e-ISSN : 0976-5166
p-ISSN : 2231-3850

Low-Cost Security Monitoring System for Indoor Areas Based on The IoT

Yulianto[1*]

Computer Science Department, School of Computer Science
Bina Nusantara University, Jakarta 11480, Indonesia
*Email: yulianto003@binus.ac.id

Sidharta[2]

Computer Science Department, School of Computer Science
Bina Nusantara University, Jakarta 11480, Indonesia
Email: sidharta@binus.ac.id

Hanugra Aulia Sidharta[3]

Computer Science Department, School of Computer Science
Bina Nusantara University, Jakarta 11480, Indonesia
Email: hanugra.sidharta@binus.ac.id

Danang Wahyu Wicaksono[4]

Computer Science Department, School of Computer Science
Bina Nusantara University, Jakarta 11480, Indonesia
Email: danang.wahyu@binus.ac.id

Abstract

The growth of the Internet of Things (IoT) to be implemented for smart homes still increases. In the smart home area, the IoT has a task to control every electricity automatically. Not only that, the IoT gives the convenience for users to control manually all the electricity in their homes remotely. One of the important things that makes the IoT concept very needed in smart homes can be used to monitor the environment through sensors. The sensor can also be utilized for security systems, especially to monitor the presence of intruders. In the previous research, several ways to provide more protection in the room are by changing the door lock from mechanical conventional to automatic. A Keypad module for typing passwords and biometric recognition was also proposed to secure the door. The room's monitoring system based-on camera for object detection was also proposed. By using the camera to detect an intruder, it needs huge computational resources. The detection times were also considered. To provide the fastest and low-cost model to detect the presence of intruders this study proposes a security monitoring system based on IoT composed of edge devices based on ESP8266 that are equipped with an ultrasonic sensor of HC-SR04 as an input sensor, and for the user Interface the based-on Java application is included. The ultrasonic sensor is used to measure a distance. When the intruder is assumed to pass through the front of the ultrasonic sensor, the edge device will read the change in distance and send the warning signal to the monitoring system based on Java. With a satisfactory result, this model can detect the presence of intruders with a speed of only 274.798 milliseconds.

***Keywords*: object detection; smart room; ESP8266, ultrasonic sensor, Redis, Java, JSON Arduino.**

1. Introduction

In the modern era, where technology is growing fast, now every house is equipped with IoT technology, which makes every piece device in the house and environment can act automatically, smartly, and securely [1], it is referred to as a smart home system [2]. The research of smart home system covers several sub research i.e., the health of monitoring system that can be used to oversee the patient [3], the monitoring of mental older person [4], the monitoring of rooms temperature [5]. The monitoring of electricity consumption [6], Security access for door lock system [7], intruder detection [8][9][10], etc. By specific the intruder detection system is still challenging even though could handle by the research of door lock systems. The intruder still has the possibility to penetrate the room from window, roof, and other way. The one way to minimize an intruder to access into room is by giving an alarm notification. The security system to identify an intruder that doesn't have access to specific rooms and environments is needed to facilitate centralized monitoring convenience [9].

A recent study about intruder detection has been proposed by using the combination of web camera sensor, Raspberry Pi 3B+, and YOLO V3 framework algorithm that installed into Raspberry [11]. The total time required to detect is 49.673 seconds. Inspired by research about intruder detection that has been proposed before [11], this study proposed a framework based on IoT that can detect someone's presence in the room. The proposed method is composed of three parts. The first was an edge device that was put in a specific room or near the door. The edge device utilizes an ESP8266 as a core microcontroller. The ultrasonic sensor with the series of HC-SR04 was also included and connected to ESP8266, which functions as distance sensor. The second part was a monitoring system that functioned as a user interface. The monitoring system was developed by using a Java programming language. The reason for the Java application was used in this study is the Java programming language has fast performance for time execution especially in recursive operation [20]. In the third part of this study, the Redis no SQL databases were also proposed as the coupling between the edge devices with Java applications. Redis utilizes random-access memory (RAM) as its main memory environment. Every request both published and subscribed to by the client application to the server can be handled directly to RAM. It is possible to make the read and write data can fastest [17]. The objective of this study is to propose the framework for intruder detection, where the advantage of this proposed method is can detect an intruder by using low-cost component and fastest detection. With the maximum time detection was 27479,8 microseconds or 274.798 milli second the proposed framework can detect faster than previous research that has been proposed using computer vision technique [11].

2. Study Literature

Study literature related to IoT devices has been explained in detail by Schiller et al [12]. The Schiller in their research talks about several sensors that can be utilized in IoT architecture, for example for wireless data communications several devices that can be used are IPv6 over Low-power Wireless Personal Area Networks (6LoWPAN), Thread, Berkeley Low-power IP stack (BLIP), Constrained Application Protocol (CoAp or ZigBee) & Message Queuing Telemetry Transport (MQTT), and Graphite open-source data storage platform (Grafana). 6LoWPAN worked on IPv6 upon the User Datagram Protocol (UDP) transport layer. The thread is worked on IEEE 802.15.4 on 6LoWPAN architecture. The thread is commonly used on Personal Area Networks (PAN). The BLIP can work on TCP or UDP transport layer. CoAp & MQTT is a choice of the lightweight communication protocol to send (publish) and receive (subscribe) the data. To receive the data, Grafana as a graphical of dashboard display can be joined with the MQTT protocol.

The attack recognition system on the edge device has been proposed to classify which edge point that suspected as a threat in a mass of networking systems. Several machine learning algorithms were used to evaluate which algorithm of machine learning (as an example, J48, Naïve Bayes, Multilayer Perceptron, and Multinomial logistic regression) is more robust and suitable to classify the edge devices that potentially be a threat [13]. The research focused on human activity recognition has been conducted, where all the sensor data is classified using a deep learning algorithm to recognize human activity up to a fine-grained level [14].

The wireless sensor network (WSN) is a device that is equipped with the sensor to monitor physical environment. The WSN devices will read the environment and publish the data to server or other devices through a network. The WSN has an issue of attacked possibility by a hacker to make a false diagnosis against reality's condition. The research that proposed by Yadav et al [15], is tried to detect pre-hacking that attempted to WSN peripherals by using an artificial intelligence of deep learning algorithm. Other research to secure the physical devices likes computing, network devices was also conducted [16].

The NoSQL database promises how fast the throughput transfer rate of data is compared with the Relational Database Management Systems (RDBMS) platform. It is caused by the NoSQL database working in RAM hardware and the RDBMS access directly to the hard drive. The throughput benchmark to get information on how fast between two NoSQL platforms has been conducted by [17]. Redis and HBase are two platform NoSQL databases that are to be object measurement. The Yahoo Cloud Serving Benchmark (YCSB) was the platform used to measure the performance speed between that product.

The ubiquity of Arduino especially for Node MCU ESP8266 has been to be a main component that always used among hobbies, academics, and industries to develop many projects related to IoT devices [18][19]. The microcontroller of ESP8266 has also equipped with wireless communication protocol that can be used to communicate with other devices and other infrastructure [19].

A comparison of several famous programming languages, namely C, Java, Python, and MATLAB has been conducted by Cubukcu et al [20]. Their research tried to find the best algorithm with criteria for the fastest calculation, especially for recursive programming. The Fibonacci and Hanoi's Towers algorithm as test cases was used. The fact found that Java algorithms give satisfied time execution with a score of 15.4 seconds faster than C.

3. Proposed Method

To propose the security monitoring system that has advantage on fast detection and low-cost component, this study proposed the framework that can be seen in Fig. 1. The proposed framework is composed of edge device, cloud server application and monitoring system. The edge device is equipped with a single sensor of HC-SR04 ultrasonic sensor. The edge device uses an ESP8266 as core microcontroller, that has task to manage the sensor's data, and data synchronize to the cloud server through Wi-Fi protocol. In the cloud server was also installing the Redis no SQL databases. Redis acts as application programming interface (API) that has task to handle the publish and subscribe request from client devices. And the last part is a monitoring system that is composed of desktop application based-on Java programming languages, and MySQL databases. The monitoring system is a user interface that has the function as the main interface to control and monitor every edge device. The SQL databases were installed in local database that has a task to store every event, when the edge device detects an intruder.

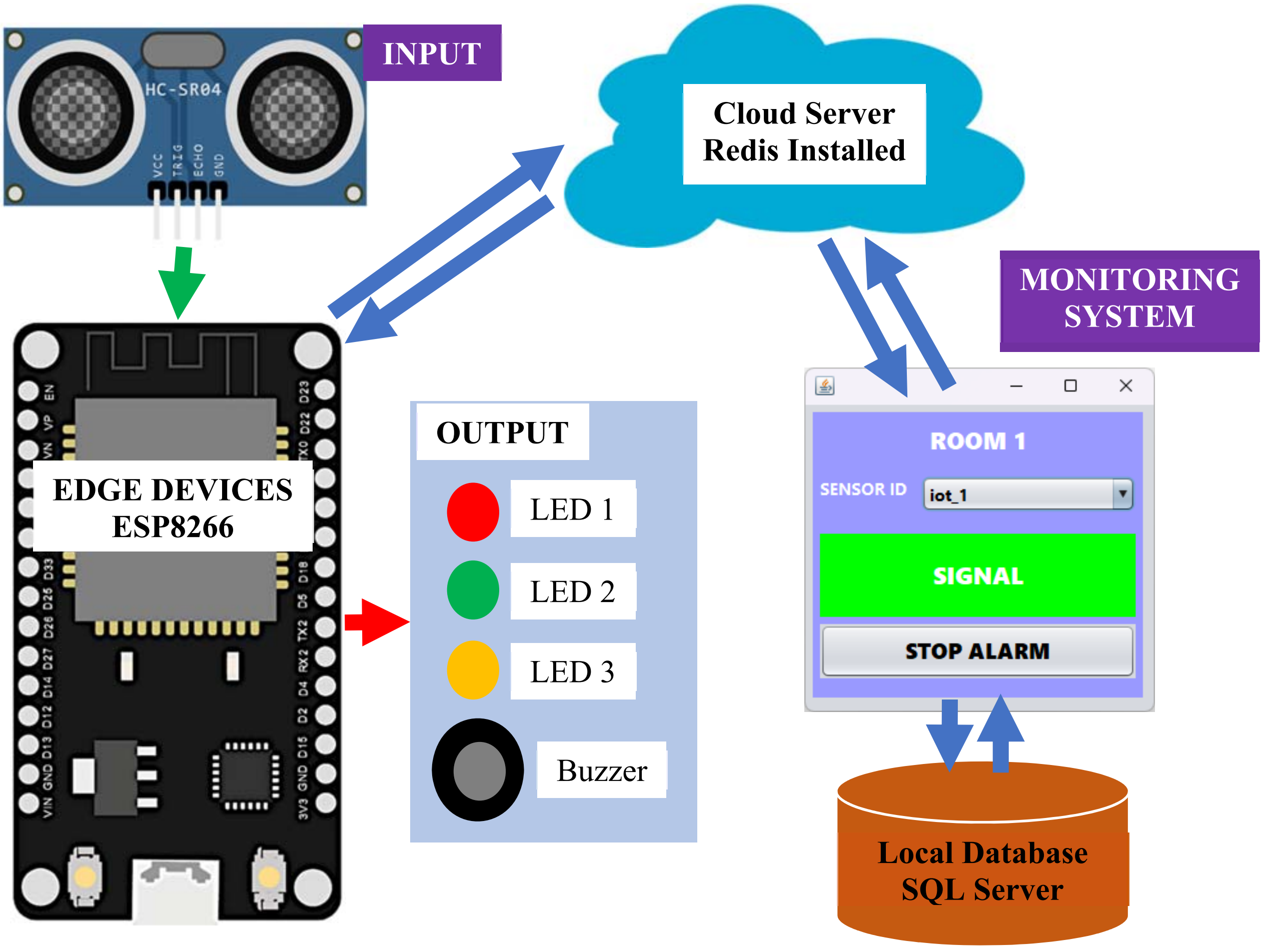


Fig. 1. Overall diagram of low-cost security monitoring system by using an ultrasonic sensor.

3.1. *Input*

The input uses an ultrasonic sensor that is used to measure the distance. The series of ultrasonic sensors is HC-SR04. This sensor was equipped with 4 pins which consisted of Voltage Common Collector (VCC), trigger (TRIG), ECHO and ground (GND). The HC-SR04 is an ultrasonic sensor that generally can be used to measure distance without contact to object in a range minimum between 2 cm (centimeters) to a maximum distance of 400 cm [21]. The sensor emitted the sound with a frequency greater than 20 Kilo Hertz (KHz), where the sound can't be heard by humans. The sensor is equipped with a pair of ultrasonic transducers. The one transducer acts as a transmitter that converts the electrical signal produced by microcontroller devices into ultrasonic sounds that work on a 40 KHz frequency. The sensor works well on a voltage of 5 volts with a current is 15 mA (Milliampere) for operation [22]. The ultrasonic sensor is connected directly to Node Micro Controller Unit (MCU) ESP8266 with the detail of the pin's attachment can be seen in Table 1. As a power supply input pin (VCC) for HC-SR04 is obtained from the 3v3 ESP8266's pin. GND of HC-SR04 pin connected to ESP8266's GND. TRIG's pin of HC-SR04 connected to General Purposes Input Output (GPIO) pin 12 on ESP8266, and the ECHO's pin of HC-SR04 connected to pin 14 on ESP8266.

Table 1. Pin's attachment between ultrasonic sensor with Node MCU ESP8266.

Ultrasonic Sensor HC-SR04 Pins	Node MCU ESP8266 Pins
VCC	3v3
GND	GND
TRIG	GPIO pin 12
ECHO	GPIO pin 14

3.2. *Edge devices by ESP8266*

The ESP8266 is a microcontroller board that is equipped with a wireless communication network based on 802.11 b/g/n Wi-Fi protocols. The power work for ESP8266 is between 3.0 volt to 3.6-volt direct current (DC). The ESP8266 is designed for mobile project purposes, so the ESP8266 still can operate well between 56 milli Ampere (mA) to 170 mA in conditions connected to the access point. This controller board has been used in many projects, for example for home control automation [23][24], smart grid system to support energy harvesting [25][26][27], smart medicine [28][29], and other interesting projects.

3.3. *Output*

The output devices that are connected directly to ESP8266 are light emitting diode (LED) 1, LED 2, LED 3, and buzzer speaker. LED 1 is a red color. LED 2 is a green color, and LED 3 is a yellow color. LED 1 functions as an indicator that will turn light red when the ESP8266 doesn't connect to the monitor system. LED 1 is attached to GPIO port 5 on the ESP8266. LED 2 will turn green light if the ESP8266 is connected to monitor system apps. LED 2 is connected to GPIO port number of 4 on the ESP8266. LED 3 will turn to a blinking yellow light if the ultrasonic sensor of HC-SR04 detects an object that is approaching the sensor. LED 3 is connected to GPIO pin number of 15 on the ESP8266, and the buzzer speaker attached to the ESP8266 as an additional audio indicator will turn on when the object (intruder) catches approach the sensor. The buzzer speaker is connected to GPIO pin number 13 on the ESP8266. The detail of every pinpoint connection between Node MCU ESP8266 and several output devices can be seen in Table 2. All the output devices that are connected to Node MCU ESP8266 use a common cathode or connecting the cathode pin to ground terminal.

Table 2. Pin's attachment between Node MCU ESP8266 with several output devices.

Node MCU ESP8266 Pins	Output Devices
GPIO pin 5	LED 1 (Red)
GPIO pin 4	LED 2 (Green)
GPIO pin 15	LED 3 (Yellow)
GPIO pin 13	Buzzer Speaker

3.4. *Cloud server*

The cloud server is a concept for services that provides a computational resource, complete with static internet protocol so that the client devices can be connected or accessed to the server through the internet and an uninterrupted electricity guarantee that will keep the server always on [30]. Cloud servers also can be in a virtual service form [31]. The cloud server will be an intermediary of data transfer from ESP8266 to monitoring systems and vice versa. So that the cloud server can be used to central intermediary data, the cloud server needs to install Redis. Redis is a no SQL database that is commonly used to cache the data from the SQL database and then loaded into RAM (random access memory). Redis also can be used as a message broker that has the ability to publish and subscribe the JavaScript Object Notation (JSON) data between edge devices to the server or vice versa be high-speed in less than 0.5 seconds [32]. The default port communication for Redis is 6379.

3.5. *Monitoring system*

There are so many programming languages that can be used to develop programs, especially to build an application based on desktop environment [20]. One of them is a Java programming language. Java programming can be done by using an integrated development environment (IDE) based on Apache NetBeans. The reason to use an Apache NetBeans for programming IDE is in Apache NetBeans, there is contains a compiler system like MAVEN was integrated that has tasks to give the convenience for compiling the Java application [33]. Fig. **2** is an example of a monitoring system designed to monitor 3 sensors, there are to monitor the sensor that located in room 1, room 2, and room 3.

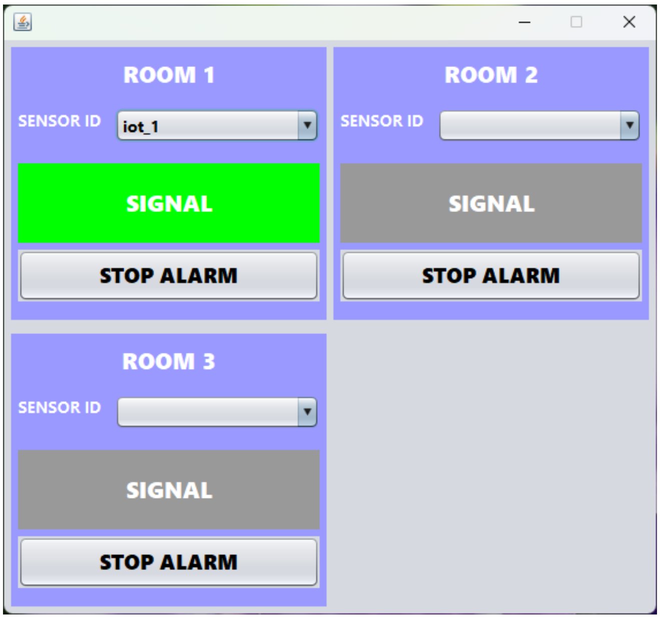


Fig. 2. Security monitoring system based on Java application.

3.6. *Local Database SQL Server*

In the database there are 4 tables. That is table of data_edge, data_status, data_detection, and data_stop_alarm. The attributes that are contained in data_edge table are pk_edge, id_label, and id_edge. Data_edge table is used to store the information related to the id_edge on every sensor device which has unique characteristics that associated with id_label on monitoring system that displayed in Fig. 2. The Table 2. Is the detail of databases attribute that used in this study. The attributes of pk_edge, pk_status, pk_detection, and pk_stop is used to store the primary key that is formatted as integer and filled by auto increment. Fig. 3 is the brief visualization of entity relationship diagram databases, which is a summary of the Table 3.

Table 3. Detail of attribute database that used in monitoring system.

Table Name	Attribute	Type of Data	Size of Data
data_edge	pk_edge	Auto increment of integer	-
	id_label	Varchar	25
	id_edge	Varchar	25
data_status	pk_status	Auto increment of integer	-
	fk_edge	Integer of foreign key from pk_edge's attribute from data_edge's table.	-
	status	Tiny integer	1
data_detection	pk_detection	Auto increment of integer	-
	fk_status	Integer of foreign key from pk_status's attribute from data_status's table.	-
	time	Time	-
	date	Date	-
	time_stamp	Double integer	-
	alert	Tiny integer	1
data_stop_alarm	pk_stop	Auto increment of integer	
	fk_detection	Integer of foreign key from pk_detection's attribute from data_detection's table.	-
	date	Date	-
	time	time	-

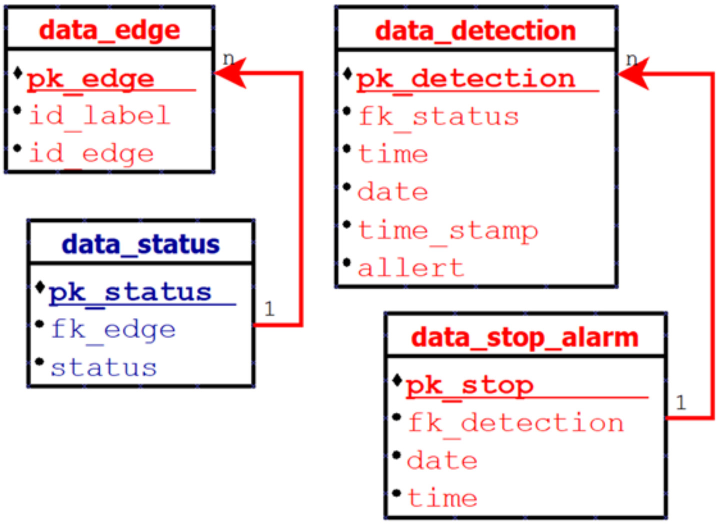


Fig. 3. Entity relationship diagram (ERD) of intruder detection databases.

3.7. *Sequence diagram*

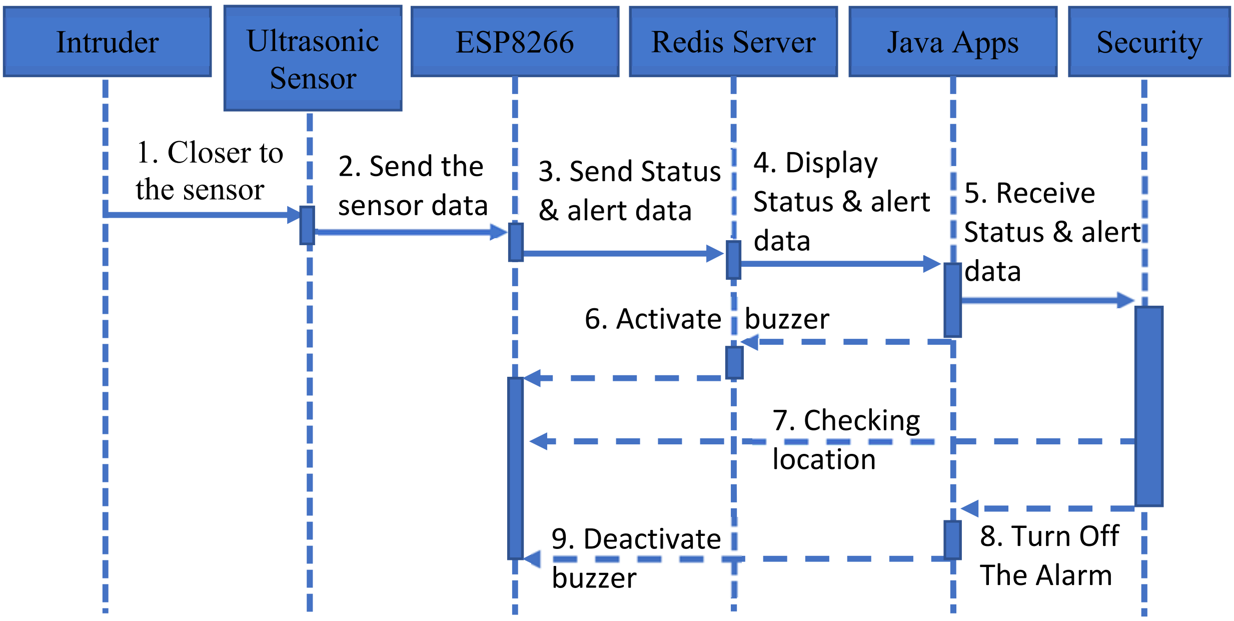


Fig. 4. Sequence diagram of the overall proposed method.

The sequence diagram is a visualization technique that is used to show the flow of how every actor and object works and communicates together [34]. Fig. 4 describes the overall sequence process of how the proposed model interact with each other state. The actors are intruders and security people. The ultrasonic sensor and ESP8266 module are integrated in one unit. The ESP8266 acts as edge device and Java apps acts as monitoring systems. The security person is always on guard in front of Java Apps. The ESP8266 always communicates with Java apps that are initiated by Redis Server. The communication between ESP8266 and Java Apps is asynchronous. The detail of how the sequence diagram talks, will be explained as follows:

3.7.1. Closer to the sensor

Is the condition of the intruder walking close to the ultrasonic sensor. The limit distance is set to maximum of 10 centimeters. When the intruder's position is near in range of fewer than 10 centimeters against to sensor, the system will activate an alert in the form of sound and visual both on the edge device of ESP8266 and monitoring system.

3.7.2. Send the sensor data

Basically, the ultrasonic sensor will send the data continuously to the ESP8266 microcontroller module through a wire cable connection.

3.7.3. Send status and alert data

When the intruder is detected by the ultrasonic sensor by distance, the ESP8266 module will send an alert notification to the cloud server that is handled by Redis Server. The connection between Node MCU ESP8266 with Redis server is two-way communication. The Redis server works as central communication that bridges data exchange between the ESP8266 with monitoring system. All the communication data between ESP8266 with the monitoring system is wrapped in JavaScript Object Notation (JSON)'s format.

3.7.4. Display status and alert data

ESP8266 will publish the data continuously to the Redis server. The data consists of life status and alert notifications. The monitoring system based on the Java application will subscribe to the data information which is on the Redis server. When no intruder is in front of the sensors the alert data is set to 0. And the live status of ESP8266 will send the number 1 to indicate the ESP8266 is live.

3.7.5. Receive status and alert data

The monitoring system is programmed using a Java application. The app is used to monitor the condition of every sensor. The monitoring system is monitored manually by security people. The monitoring system will continuously subscribe to the data from the Redis server. The data consists of life status, distance, date, time, and time stamp from every node sensor. In the scenario the node sensor detects the intruder, the monitoring system will display the warning notification on the monitor computer, so the security personnel will check the room directly to see that there is an intruder identified.

3.7.6. Activate buzzer

When the intruder is monitored by the monitoring system, due to approaching the sensor, the monitoring system will send a command of signal alert to activate a buzzer speaker that attached in ESP8266. The alert data is wrapped in JSON format and sent to the Redis server. Be continuously the ESP8266 not only publish the several data like time and distance, but also the ESP8266 actively subscribe the alert information that sent by monitoring system to Redis server.

3.7.7. Checking Location

When the alarm system is on, the security personnel will check the point location to ensure that condition where the sensor is located. After that the security personnel can turn off the alarm notification by clicking the button which is on the monitoring system.

3.7.8. Turn off the alarm

The user or security personnel have access to turn off the alarm system, by clicking the button that provides in the java monitoring system application. Immediately the monitoring systems will send the signal data to turn off the speaker buzzer on Node MCU ESP8266. All the data transmission is bridged by the Redis server.

3.7.9. Deactivate buzzer

If the monitoring system is in an alert mode, and the stop alarm's button is pressed, the monitoring system will send a command to turn off the buzzer speaker.

4. Experiment

4.1. *Prepare the server*

In this experiment the server that was used has specifications as follows; the processor uses intel Xeon CPU E5-2695 v2 @ 2.4 GHZ, The RAM is 1 GB. The operating system that is used was Ubuntu 20.04.5 LTS with the Linux Kernel version is 3.10.0-1160.53.1.vz7.185.3. of Architecture x86-64. In the server was also installed the Redis application to handle the no SQL databases of client server services.

4.2. *Built an edge devices*

Several third-party libraries for ESP8266 were installed into the Arduino integrated development environment (IDE), i.e., ArduinoJson.h, NTPClient.h, and Redis.h. ArduinoJson.h is the library that can be used to wrap several variables to be JSON formatted with key and value. After the data that composed of variable and value wrapped into JSON formatted, after that the data can be sent to other devices through wire able or wireless communication schema. The NTPClient.h was used to get the information about the date and time by real-time from internet directly. Redis.h is the library that contains some rules to communicate with Redis server, both for publish nor subscribe. Fig. 5 is the wire installation of buzzer speaker, two light emitting diodes, and one unit of ultrasonic sensor that paired with ESP8266 module.

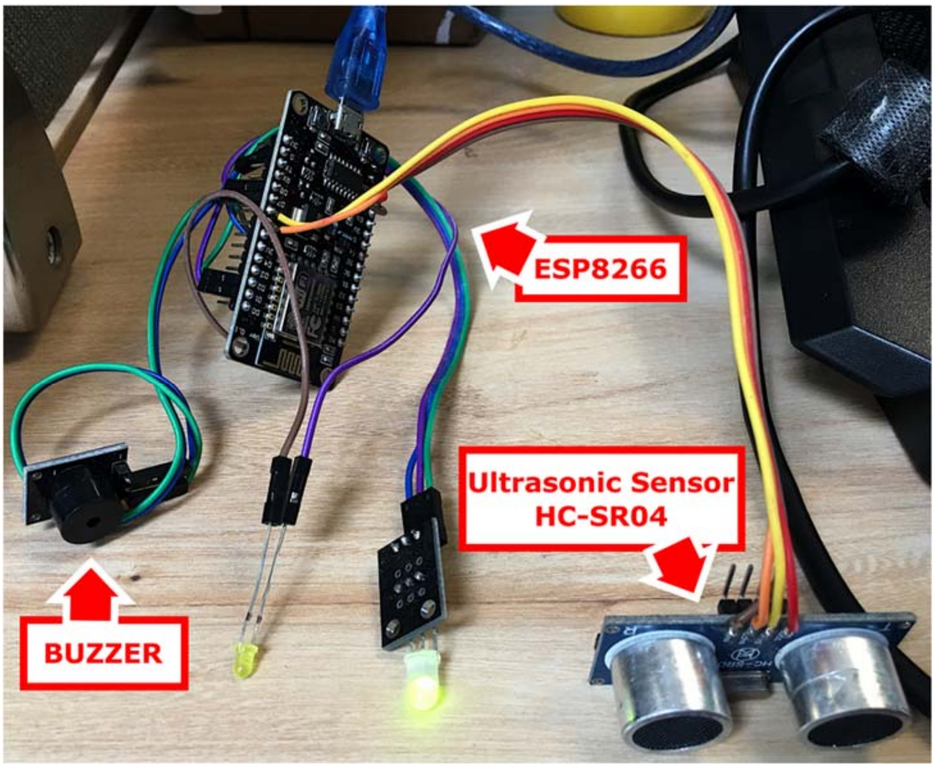


Fig. 5. Edge devices are based on ESP8266, equipped with ultrasonic sensor of HC-SR04 to measure the distance and speaker buzzer as an output in form of sound.

4.3. *Built The Monitoring System*

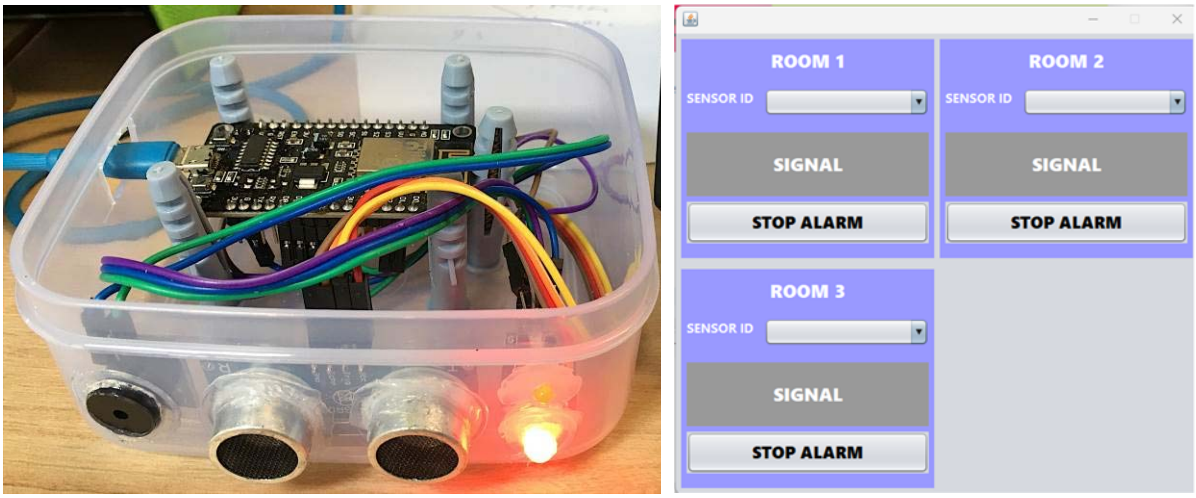
The java compilers that used in this experiment was Apache NetBeans IDE 17. The reason of NetBeans IDE was used in this experiment is because the ease of use especially in use of JFrame Form [35]. The NetBeans provide convenience to do drag and drop activity for swing containers, swing controls and other swing menus in design mode.

5. Result and Discussion

This study the edge device that was used for experimental is one device. On the monitoring system program, the threshold distance of edge devices was set to 25 centimeters (cm). So, when the intruder approach to edge device's sensor less than 25 cm, the monitor system will save the incident in the form of date, time, and time stamp to the MySQL databases. The total experiment was done as many 10 times to simulate the intruder approach to the sensor.

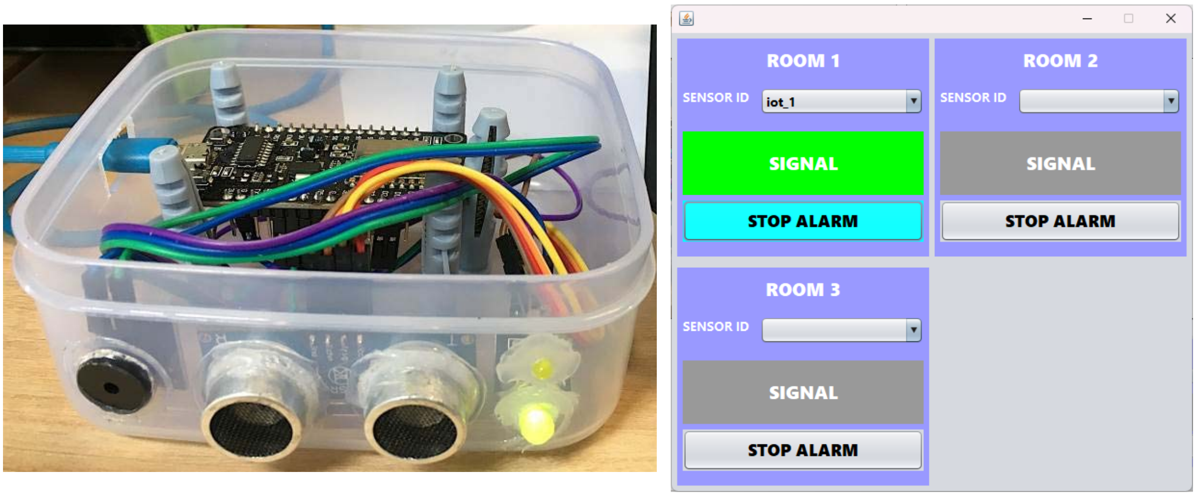
The Fig. 6 a., Fig. 7 a., and Fig. 8 a. are the edge devices that derive from the Fig. 5 packed into one box. The Fig. 6 a. is a condition when the edge devices not yet associated with monitoring system that displayed in Fig. 6 b., so the indicator light emitting diode lighting red.

Edge device that displayed in Fig. 7. a., its light emitting diode lighting green because the device has registered in monitoring system that displayed in Fig. 7 b. When the edge devices turned on, by automatically every edge device will send the number sensor identification to the monitoring systems. Fig. 7 b. is an example of the monitoring system that has registered to edge devices, which have the sensor id containing iot_1 value. After the sensor id registered to monitoring system, the signal's label on monitoring system will lighting green. The Fig. 8 a. and Fig. 8 b. is the condition when the edge devices detect an intruder. The red label warning will turn on give the information to user that there is something that approaches the sensor. The Fig. 8 is used to measure the duration between the edge device detect the intruder and send the data to monitoring system application, and the result presented in Table 4.



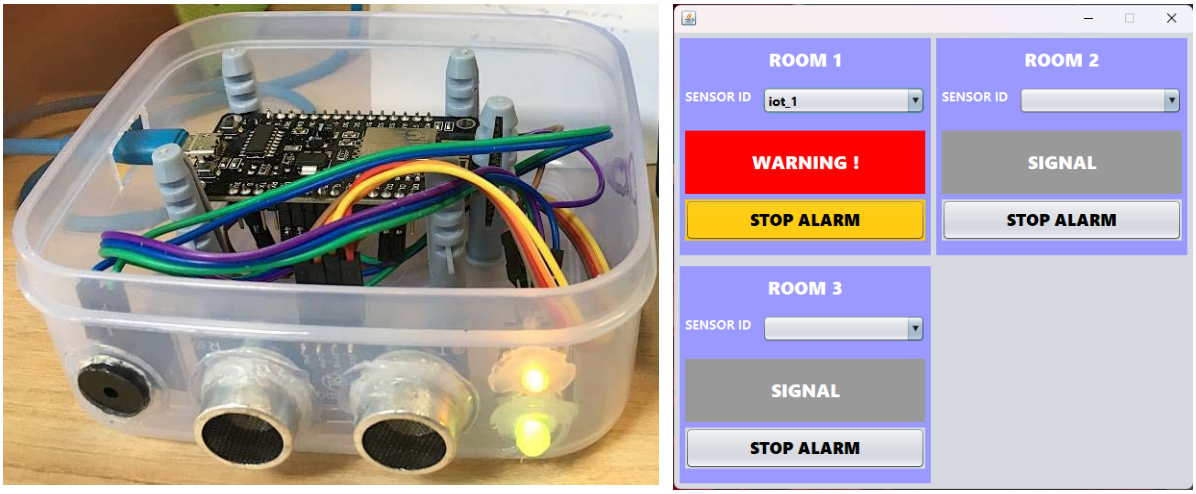
a b

Fig. 6. a. The edge device not registered to monitoring system yet indicated by light emitting diode lighting red, b. The monitoring system is not registered to the edge device yet.



a b

Fig. 7. a. The edge device condition connected and monitored by the monitoring system indicated the light emitting diode lighting green, b. The monitoring system was registered to the edge devices.



a b

Fig. 8. a. The edge device detects an intruder, b. The monitoring system receives a signal from the edge devices that suspect the existence of an intruder.

As shown in Table 4, there are some columns that consist of pk_detection, date, time, and time_stamp. The pk_detection is used to store the primary key in the database. The date and time column are used to store the time of when the edge devices detect the intruder, and the time_stamp's column is used to store how long the edge devices process the presence detection and send the data to the Redis cloud server. In the 10 times experiments to detect the intruder, the average time stamp obtained 27479,8 microseconds.

Table 4. The data that is obtained when the edge devices detect the presence of the intruder.

pk_detection	date	time	time_stamp
11	2023-06-22	21:09:05	21548 microseconds
12	2023-06-22	21:09:16	22258 microseconds
13	2023-06-22	21:09:25	45498 microseconds
14	2023-06-22	21:09:37	24197 microseconds
15	2023-06-22	21:09:47	24277 microseconds
16	2023-06-22	21:09:59	21736 microseconds
17	2023-06-22	21:10:18	34167 microseconds
18	2023-06-22	21:10:30	21445 microseconds
19	2023-06-22	21:10:47	37214 microseconds
20	2023-06-22	21:11:00	22458 microseconds
Time stamp average			**27479,8 microseconds**

The final stage of this study is compared to the previous research that use a webcam as a sensor, Raspberry as an edge device, and Yolo for the core algorithms to detect the intruder that has been proposed before [11]. The comparison target is the time required to detect an intruder. As a result, the method that is presented in this study gives a satisfactory result, especially in speedy time. The time to detect an intruder in this study is 27479,8 microseconds, faster than the previous method which takes a time of 49.673 seconds [11]. The table of comparison that has been narrated in detail can be seen in Table 5.

Table 5. Comparison time performance with identified study.

Study Identification	Sensor Devices	Edge Devices	Algorithm	Notification	Speed Time Average
Computer vision method [11]	Webcam	Raspberry Pi 3B+	Yolo V3	Telegram	49.673 seconds
Proposed method	**Ultrasonic HC-SR04**	**Node MCU ESP8266**	**Threshold distance**	**Buzzer and Monitoring system remotely based on Java application.**	**27.479,8 microseconds**

6. Conclusion

In this study, the low cost of the monitoring system has been explained. Several architectures were also included. They are grouped into three sections. The edge device is based on an ESP8266 microcontroller that is equipped with an ultrasonic sensor of HC-SR04 to detect an intruder. On the middle architecture, the no SQL databases of Redis that were installed on the cloud server were also used. For the last part that acts as a user interface, the application based on the desktop that was built by using Java programming language was also proposed. The model proposed in this study works to measure the distance change that happened in front of the ultrasonic sensor as a result that an intruder comes closer. In this study it is possible to add more edge devices to be placed at several points that want to be monitored. Between the edge devices and monitoring system connect each other mediated by Redis API. For evaluation, the 10 times of simulations attempted to measure the times required from the edge devices to send the signal to the monitoring system. The satisfactory result shows that for detecting an intruder, it just takes 27479,8 microseconds times faster than by using Raspberry Pi equipped with a web camera sensor that has been proposed by other researchers before.

Acknowledgments

This research was shaded and funded by Bina Nusantara University, where the research collaborates with a building management team in 2023 with the contract number is KI004007.

Conflict of Interest Statement:

The authors declare that all the research that was conducted in this study has no conflict of interest.

References

[1] Z. N. Mohammad, F. Farha, A. O. M. Abuassba, S. Yang, and F. Zhou, "Access control and authorization in smart homes: A survey," *Tsinghua Sci Technol*, vol. 26, no. 6, pp. 906–917, 2021, doi: 10.26599/TST.2021.9010001.

[2] H. Sequeiros, T. Oliveira, and M. A. Thomas, "The Impact of IoT Smart Home Services on Psychological Well-Being," *Information Systems Frontiers*, vol. 24, no. 3, pp. 1009–1026, 2022, doi: 10.1007/s10796-021-10118-8.

[3] S. P. Chatrati *et al.*, “Smart home health monitoring system for predicting type 2 diabetes and hypertension,” *Journal of King Saud University - Computer and Information Sciences*, vol. 34, no. 3, pp. 862–870, 2022, doi: 10.1016/j.jksuci.2020.01.010.

[4] P. Pirzada, A. Wilde, G. H. Doherty, and D. Harris-Birtill, “Ethics and acceptance of smart homes for older adults,” *Inform Health Soc Care*, vol. 47, no. 1, pp. 10–37, 2022, doi: 10.1080/17538157.2021.1923500.

[5] H. Hu *et al.*, “Assembling Hollow Cactus-Like ZnO Nanorods with Dipole-Modified Graphene Nanosheets for Practical Room-Temperature Formaldehyde Sensing,” *ACS Appl Mater Interfaces*, vol. 14, no. 11, pp. 13186–13195, 2022, doi: 10.1021/acsami.1c20680.

[6] A. Q. H. Badar and A. Anvari-Moghaddam, “Smart home energy management system–a review,” *Advances in Building Energy Research*, vol. 16, no. 1, pp. 118–143, 2022, doi: 10.1080/17512549.2020.1806925.

[7] J. Guntur, S. S. Raju, T. Niranjan, S. K. Kilaru, R. Dronavalli, and N. S. S. Kumar, “IoT-Enhanced Smart Door Locking System with Security,” *SN Comput Sci*, vol. 4, no. 2, 2023, doi: 10.1007/s42979-022-01641-9.

[8] B. Lahasan and H. Samma, “Optimized Deep Autoencoder Model for Internet of Things Intruder Detection,” *IEEE Access*, vol. 10, pp. 8434–8448, 2022, doi: 10.1109/ACCESS.2022.3144208.

[9] P. De, A. Chatterjee, and A. Rakshit, “PIR-Sensor-Based Surveillance Tool for Intruder Detection in Secured Environment: A Label-Consistency-Based Modified Sequential Dictionary Learning Approach,” *IEEE Internet Things J*, vol. 9, no. 20, pp. 20458–20466, 2022, doi: 10.1109/JIOT.2022.3178160.

[10] O. Taiwo and A. E. Ezugwu, “Internet of Things-Based Intelligent Smart Home Control System,” *Security and Communication Networks*, vol. 2021, 2021, doi: 10.1155/2021/9928254.

[11] N. Surantha, I. M. P. Mertha, J. A. Widjaja, and P. D. Andersen, “SMART HOME SECURITY SYSTEM FOR INTRUDER DETECTION USING YOLO V3 ALGORITHM,” *ICIC Express Letters, Part B: Applications*, vol. 14, no. 3, pp. 295–302, 2023, doi: 10.24507/icicelb.14.03.295.

[12] E. Schiller, A. Aidoo, J. Fuhrer, J. Stahl, M. Ziörjen, and B. Stiller, “Landscape of IoT security,” *Comput Sci Rev*, vol. 44, 2022, doi: 10.1016/j.cosrev.2022.100467.

[13] D. J. Atul, R. Kamalraj, G. Ramesh, K. Sakthidasan Sankaran, S. Sharma, and S. Khasim, “A machine learning based IoT for providing an intrusion detection system for security,” *Microprocess Microsyst*, vol. 82, 2021, doi: 10.1016/j.micpro.2020.103741.

[14] M. Abdel-Basset, H. Hawash, V. Chang, R. K. Chakrabortty, and M. Ryan, “Deep Learning for Heterogeneous Human Activity Recognition in Complex IoT Applications,” *IEEE Internet Things J*, vol. 9, no. 8, pp. 5653–5665, 2022, doi: 10.1109/JIOT.2020.3038416.

[15] R. Yadav, I. Sreedevi, and D. Gupta, “Augmentation in performance and security of WSNs for IoT applications using feature selection and classification techniques,” *Alexandria Engineering Journal*, vol. 65, pp. 461–473, 2023, doi: 10.1016/j.aej.2022.10.033.

[16] S. Achar, N. Faruqui, M. Whaiduzzaman, A. Awajan, and M. Alazab, “Cyber-Physical System Security Based on Human Activity Recognition through IoT Cloud Computing,” *Electronics (Switzerland)*, vol. 12, no. 8, 2023, doi: 10.3390/electronics12081892.

[17] M. Alzaidi and A. Vagner, “Benchmarking Redis and HBase NoSQL Databases using Yahoo Cloud Service Benchmarking tool,” *Annales Mathematicae et Informaticae*, vol. 56, pp. 1–9, 2022, doi: 10.33039/ami.2022.12.006.

[18] F. Corno and L. Mannella, “Security Evaluation of Arduino Projects Developed by Hobbyist IoT Programmers,” *Sensors*, vol. 23, no. 5, 2023, doi: 10.3390/s23052740.

[19] H. Taherdoost, “Security and Internet of Things: Benefits, Challenges, and Future Perspectives,” *Electronics (Switzerland)*, vol. 12, no. 8, 2023, doi: 10.3390/electronics12081901.

[20] C. Cubukcu, Z. B. G. Aydin, and R. Samli, “Comparison of C, Java, Python and Matlab Programming Languages for Fibonacci and Towers of Hanoi Algorithm Applications,” *Boletim da Sociedade Paranaense de Matematica*, vol. 41, 2023, doi: 10.5269/bspm.52209.

[21] J. Susilo, A. Febriani, U. Rahmalisa, and Y. Irawan, “Car parking distance controller using ultrasonic sensors based on arduino uno,” *Journal of Robotics and Control (JRC)*, vol. 2, no. 5, pp. 353–356, 2021, doi: 10.18196/jrc.25106.

[22] L. M. Last Minute, “How HC-SR04 Ultrasonic Sensor Works & Interface It With Arduino,” *https://lastminuteengineers.com/arduino-sr04-ultrasonic-sensor-tutorial/*, Jun. 11, 2022.

[23] O. Taiwo, A. E. Ezugwu, O. N. Oyelade, and M. S. Almutairi, “Enhanced Intelligent Smart Home Control and Security System Based on Deep Learning Model,” *Wirel Commun Mob Comput*, vol. 2022, 2022, doi: 10.1155/2022/9307961.

[24] P. Manojkumar *et al.*, “A novel home automation distributed server management system using Internet of Things,” *International Journal of Ambient Energy*, vol. 43, no. 1, pp. 5478–5483, 2022, doi: 10.1080/01430750.2021.1953590.

[25] E. García, N. Ponluisa, E. Quiles, R. Zotovic-Stanisic, and S. C. Gutiérrez, “Solar Panels String Predictive and Parametric Fault Diagnosis Using Low-Cost Sensors,” *Sensors*, vol. 22, no. 1, 2022, doi: 10.3390/s22010332.

[26] L. H. Fang and R. B. A. Rahim, “Design of Savonius model wind turbine for power catchment,” *International Journal of Electrical and Computer Engineering*, vol. 12, no. 3, pp. 2285–2299, 2022, doi: 10.11591/ijece.v12i3.pp2285-2299.

[27] H. Thakkar, P. V. Ramana, H. Panchal, and K. K. Sadasivuni, “Design and experimental analysis of solar-powered water desalination system using humidification–dehumidification,” *International Journal of Ambient Energy*, vol. 43, no. 1, pp. 3485–3496, 2022, doi: 10.1080/01430750.2020.1831602.

[28] K. Islam, F. Alam, A. I. Zahid, M. M. Khan, and M. Inamabbasi, “Internet of Things- (IoT-) Based Real-Time Vital Physiological Parameter Monitoring System for Remote Asthma Patients,” *Wirel Commun Mob Comput*, vol. 2022, 2022, doi: 10.1155/2022/1191434.

[29] K. V. S. S. Ganesh, S. P. S. Jeyanth, and A. R. Bevi, “IOT based portable heart rate and SpO2 pulse oximeter,” *HardwareX*, vol. 11, 2022, doi: 10.1016/j.ohx.2022.e00309.

[30] Y. Wang, D. Nörtershäuser, S. Le Masson, and J.-M. Menaud, “Potential effects on server power metering and modeling,” *Wireless Networks*, vol. 29, no. 3, pp. 1077–1084, 2023, doi: 10.1007/s11276-018-1882-1.

[31] M. S. Bali, K. Gupta, D. Gupta, G. Srivastava, S. Juneja, and A. Nauman, “An effective technique to schedule priority aware tasks to offload data on edge and cloud servers,” *Measurement: Sensors*, vol. 26, 2023, doi: 10.1016/j.measen.2023.100670.

[32] R. K. Singh and H. K. Verma, “Redis-Based Messaging Queue and Cache-Enabled Parallel Processing Social Media Analytics Framework,” *Computer journal*, vol. 65, no. 4, pp. 843–857, 2022, doi: 10.1093/comjnl/bxaa114.

[33] C. Macho, S. Beyer, S. McIntosh, and M. Pinzger, “The nature of build changes: An empirical study of Maven-based build systems,” *Empir Softw Eng*, vol. 26, no. 3, 2021, doi: 10.1007/s10664-020-09926-4.

[34] M. Shirole and R. Kumar, “Concurrent behavioral coverage criteria for sequence diagrams,” *Innov Syst Softw Eng*, vol. 19, no. 2, pp. 157 – 176, 2023, doi: 10.1007/s11334-021-00413-7.

[35] R. S. Silva and J. L. Sobral, “Efficient High-Level Programming in Plain Java,” *Int J Parallel Program*, vol. 51, no. 1, pp. 22 – 42, 2023, doi: 10.1007/s10766-022-00747-0.

Authors Profile

Yulianto, received a Bachelor of Computer Science degree from Dian Nuswantoro University (UDINUS) in 2017. During college at UDINUS university the author was interested in computer vision and the thesis was about human recognition based-on feature extraction of histograms of oriented gradients that were classified using a decision tree algorithm. After that, the author got a chance to follow the scholarship program at Bina Nusantara University (BINUS) and received a Master of Computer Science degree in the first year of 2022. His research covers electronic appliances, the internet of things, and computer science fields. The last research focuses on computer vision with specifically referred to super-resolution images based on generative adversarial networks algorithm. Now the author works at Bina Nusantara University as a lecturer and researcher. Anytime the author can be contacted by email at yulianto003@binus.ac.id or by whats app at 6285290553327 directly.

Sidharta, is a graduate of S1 Mathematics at the Sepuluh Nopember Institute of Technology Surabaya in 2006. His research is related to algorithms for optimization. His final project is the implementation of the Ant Colony Algorithm for shortest route search. In 2008 he took a Master’s in Management Technology with a specialization in Information Technology. Graduated in 2011 with research on information technology development portfolios at Telkom Surabaya Call Center. Currently conducting further study at the Doctor of Computer Science program at Bina Nusantara University. The focus of his research is deep learning for the detection and classification of building damage after natural disasters.

Hanugra Aulia Sidharta, is a PhD student at the Institut Sepuluh Nopember. He got the Bachelor of Engineering degree at Brawijaya University in 2008, he also got the Master of Technology Management degree at Institut Teknologi Sepuluh Nopember in 2013. His research covers several cross-cutting areas including Computer Networking, Devops, Internet of Things and Artificial Intelligence. Now he is concentrating on research in computer vision, in particular on research that uses the architecture of generative adversarial networks.

Danang Wahyu Wicaksono, is Lecturer at Bina Nusantara University in major of Bachelor of Computer Science. He got a Bachelor of Science degree at Institut Teknologi Sepuluh Nopember in 2014, after that he got the Master of Information System degree also at Teknologi Sepuluh Nopember in 2017. Now, in addition to his teaching activity, he is also a professional programmer, especially in web programming and mobile programming.