100%	100%	100%	100%	100%	100%

Table 2. Test Send Data from Sender to Receiver

No.	NodeMCU Sender	NodeMCU Receiver	Status
1	sent	received	valid
2	sent	received	valid
3	sent	received	valid
4	sent	received	valid
5	sent	received	valid
6	sent	received	valid
7	sent	received	valid
8	sent	received	valid
9	sent	received	valid
10	sent	received	valid

The data in Table 2 shows that the receiver node has successfully received data from each sender node. The data sent is all sensor data such as PIR1 sensor data, PIR2 sensors, ultrasonic sensors, LDR sensors, gas sensors, and fire sensors. All data is sent in the form of a string. One row of data in the table represents 10 attempts for each node. Furthermore, after all, data from the sender node has been received by the receiver node, the next step is to test the data transmission from the receiver node to the gateway, which in this case utilizes serial communication. The test data for this process is shown in Table 3.

Table 3. Test Send Serial Data from The Receiver to Gateway

No.	NodeMCU Receiver	Gateway	Status
1	sent	received	valid
2	sent	received	valid
3	sent	received	valid
4	sent	received	valid
5	sent	received	valid
6	sent	received	valid
7	sent	received	valid
8	sent	received	valid
9	sent	received	valid
10	sent	received	valid

Table 3 shows that serial communication between the sender and gateway nodes was successful. This success is based on data row data that was successfully received at the gateway following the format in the form of a predetermined string. The data that has been successfully received by this gateway has been stored in a variable and can then be sent to the server at the next testing step.

Specifically, for testing the delivery of data to the server, it is done by testing all hardware and applications on the system. In other words, this test goes into testing the entire system with possible conditions. The test involved 10 sender nodes, receiver nodes, gateways, PHP applications which include storage to the database, front end display, web application notifications, android, and LINE message notifications. The data from this test are presented in the form of tables and figures. The data from the test results for sending data from the sender node to the data successfully stored in this database is shown in Table 4.

Table 4. Test Send Data from All Sender, Receiver, Gateway, and Storing Data on The Server

Sender Number	NodeMCU Sender (send)	NodeMCU Receiver (receive and send to gateway serially)	Gateway (receive serial data and send to server)	Server (Database)
1	✓	✓	✓	valid and saved
2	✓	✓	✓	valid and saved
3	✓	✓	✓	valid and saved
4	✓	✓	✓	valid and saved
5	✓	✓	✓	valid and saved
6	✓	✓	✓	valid and saved
7	✓	✓	✓	valid and saved
8	✓	✓	✓	valid and saved
9	✓	✓	✓	valid and saved
10	✓	✓	✓	valid and saved

The test data in Table 4 shows that all sensor data for each sender node has been successfully sent to the receiver node and then continued to the gateway until it is stored in the database. An example of a successfully saved data row is shown in Fig. 4.

id	pesan	sensor1	sensor2	sensor3	sensor4	sensor5	sensor6	waktu
11996	1	0	0	372	0	0	1.58	2022-02-11 13:22:06
11997	4	0	0	356	0	0	3.94	2022-02-11 13:22:07
11998	1	0	0	366	0	Recovered	1.93	2022-02-11 13:22:08
11999	4	0	0	316	0	0	2.85	2022-02-11 13:22:09
12000	2	0	0	338	Recovered	Recovered	1.45	2022-02-11 13:22:09
12001	3	Recovered	Recovered	366	0	0	3.71	2022-02-11 13:22:10

Fig. 4. Sample Data Saved in the Database

From Fig. 4, it can be seen that the data is already stored in the database. The "message" column shows the number of the sender node. The "sensor1" and "sensor2" columns are the PIR1 and PIR2 sensor data columns. The "sensor3" column is the ultrasonic sensor data column, the "sensor4" column is the fire sensor column, and the "sensor5" column is the fire sensor column. The gas sensor data column and the "sensor6" column is the LDR sensor data column. Especially for fire sensor data, in this study, the data to be stored is negated first so that when there is no fire, the value 0 will be written in the database.

Related the time that the sending and receiving of data from each node can be seen in Fig. 4. This message column shows the number of the sender. Based on this data, data from node sender 4 is received at the longest every two seconds, then data from node sender 1 also has an interval of at most every 2 seconds. This is an advantage of ESP-NOW compared to ESP-MESH. Because each ESP-NOW sender node sends data directly to the receiver node, so long delays are avoided. The string "Recovered" indicates that

something went wrong in the monitored home or on the sender node previously and has been followed up by the head of security or the homeowner. These data can be accessed on the web page.

As explained in the design section, data access rights on android web pages are divided into two parts, namely for the head of security and the owner of the house. The head of security has access to monitor the entire condition of the house registered on the system, while the homeowner can only see the condition of his house. The initial page display of this system web application is shown in Fig. 5.

| (a) | (b) |
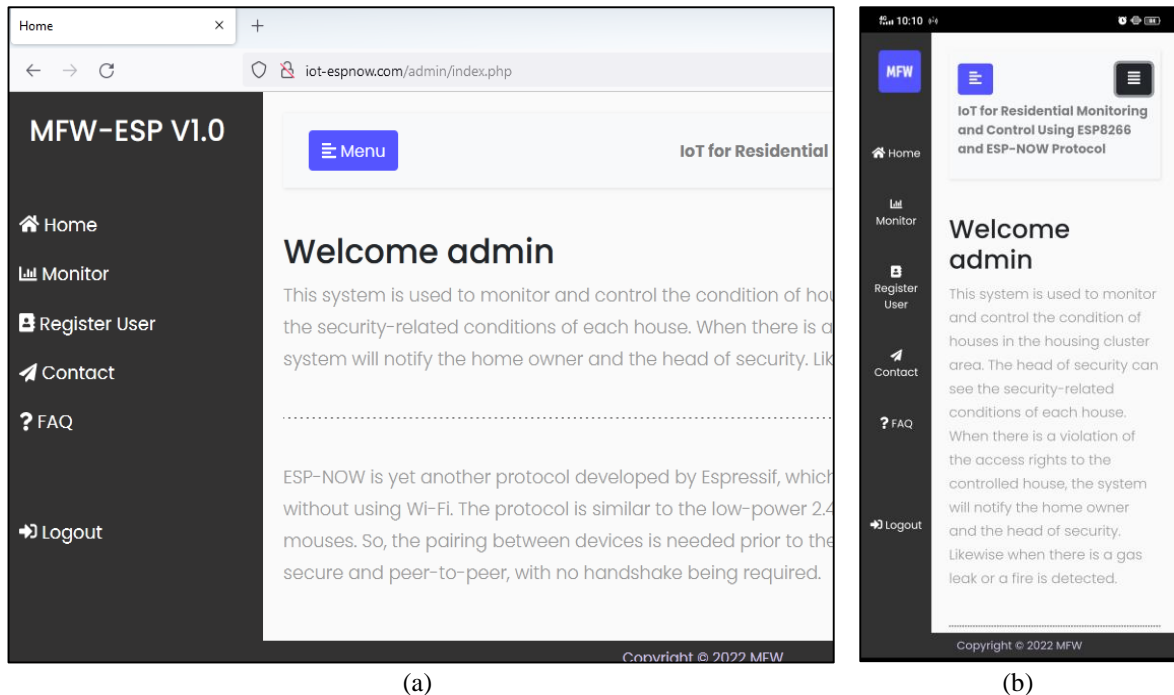
Fig. 5. IoT and ESP-NOW House Monitoring Start Page (a) Web Page, (b) Android Page

From Fig. 5, it can be seen that the head of security has more roles besides monitoring the condition of each house, namely, the head of security can register new users, in this case, the new node to be monitored. The monitoring page presented related to monitoring is shown in Fig. 6.

Fig. 6. IoT and ESP-NOW House Monitoring Alert

The data presented in Fig. 6 for the head of security shows that there was an intrusion at the sender node 3 (house number 3), fire and gas leaks were detected at sender node 2 (house number 1), and there was a gas leak at the sender node 1 (house number 1). When this happens, this web application will automatically send notifications to the head of security and homeowners both on the web page and on the Android application.

The recovery status on the web page, both on the home owner's page and on the head of the security page, can be changed by pressing the "Recover" button, as shown in Fig. 6. In this system, it is recommended that both the homeowner and the head of security press the "Recover" button after taking direct action at the scene of problems that arise. When the button is pressed, it will change its status to "Safe," as shown in Fig. 8. Pressing the button manually is intended to ensure that the necessary actions for this have been taken and the problem has been resolved. The refresh time of this page is 1 second. This means that the time required for the LINE notification message to reach the user is the data transmission time from the sender node to the server plus the page refresh time. The notification display that appears on the side of the security header is shown in Fig. 7.
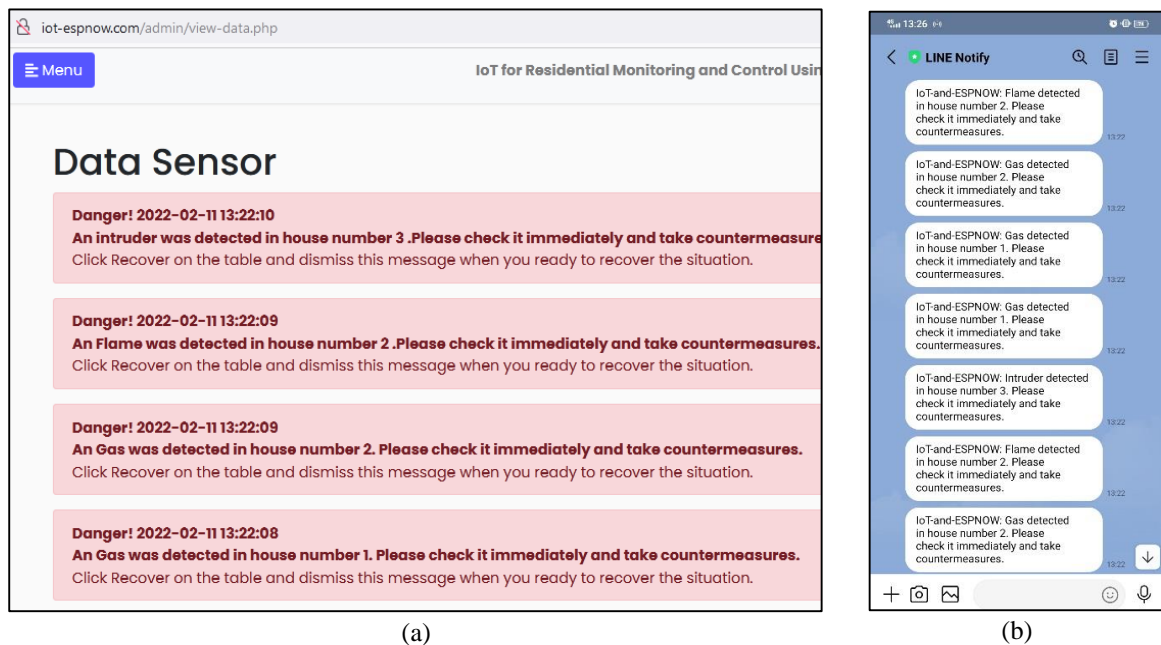

(a)   (b)

Fig. 7. Notification (a) Web Page, (b) LINE Notify Message

Fig. 8. Page View After Situation Recovered

At this point, both the head of security and the owner of the house can immediately take the necessary action according to the situation. When the countermeasures are ready to be taken, the head of security or the homeowner can press the safe button on the web application or Android application. The change in appearance when the action is taken is shown in Fig. 8.

The display in Fig. 8 shows and is an indicator that the necessary countermeasures related to this condition have been carried out. The display in Fig. 6 is related to conditions, Fig. 7. is related to notifications, and Fig. 8 is related to the results of the action, which also appears on the Android application the users. Based on all the results of the tests carried out, it can be concluded that this system has been running well, following the initial goal of monitoring housing conditions using IoT and ESP-NOW, where all data from each sender node can be stored on the server, and the system can provide notifications through the application such as Web, Android, and LINE Notify.

4. CONCLUSION

IoT for residential monitoring using ESP8266 and ESP-NOW Protocol has been running well according to the initial goal with a success percentage of 100%, where the system can present the condition of each house, can provide notifications via web applications and LINE messaging to the head of security and homeowners without a long delay. The delivery time from the sensor to the notification to the user is 3 to 4 seconds because there is a delay that is stored on the web page to refresh for 1 second. For further development, it may be possible to add more sender nodes and add more receiver nodes and test their reliability.

REFERENCES

[1] Y. Chia, R. Arjadi, E. Setyaningsih, P. Wibowo, and M. Sudrajat, "Performance Evaluation of ESP8266 Mesh Networks," *J. Phys. Conf. Ser.*, vol. 1230, p. 12023, Jul. 2019, https://doi.org/10.1088/1742-6596/1230/1/012023.

[2] M. C. Yuen, S. Y. Chu, W. H. Chu, H. S. Cheng, H. L. Ng, and S. P. Yuen, "A low-cost IoT smart home system," *Int. J. Eng. Technol.*, vol. 7, no. 4, pp. 3143–3147, 2018, https://www.sciencepubco.com/index.php/ijet/article/view/21554.

[3] A. Ometov, S. Bezzateev, N. Voloshina, P. Masek, and M. Komarov, "Environmental monitoring with distributed mesh networks: An overview and practical implementation perspective for urban scenario," *Sensors (Switzerland)*, vol. 19, no. 24, pp. 1–19, Dec. 2019, https://doi.org/10.3390/s19245548.

[4] M. F. Wicaksono and M. D. Rahmatya, "Smart Cluster Housing Monitoring with ESP32, ESP-Mesh and Django," *J. Eng. Res.*, vol. 9, pp. 1–16, 2021, https://doi.org/10.36909/jer.ASSEEE.16099.

[5] Z. Guo, X. Ma, P. Zhang, and Z. Liu, "A dust sensor monitoring system using Wi-Fi mesh network," *J. Phys. Conf. Ser.*, vol. 1754, pp. 1–7, 2021, https://doi.org/10.1088/1742-6596/1754/1/012129.

[6] R. Muhendra, A. Rinaldi, M. Budiman, and Khairurrijal, "Development of WiFi Mesh Infrastructure for Internet of Things Applications," *Procedia Eng.*, vol. 170, pp. 332–337, 2017, https://doi.org/10.1016/j.proeng.2017.03.045.

[7] M. I. Labib, M. ElGazzar, A. Ghalwash, and S. N. AbdulKader, "An efficient networking solution for extending and controlling wireless sensor networks using low-energy technologies," *PeerJ Comput. Sci.*, vol. 7, pp. 1–19, 2021, https://doi.org/10.7717/peerj-cs.780.

[8] T. N. Hoang, S. T. Van, and B. D. Nguyen, "ESP-NOW Based Decentralized Low Cost Voice Communication Systems for Buildings," in *Proceedings - 2019 International Symposium on Electrical and Electronics Engineering, ISEE 2019*, 2019, pp. 108–112, https://doi.org/10.1109/ISEE2.2019.8921062.

[9] A. A. Peters, F. J. Vargas, C. Garrido, C. Andrade, and F. Villenas, "PL-TOON: A Low-Cost Experimental Platform for Teaching and Research on Decentralized Cooperative Control," *Sensors*, vol. 21, no. 6, pp. 1–24, 2021, https://doi.org/10.3390/s21062072.

[10] A. Galajdova and R. Rakay, "Long Range IoT Communication," *MM Sci. J.*, vol. 4, pp. 4781–4785, 2021, https://doi.org/10.17973/MMSJ.2021_10_2021038.

[11] J. Linggarjati, "Design and Prototyping of Temperature Monitoring System for Hydraulic Cylinder in Heavy Equipment using ESP32 with data logging and WiFi Connectivity," *IOP Conf. Ser. Earth Environ. Sci.*, vol. 998, no. 1, p. 012042, 2022, https://doi.org/10.1088/1755-1315/998/1/012042.

[12] I. Ardiansah, N. Bafdal, E. Suryadi, and A. Bono, "Greenhouse monitoring and automation using arduino: A review on precision farming and Internet of Things (IoT)," *Int. J. Adv. Sci. Eng. Inf. Technol.*, vol. 10, no. 2, pp. 703–709, 2020, https://doi.org/10.18517/ijaseit.10.2.10249.

[13] N. Scarpato, A. Pieroni, L. Di Nunzio, and F. Fallucchi, "E-health-IoT universe: A review," *Int. J. Adv. Sci. Eng. Inf. Technol.*, vol. 7, no. 6, pp. 2328–2336, 2017, https://doi.org/10.18517/ijaseit.7.6.4467.

[14] P. Gope, R. Amin, S. K. Hafizul Islam, N. Kumar, and V. K. Bhalla, "Lightweight and privacy-preserving RFID authentication scheme for distributed IoT infrastructure with secure localization services for smart city environment," *Futur. Gener. Comput. Syst.*, vol. 83, pp. 629–637, 2018, https://doi.org/10.1016/j.future.2017.06.023.

[15] M. Kashyap, V. Sharma, and N. Gupta, "Taking MQTT and NodeMcu to IoT: Communication in Internet of Things," *Procedia Comput. Sci.*, vol. 132, pp. 1611–1618, 2018, https://doi.org/10.1016/j.procs.2018.05.126.

[16] R. R. Rout, S. Vemireddy, S. K. Raul, and D. V. L. N. Somayajulu, "Fuzzy logic-based emergency vehicle routing:

An IoT system development for smart city applications," *Comput. Electr. Eng.*, vol. 88, p. 106839, 2020, https://doi.org/10.1016/j.compeleceng.2020.106839.

[17] M. F. Wicaksono, S. Syahrul, and M. Dwi Rahmatya, "Development of Laboratory Equipment Inventory System Using Radio Frequency and Internet of Things," *J. Ilm. Tek. Elektro Komput. dan Inform.*, vol. 7, no. 2, p. 249, 2021, https://doi.org/10.26555/jiteki.v7i2.21114.

[18] H. Hidayat, N. S. Hasibuan, and F. Wicaksono, "Design And Implementation Of Electronic Examination Device For Improving The Blind Students' Comfort," *J. Eng. Sci. Technol.*, vol. 16, no. 1, pp. 807–815, 2021, https://jestec.taylors.edu.my/Vol%2016%20issue%201%20February%202021/16_1_56.pdf.

[19] K. Dimililer, H. Dindar, and F. Al-Turjman, "Deep learning, machine learning and internet of things in geophysical engineering applications: An overview," *Microprocess. Microsyst.*, vol. 80, p. 103613, 2021, https://doi.org/10.1016/j.micpro.2020.103613.

[20] J. Mocnej, M. Miškuf, P. Papcun, and I. Zolotová, "Impact of Edge Computing Paradigm on Energy Consumption in IoT," *IFAC-PapersOnLine*, vol. 51, no. 6, pp. 162–167, 2018, https://doi.org/10.1016/j.ifacol.2018.07.147.

[21] L. A. Kumar, V. Indragandhi, R. Selvamathi, V. Vijayakumar, L. Ravi, and V. Subramaniyaswamy, "Design, power quality analysis, and implementation of smart energy meter using internet of things," *Comput. Electr. Eng.*, vol. 93, p. 107203, 2021, https://doi.org/10.1016/j.micpro.2021.104025.

[22] A. K. Podder, A. A. Bukhari, S. Islam, S. Mia, M. A. Mohammed, N. M. Kumar, K. Cengiz, and K. H. Abdulkareem, "IoT based smart agrotech system for verification of Urban farming parameters," *Microprocess. Microsyst.*, vol. 82, p. 104025, 2021, https://doi.org/10.1016/j.micpro.2021.104025.

[23] M. Murad, O. Bayat, and H. M. Marhoon, "Design and implementation of a smart home system with two levels of security based on IoT technology," *Indones. J. Electr. Eng. Comput. Sci.*, vol. 21, no. 1, pp. 546–557, 2021, https://doi.org/10.11591/ijeecs.v21.i1.pp546-557.

[24] S. Bin Saleh, S. B. Mazlan, N. I. B. Hamzah, A. Z. Z. B. A. Karim, M. S. B. Zainal, S. A. B. Hamzah, D. B. M. Nor, and H. B. M. Poad, "Smart home security access system using field programmable gate arrays," *Indones. J. Electr. Eng. Comput. Sci.*, vol. 11, no. 1, pp. 152–160, 2018, https://doi.org/10.11591/ijeecs.v11.i1.pp152-160.

[25] N. Surantha and W. R. Wicaksono, "Design of Smart Home Security System using Object Recognition and PIR Sensor," *Procedia Computer Science*, vol. 135, pp. 465–472, 2020, https://doi.org/10.1016/j.procs.2018.08.198.

[26] B. Alathari, M. F. Kadhim, S. Al-Khammasi, and N. S. Ali, "A framework implementation of surveillance tracking system based on pir motion sensors," *Indones. J. Electr. Eng. Comput. Sci.*, vol. 13, no. 1, pp. 235–242, 2019, https://doi.org/10.11591/ijeecs.v13.i1.pp235-242.

[27] A. Murad, O. Bayat, and H. M. Marhoon, "Implementation of rover tank firefighting robot for closed areas based on arduino microcontroller," *Indones. J. Electr. Eng. Comput. Sci.*, vol. 21, no. 1, pp. 56–63, 2021, https://doi.org/10.11591/ijeecs.v21.i1.pp56-63.

[28] S. Dhingra, R. B. Madda, R. Patan, P. Jiao, K. Barri, and A. H. Alavi, "Internet of things-based fog and cloud computing technology for smart traffic monitoring," *Internet of Things (Netherlands)*, vol. 14, pp. 1–19, 2021, https://doi.org/10.1016/j.iot.2020.100175.

[29] K. D. Kang, H. Kang, I. M. S. K. Ilankoon, and C. Y. Chong, "Electronic waste collection systems using Internet of Things (IoT): Household electronic waste management in Malaysia," *J. Clean. Prod.*, vol. 252, p. 119801, 2020, https://doi.org/10.1016/j.jclepro.2019.119801.

[30] A. Elrayes, M. H. Ali, A. Zakaria, and M. H. Ismail, "Smart airport foreign object debris detection rover using LiDAR technology," *Internet of Things (Netherlands)*, vol. 5, pp. 1–11, 2019, https://doi.org/10.1016/j.iot.2018.11.001.

[31] M. M. Rayguru, M. R. Elara, R. Parween, B. Ramalingam, R. E. Abdulkader, and L. Yi, "A robust output feedback strategy for liquid handling using reconfigurable robots," *Autom. Constr.*, vol. 130, no. May, p. 103800, 2021, https://doi.org/10.1016/j.autcon.2021.103800.

[32] G. Montazeaud, C. Langrume, S. Moinard, C. Goby, A. Ducanchez, B. Tisseyre, and G. Brunel, "Development of a low cost open-source ultrasonic device for plant height measurements," *Smart Agric. Technol.*, vol. 1, p. 100022, 2021, https://doi.org/10.1016/j.atech.2021.100022.

[33] F. Valocky, M. Orgon, and I. Fujdiak, "Experimental Autonomous Car Model with safety sensor in Wireless Network," *IFAC-PapersOnLine*, vol. 52, no. 27, pp. 92–97, 2019, https://doi.org/10.1016/j.ifacol.2019.12.739.

[34] L. F. Maldaner, J. P. Molin, T. F. Canata, and M. Martello, "A system for plant detection using sensor fusion approach based on machine learning model," *Comput. Electron. Agric.*, vol. 189, p. 106382, 2021, https://doi.org/10.1016/j.compag.2021.106382.

[35] K. Aishwarya, R. Nirmala, and R. Navamathavan, "Recent advancements in liquefied petroleum gas sensors: A topical review," *Sensors Int.*, vol. 2, no. March, p. 100091, 2021, https://doi.org/10.1016/j.sintl.2021.100091.

[36] J. Saini, M. Dutta, and G. Marques, "Indoor air quality monitoring systems based on internet of things: A systematic review," *Int. J. Environ. Res. Public Health*, vol. 17, no. 14, pp. 1–22, 2020, https://doi.org/10.3390/ijerph17144942.

[37] W. L. Hsu, J. Y. Jhuang, C. S. Huang, C. K. Liang, and Y. C. Shiau, "Application of Internet of Things in a kitchen fire prevention system," *Appl. Sci.*, vol. 9, no. 17, 2019, https://doi.org/10.3390/app9173520.

[38] B. Sarwar, I. S. Bajwa, N. Jamil, S. Ramzan, and N. Sarwar, "An intelligent fire warning application using IoT and an adaptive neuro-fuzzy inference system," *Sensors (Switzerland)*, vol. 19, no. 14, pp. 1–18, 2019, https://doi.org/10.3390/s19143150.

[39] Y. Tjandi and S. Kasim, "Electric Control Equipment Based on Arduino Relay," *J. Phys. Conf. Ser.*, vol. 1244, no. 1, pp. 1–7, 2019, https://doi.org/10.1088/1742-6596/1244/1/012028.

[40] Asmaidi, M. A. Ihsan, D. S. Putra, M. Zulfa, R. Rusnanda, and M. Kartawijaya, "Development of e-journal system in South Aceh Polytechnic using web based application," in *IOP Conference Series: Earth and Environmental Science*, 2021, vol. 644, no. 1, pp. 1–12, https://doi.org/10.1088/1755-1315/644/1/012056.

[41] A. N. Azlina, A. B. Zanariah, and Y. Faridah, "Intelligent Home Automated System," *Indones. J. Electr. Eng. Comput. Sci.*, vol. 15, no. 2, pp. 733–742, 2019, https://doi.org/10.11591/ijeecs.v15.i2.pp733-742.

[42] T. O. Oladele, R. O. Ogundokun, A. A. Adegun, E. A. Adeniyi, and A. T. Ajanaku, "Development of an Inventory Management System Using Association Rule," *Indones. J. Electr. Eng. Comput. Sci.*, vol. 21, no. 3, pp. 1868–1876, 2021, https://doi.org/10.11591/ijeecs.v21.i3.pp1868-1876.

[43] C. P. C. Munaiseche, D. R. Kaparang, and P. T. D. Rompas, "An Expert System for Diagnosing Eye Diseases using Forward Chaining Method," in *IOP Conference Series: Materials Science and Engineering*, 2018, vol. 306, no. 1, pp. 1–8, https://doi.org/10.1088/1757-899X/306/1/012023.

[44] A. Setiyadi and E. B. Setiawan, "Information System Monitoring Access Log Database on Database Server," in *IOP Conference Series: Materials Science and Engineering*, 2018, vol. 407, no. 1, pp. 1–6, https://doi.org/10.1088/1757-899X/407/1/012110.

BIOGRAPHY OF AUTHORS

Mochamad Fajar Wicaksono Completed his master's degree in 2014. Currently active as a professional teacher with a certification issued by DIKTI in the field of computer engineering. The author's area of interest is instrumentation and control, especially IoT and embedded systems. Email: mfajarw@email.unikom.ac.id

Myrna Dwi Rahmatya Completed her master's degree in Information Systems in 2014. Currently active as a professional lecturer based on the certification issued by DIKTI. One of the fields occupied by the author is web programming. Email: myrna@email.unikom.ac.id