Journal of Engineering Science and Technology
Vol. 19, No. 4 (2024) 1156 – 1170

SERVER SECURITY AND AIR QUALITY MONITORING SYSTEM USING RASPBERRY PI 4 AND TELEGRAM

MOCHAMAD FAJAR WICAKSONO[1,*], MYRNA DWI RAHMATYA[2]

[1]Computer Engineering Department, Universitas Komputer Indonesia
[2]Management Informatics Department, Universitas Komputer Indonesia
Jalan Dipati Ukur No. 112-116, Bandung, 40126, Jawa Barat, Indonesia
*Corresponding Author: mfajarw@email.unikom.ac.id

Abstract

The purpose of this research is to develop a system that uses Raspberry Pi and IoT technologies to monitor server room conditions such as temperature, humidity, air quality, and access rights security. The contribution of this research is that it makes it easier for users to monitor the server room and take appropriate action when something goes wrong. The method utilized in this study is experimental. The output obtained is in the form of a web display and telegram notifications. The brain of this system is the Raspberry Pi 4. The input received is temperature, humidity, carbon monoxide levels, carbon dioxide, and dust particle values. The thing that causes the Raspberry Pi 4 to activate the buzzer and send notifications via Telegram is when the temperature value is more than 23^0C, the humidity is less than 45% or 60%, the measured carbon monoxide value is more than 50 ppm, or carbon dioxide is more than 800 ppm or the AQI value of dust particles is more than 100. Regarding access rights, users can only enter the server room using a registered RFID card. Access rights violations will be detected by PIR1, PIR2, and Ultrasonic sensors. If this happens, the Raspberry Pi will take a photo with the Pi Camera, activate the buzzer, and send it via Telegram. The entire system testing process is tested into two main parts: temperature, humidity, AQI testing, and security testing related to access rights according to the program scenario and initial objectives. Test results show that every part is functioning 100%. The overall test shows 100% success where the system can do actions according to program conditions and scenarios.

Keywords: AQI, Monitoring, Raspberry Pi4, Security, Server room, Telegram.

1. Introduction

The server room stores servers containing programs and databases. The server room must be kept secure because it functions as a datacenter for all company operations. Today, security can be defined as either physical security (as in physical access control) or logical security. Physical security is necessary to prevent physical attacks on server rooms. Abuse of privilege is a type of physical attack on the server room. This assault includes unlawful system access, such as physical breakthroughs and theft. Physical security necessitates processes and tactics to prevent physical attacks. A well-developed management plan can help avoid espionage and theft [1]. The prevention procedure can also decrease risk, identify the source of damage, and implement mitigation measures. Physical security implementation, as well as good rules and procedures, are equally critical as personnel knowledge [2].

Server rooms can be monitored over the Internet of Things (IoT). The Internet of Things enables data collection and sharing with other devices over the Internet. The Internet of Things (IoT) gathers, stores, and analyzes data [3]. Based on government regulations, the degree of humidity and temperature in the server room are among the most vital variables to monitor [4]. In 2017, the Indonesian government implemented temperature and humidity standards. The usual temperature is between 21 and 23 degrees Celsius, with humidity ranging from 45%-60%. High humidity can lead to corrosion and short circuits. Monitoring server room security involves more than just temperature and humidity, including defense systems. Aside from the foregoing, there is one aspect of the server room that must be addressed: air quality. This is important because dust particles might clog the ventilation system over time.

Plenty of research has been conducted to monitor the server room's status. In 2017, research remained focused on temperature and humidity issues, with no warning signs sent to users [5]. A study published in 2017 used fuzzy logic to construct clever air conditioners. However, it only focused on temperature [6].

Another study in 2018 was made. However, it only focuses on temperature and humidity [7]. A server room-related research was undertaken in 2018. This research focuses solely on securing the server room door using an application and a door lock [8]. Another research in 2018 was carried out. However, the alerts were still in the format of SMS [9].

In 2019, new research discovered that this server room may send notifications over Telegram. However, these studies focus primarily on the degree of humidity and temperature [10]. Even in 2019, there was a study on server room observation. However, this study solely looks at server room HVAC [11]. Apart from that, in the same year, there was research that focused on server rooms that measured temperature and humidity and used WhatsApp notifications. However, this research uses too many resources because it uses two controllers, while only one sensor is used [12].

In 2022, there are two other studies on server space. One of these studies focuses solely on the degree of humidity and temperature in the server rack [13]. Still, in 2022, another study was carried out and achieved its goal. The research monitored temperature, humidity, and server room access rights and used telegram notifications. However, this research did not measure the air quality in the server room, which also affects the server room [14].

Based on the explanation above, it can be said that each research focuses on one part only, for example, temperature and humidity only, or only dust particles or only temperature and humidity and access rights. So, it can be concluded that in general there has been no research that monitors server rooms regarding temperature, humidity, air quality, and access rights. Therefore, in this research, a system will be created that monitors server conditions regarding temperature, humidity, and air quality which includes levels of carbon monoxide, carbon dioxide, and dust particles, and regulates access rights to the server room.

In this research, Raspberry Pi 4 served as the main controller of the system. The control scope is divided into two main parts, namely the outside and the inside of the server room. Inside the server room data will be read for temperature, humidity, carbon monoxide levels, carbon dioxide, dust particles, ultrasonic sensor, and PIR sensor values. When the temperature value is more than 230C or the humidity is less than 45% or 60%, the Raspberry Pi will sound the buzzer and send a notification over Telegram. Regarding air quality, if the measured carbon monoxide value is more than 50 ppm carbon dioxide is more than 800 ppm, or the AQI value of dust particles is more than 100, the Raspberry Pi will sound the buzzer and send a notification over Telegram.

On the outside of the server room, there is an RFID reader, which functions to limit access rights to the server room. In other words, users can only enter the server room using a registered RFID card. Primarily, access rights violations will be detected by PIR1, PIR2, and Ultrasonic sensors. The ultrasonic sensor is installed near the entrance, while the PIR1 and PIR2 sensors are installed opposite each other at the top of the entrance. If the valid status of the RFID is not yet active, but the ultrasonic sensor detects a distance of less than 300cm or the PIR1 or PIR2 sensors detect movement, then the Raspberry Pi will immediately take a photo with the Pi Camera, activate the buzzer and send it via Telegram.

2. Methods

The experimental method was used to create the system in this study. Many experiments were conducted during its implementation, all of which naturally adhered to the fundamental theories underlying this study, which finally sought to produce outcomes that met the original goals. Experimental method consists of study of literature, design, simulation, implementation, and testing as shown in Fig. 1.

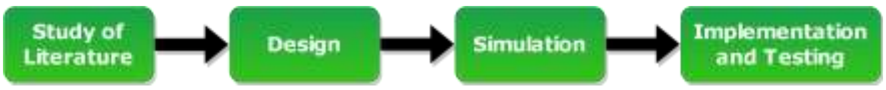

Fig. 1. The flow of the experimental method in this research.

2.1. Study of literature

At this stage, a literature study related to monitoring server security and air quality is used to obtain comparisons and research gaps with previous research. Besides, at this stage, a literature search is conducted to get a list of appropriate hardware and software needed to build this system.

2.2. Design

This stage creates and produces hardware and software design with the supporting theory produced in the previous phase. In addition, this stage also produces a design of sensor placement as well as actuators. The design of the hardware part consists of block diagrams and electronic schemes. The software part design covers the design of algorithms to implement this system. Figure 2 shows the shape of the block diagram produced at this step.

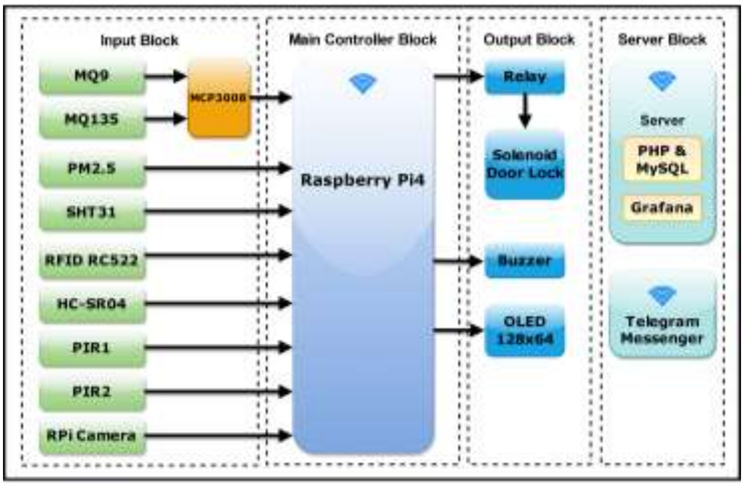

Fig. 2. Server security and air quality monitoring block diagram.

This mini-PC is primarily built with a Broadcom BCM2711 and 4GB RAM. Regarding peripherals, the Raspberry Pi 4 has 40 GPIO pins, a CSI port for the camera, 4 USB ports, a LAN port, two micro-HDMI ports, and an audio port [15, 16]. The operating system for the Raspberry Pi 4 used in this research is Raspbian OS [17, 18]. In the input block, there are several sensors and cameras. The MQ-9 sensor module is a sensor that can be used to read carbon monoxide levels [19]. To operate, this sensor module requires a 5V DC supply power [20]. The output read by the Raspberry Pi4 from this module is analog data. Apart from being related to air quality in the server room, in this research, the MQ-9 is beneficial for detecting fires and triggering warnings for server room managers. The ppm value ranges of carbon monoxide are shown in Table 1 [20].

Table 1. Carbon monoxide ppm range.

Carbon monoxide (ppm)	Category
≤ 50	Good
51 - 100	Medium
101 - 199	Unhealthy
200 - 299	Very Unhealthy
> 300	Harmful

Furthermore, measurements of carbon dioxide levels can be carried out using the MQ-135 sensor module where the output from this sensor is in the form of analog data [21-24]. The recommended ppm value range for carbon dioxide in rooms is ≤ 1000 ppm [25]. So, when the ppm value read exceeds this limit, Raspberry Pi 4 will send an alert to the server room manager via the Telegram messaging application. In the next section, there is the PM2.5 module. In this research, the sensor module used is PMS5003 which functions to measure air quality related to dust particles [26-29]. The reason for using and measuring this is because the air quality of the server room must be considered and the server room must be free from dust. The AQI parameters have been formulated by the United States Environmental Protection Agency in 2024 [30]. The output obtained from this sensor module is in the form of AQI values. In this research, when the AQI value is greater than 100, Raspberry Pi 4 will send an alert to the server room manager. The AQI value provisions are shown in Table 2.

Table 2. AQI Range for Particle Pollution

AQI Range	Category Breakpoints (μg/m^3)
Good (0- 50)	0.0 - 9.0
Moderate (51-100)	9.1 - 35.4
Unhealthy for sensitive group (101-150)	35.5 - 55.4
Unhealthy (151-200)	55.5 - 125.4
Very Unhealthy (201-300)	125.5 - 255.4
Hazardous (>301)	> 225.5

The next part is the SHT31 sensor module. This sensor module can be used to read temperature and humidity data [31]. The temperature range that can be measured on this module is -400C - 900C with an accuracy of ±0.3%. Meanwhile, for humidity, the range is between 0-100% with an accuracy of ±2% [32].

This involves the RFID module, Ultrasonic Sensor, PIR Sensor, Solenoid door lock, and Pi Camera regarding server room access rights. In this research, the MFRC5-22 RFID module was used. This module works with a frequency of 13.56MHz with a reading range of between 3cm-5cm [33-35]. Data can be read from this module using SPI communication [36-38]. When a tag is recognized, a valid status will be obtained then the Raspberry Pi 4 will deactivate the solenoid and measurement status from ultrasonic, and the PIR sensor will be ignored. The ultrasonic sensor used is HC-SR04. This sensor reads distance data with a measurement range of between 2cm-400cm [39, 40]. The pins used are the echo pin and trigger pin. The data read and converted by the Raspberry Pi 4 comes from the echo pin [41, 42]. This sensor is placed at the edge of the entrance door, so the measured distance will decrease when the door opens. The PIR sensor in this research functions to detect human movement and presence in the server room [43-45]. This research employed two PIR sensors. The type of PIR sensor utilized is HC-SR501. This sensor can operate when powered by 5V DC, and its output is a digital value [46, 47]. These two sensors are installed opposite each other at the top of the entrance. When there is a violation of access rights and detects human presence, the output value will be HIGH [48].

The next part is the RPi Camera which functions to take pictures when access rights are violated. The Pi Camera used is a Pi Camera V2 with an 8-megapixel image sensor [49]. This module was built using Sony IMX219 [50]. In the output section, there is a 128*64 OLED which functions to display the validity status when the user tags the RFID. The module used is the SSD1306 model with dimensions of 0.96 inches with I2C communication [51-53]. In the next block section is the server block which was built using the PHP and MySQL programming languages.

The reason for choosing these two parts is because PHP includes server-side scripting and can also be used as a backend system and MySQL is easy to integrate with PHP [54, 55]. All sensor data is sent and stored in a database. This storage of course uses MySQL commands. To visualize all the data, Grafana was used. Grafana is an open-source application that offers a simple interface for creating configurable and interactive dashboards [56, 57].

After creating an appropriate block diagram, the following step is to build the electronic circuit for this system. Figure 3 depicts the completed system's electronic circuit. From Fig. 3 it can be seen that several special communications are used between several sensors and the Raspberry Pi 4, such as Serial, SPI, and I2C communications. Serial communication is used to read data from the PMS5003 module. SPI communication is used to read data from the RFID RC522 module and analog data via the MCP3008 module. I2C communication is used for the OLED module and SHT31 sensor. the other part only uses the Raspberry GPIO directly because it does not require special communication to read the data. After creating the electronic circuit for this system, the next step is to create a software design for this system both regarding the choice of programming language and the program flow.

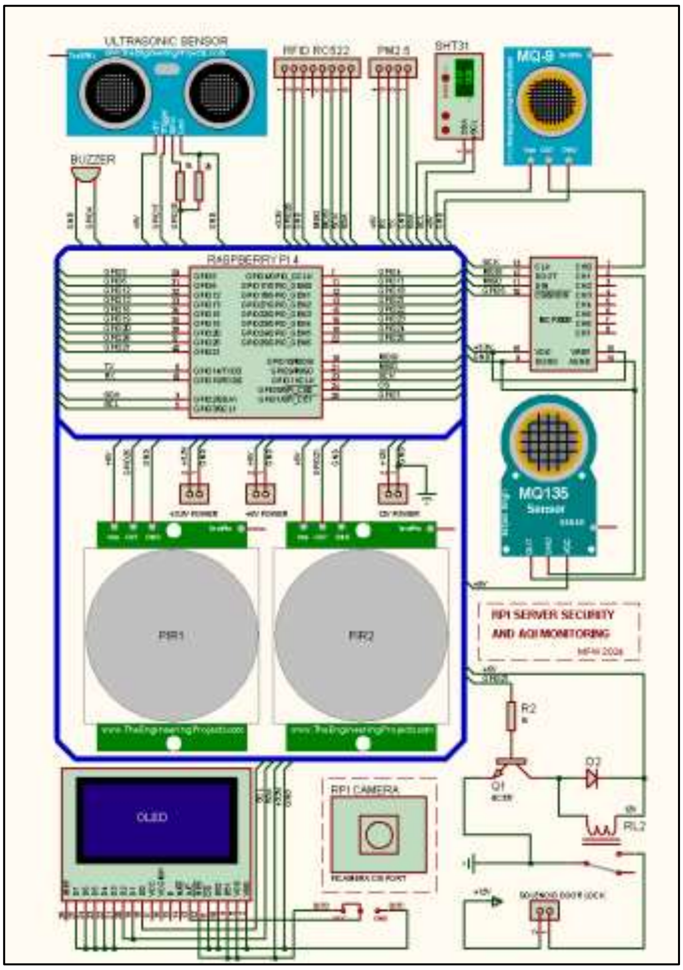

Fig. 3. Electronic circuit of server security and air quality.

In this research, the programming language used is Python. The reason for choosing this programming language is because Python is an open-source programming language and has many supporting libraries needed to build this system. The program flow shown below has been built as a text algorithm:

1. Begin.
2. I/O initialization.
3. Read RFID RC522.
4. if code number recognized then go to step 5.
5. Access granted and show recognized string on OLED.
6. Deactivate solenoid door lock and go to step 7.
7. If code number not recognized, then:
    a. Read PIR1 sensor, PIR2 sensor, Ultrasonic sensor.
    b. If distance from Ultrasonic sensor < 300 then go to step 8.
    c. If PIR1 sensor True then go to step 8.
    d. If PIR2 sensor True then go to step 8.
    e. Read MQ-9 (carbon monoxide), MQ-135 (carbon dioxide), SHT31 (Temperature and Humidity), PM2.5 (Dust particles in AQI).
    f. Send all sensor data to the server using POST method and display it on the Grafana dashboard.
    g. If temp > 23 then go to step 9.
    h. If humidity < 45% and >60% then go to step 9.
    i. If carbon monoxide ppm > 50 ppm then go to step 9.
    j. If carbon dioxide ppm > 800 ppm then go to step 9.
    k. If dust AQI > 100 then go to step 9.
    l. Go to step 3.
8. Access right violation procedure:
    a. Sound the buzzer.
    b. Capture photo using Pi Camera and send alert message via Telegram.
    c. Back to step 3
9. Sensor value out of range procedure:
    a. Sound the buzzer and send appropriate alert messages via Telegram.
    b. Back to step 3.
10. Go to step 3.

The next step is to design the position of all components of this system. The placement design that has been made is shown in Fig. 4.

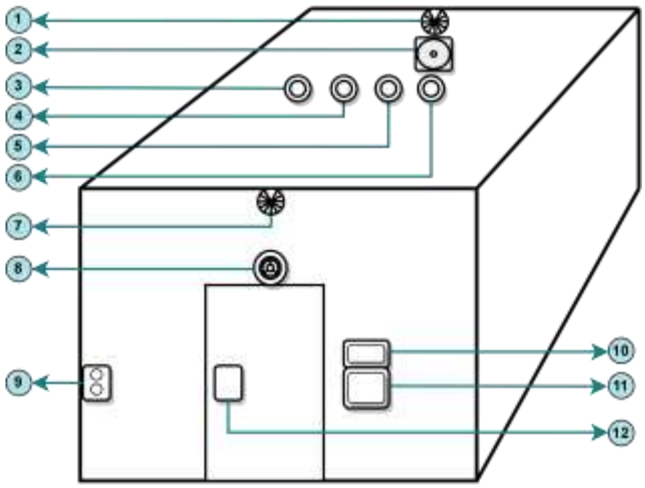

Fig. 4. Design of entire component placement.

Each of the numbers in Fig. 4 is described below:

1. PIR1 Sensor
2. Raspberry Pi 4 and Pi Camera
3. MQ-9 Sensor
4. MQ-135 Sensor
5. PM2.5 Sensor
6. SHT31 Sensor
7. PIR2 Sensor
8. Buzzer
9. Ultrasonic Sensor
10. OLED 128*64
11. RFID RC522
12. Solenoid Door Lock

2.3. Simulation

At this stage, various simulations are carried out regarding hardware and software functions according to the design in the previous stage. This simulation includes program verification per block as well as the electronic circuit that has been created.

2.4. Implementation and testing

At the implementation and testing stage, testing of the entire system is carried out. The tests carried out were following the program scenario and objectives of this research.

3. Results and Discussion

In this section, several results related to the tests that have been carried out are presented. This testing is divided into three main parts which include testing data storage in the database and display on the system dashboard, testing all sensors, and testing related to access rights to the server room as well as actions taken according to the program scenario and initial research objectives.

3.1. Sensor data transmission and display test

The first test is a test of sending and storing sensor data to a database, the final results of which will be displayed on the application dashboard page. In this test, all read sensor data has been successfully saved into the database and also displayed on the application dashboard page. The final result of this section is shown in Fig. 5.

Fig. 5. Display monitoring data on the dashboard.

In Fig. 5, it can be seen that the data on temperature, humidity, carbon monoxide, carbon dioxide, PM2.5 value, PIR1 sensor value, and PIR2 sensor value have been successfully read and displayed.

3.2. Sensor and action data test

The second test is a test related to testing sensor data related to temperature, humidity, and air quality as well as actions taken when the measured values are outside predetermined limits. The results of this test are shown in Table 3.

Table 3. Sensor and action data test results.

Test No.	SHT31 (Temp)	SHT31 (Hum)	MQ-9 (CO)	MQ-135 (CO_2)	PM2.5	Buzzer & Telegram	Status
1	22.0	46%	2.85	396	84	-	valid
2	22.2	45%	2.36	396	86	-	valid
3	22.0	45%	3.09	396	90	-	valid
4	21.4	43%	3.14	397	86	active & sent	valid
5	22.0	46%	3.10	390	86	-	valid
6	22.0	46%	3.09	397	84	-	valid
7	23.0	47%	3.04	397	88	-	valid
8	22.9	46%	3.07	396	86	-	valid
9	24.0	48%	3.04	399	84	active & sent	valid
10	24.0	48%	3.09	390	88	active & sent	valid

The test results in Table 3 regarding conditions and actions based on data from all sensors show valid results. This shows that the system can obtain data from each sensor, and then take action according to the conditions that occur. For example, in line 4 it can be seen that when one of the data obtained is outside the value range (in this case the air humidity value), the system immediately activates the buzzer and sends a Telegram message to the server manager. Conversely, when the value read from each sensor is within the specified limits, the system does not take any action. One example of an alert received by a user is shown in Fig. 6.

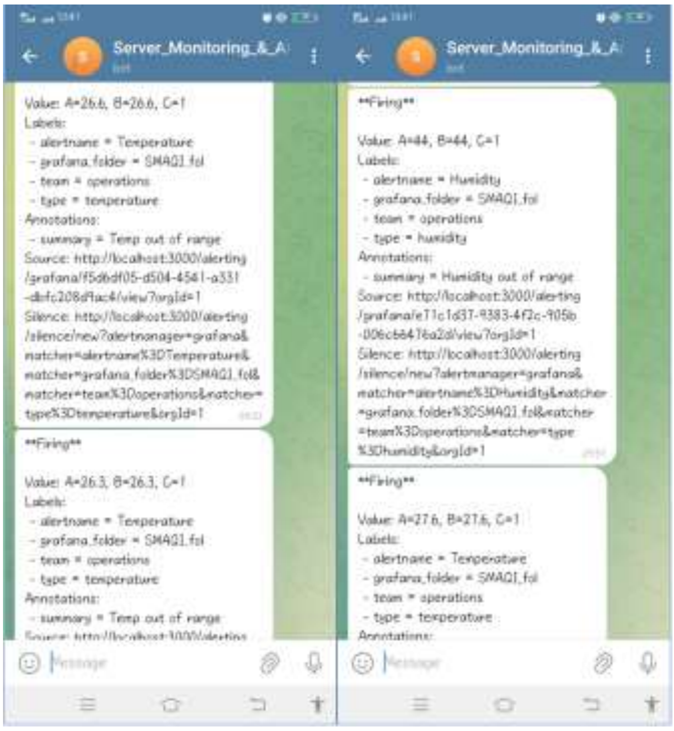

Fig. 6. An alert is received when the value of one of the sensors is outside the limits.

3.3. Testing access rights to the server room

This third test focuses on access rights to the server room and includes testing the Ultrasonic sensor, PIR1 sensor, PIR2 sensor, Solenoid door lock, and Pi Camera, as well as the actions taken in accordance with the program scenario and original research objectives. The test results from this section are shown in Table 4.

The test results in Table 4 regarding conditions and actions based on data from all sensors and actuators show valid results. At points 1, 2, 9, and 10, it can be seen that when the RFID tag is identified and the status is TRUE, the values from the ultrasonic sensor, PIR1 sensor, and PIR2 sensor will be ignored and the solenoid will be disabled. In simple terms, the person who enters the server room is the person who has access rights. However, when the status of the RFID reader is still FALSE, the values of all sensors will be compared. This can be seen in points 4, 6 and 7. In point 4, the distance measured is less than 300cm and the PIR1 sensor detects movement and is HIGH, so the Raspberry Pi 4 takes a picture using the Pi

Camera, activates the buzzer, and sends a Telegram message to the server room manager. In points 6 and 7, the condition that occurs is a violation of access rights which is detected via the PIR sensor, so the Raspberry Pi 4 immediately takes the Raspberry Pi takes a picture, activates the buzzer, and sends an alert via Telegram order. Points 3, 5, and 8 are conditions when there is no activity in the server room and all sensors and solenoids continue to work according to the program scenario. Based on the test results data in Table 3, it can be concluded that the system in the server room security section has run well following the program scenario and research objectives.

Table 4. Sensor and action data test results.

Test No.	RFID	HC-SR04	Solenoid	PIR 1	PIR 2	Pi Camera	Buzzer & Telegram
1	√	i	d	i	i	no	x
2	√	i	d	i	i	no	x
3	x	>300	a	LOW	LOW	no	x
4	x	179 cm	a	HIGH	LOW	c	a & s
5	x	>300	d	LOW	LOW	no	x
6	x	>300	a	HIGH	LOW	c	a & s
7	x	>300	a	LOW	HIGH	c	a & s
8	x	>300	a	LOW	LOW	no	x
9	√	i	d	i	i	no	x
10	√	i	d	i	i	no	x

Description:
i = ignored a = activated d = deactivated c=captured a & s = activated & sent

Based on all the test results, it can be concluded that the system created has been able to monitor the condition of the server room regarding temperature, humidity, AQI, and security of server room access by utilizing Raspberry PI 4 and IoT technology, while in other research this is still partially done, such as only monitoring the temperature and humidity alone [5-7, 9, 10, 13, 14] atau room security [8] atau AQI [11].

4. Conclusion

The server security and air quality monitoring system using Raspberry Pi 4 and Telegram has been running well, and this system has been able to run following the program scenario and initial research objectives. This can be seen from the test results, where this system has been able to measure and monitor temperature, air humidity, and AQI, limit access rights to the server room, and provide alerts to the server room manager when something is measured and detected wrong with a success percentage of 100%. For further development, it is possible to add a facial recognition feature for access rights to the server room and video capture in the server room.

References

1. Yaacoub J.P.A.; Salman O.; Noura H.N.; Kaaniche N.; Chehab A.; and Malli M.(2020). Cyber-physical systems security: Limitations, issues and future trends. *Microprocessors and Microsystems*, 77(1), 1-45.

2. Ghafir I.; Saleem J.; Hammoudeh M.; Faour H.; Prenosil V.; and Jaf S. (2018). Security threats to critical infrastructure: The human factor. *The Journal of Supercomputing*, 74(1), 4986-5002.
3. Scarpato N.; Pieroni A.; Di-Nunzio L.; and Fallucchi F.(2017). E-health-IoT universe: A review. *International Journal on Advanced Science, Engineering and Information Technology*, 7(6), 2328-2336.
4. Masoa, D. (2023). Kajian teoritis penggunaan teknologi informasi pada layanan pemasyarakatan guna meningkatkan kualitas dan kepuasan. *Innovative: Journal of Social Science Research*, 3(6), 4818-4833.
5. Jahlool J.S. (2017). Design and simulation of automatic temperature control and alert system based PIC16F887. *International Journal of Informatics and Communication Technology (IJ-ICT)*, 6(2), 95-104.
6. Dash, S.K.; Mohanty, G.; and Mohanty, A. (2012). Intelligent air conditioning system using fuzzy logic. *International Journal of Scientific & Engineering Research*, 3(12), 1-6.
7. Purwanto, F.H.; Utami. E.; and Pramono, E. (2018). Implementation and Optimization of Server Room Temperature and Humidity Control System using Fuzzy Logic Based on Microcontroller. *Journal of Physics: Conference Series*, 1140(1), 1-13.
8. Niraj, S.; Kamlesh, K.; Pankaj, W.; Danish, S.; and Arvind, G. (2018). Server room access control using smartphone. *International Journal of Engineering and Techniques*, 4(3), 56-59.
9. Sari, M.; Gunawan; and Sadikin, N.(2019). Server temperature monitoring system using web based censor and SMS gateway. *Journal of Physics: Conference Series*, 1361(1), 1-9.
10. Awaludin, M.; Rangan, A.Y.; and Yusnita, A. (2021). Internet of things (IoT) based temperature and humidity monitoring system in the chemical laboratory of the samarinda industry standardization and research center. *Tepian*, 2(3), 85-93.
11. Razzaaq, N.; Azhiim, N.; and Atmadja, M.D.(2019). Implementasi sistem monitoring Hvac ruang server menggunakan raspberry pi berbasis web (studi kasus pada Pt . Rahajasa media internet Surabaya). *Jurnal JARTEL*, 9(2), 86-92.
12. Kurniawan, D.E.; Iqbal, M.; Friadi, J.; Borman, R.I.; and Rinaldi, R.(2019). Smart monitoring temperature and humidity of the room server using raspberry pi and whatsapp notifications. *Journal of Physics: Conference Series*, 1351(1), 1-8.
13. Ab Rahman, R.; Hashim, U.R.A.; and Ahmad, S. (2020). IoT based temperature and humidity monitoring framework. *Bulletin of Electrical Engineering and Informatics*, 9(1), 229-237.
14. Wicaksono, M.F.; Rahmatya, M.D.; and Ilham (2022). IoT implementation for server room security monitoring using telegram API. *International Journal on Advanced Science, Engineering and Information Technology*, 12(5), 1931-1937.
15. Gonzalez-Huitron, V.; León-Borges, J.A.; Rodriguez-Mata, A.E.; Amabilis-Sosa, L.E.; Ramírez-Pereda, B.; and Rodriguez, H. (2021). Disease detection in tomato leaves via CNN with lightweight architectures implemented in Raspberry Pi 4. *Computers and Electronics in Agriculture*, 181(December 2020), 1-9.

16. Sprodowski, T.; Rode, J.; and Freitag, M. (2023). Evaluation of optimisation techniques on the performance of an on-line MPC controller in an occupancy grid for autonomous mobile robots. *IFAC-PapersOnLine*, 56(2), 11123-11128.
17. Juchem, J.; De-Roeck, M.; and Loccufier, M.(2023). Low-cost vision-based embedded control of a 2DOF robotic manipulator. *IFAC-PapersOnLine*, 56(2), 8833-8838.
18. Rzepka, K.; Szary, P.; Cabaj, K.; and Mazurczyk, W. (2024). Performance evaluation of Raspberry Pi 4 and STM32 Nucleo boards for security-related operations in IoT environments. *Computer Networks*, 242(1), 1-11.
19. Bujnák, M.; Pirník, R.; Nemec, D.; and Hruboš, M. (2021). Universal firefighter sensor for dangerous road tunnel environment. *Transportation Research Procedia*, 55(2019), 1019-1025.
20. Exaudi, K.; Setiawan, N.S.; Prasetyo, A.P.; Passarella, R.; Ubaya, H.; Sari, P.; and Mediswati, R.Y.T. (2023). Purwarupa sistem pemantauan polusi udara di ruang tertutup menggunakan platform thingspeak. *JTT (Jurnal Teknologi Terapan)*, 9(2), 101-110.
21. Sai, K.B.K.; Mukherjee, S.; and Sultana, H.P. (2019). Low cost IoT based air quality monitoring setup using Arduino and MQ series sensors with dataset analysis. *Procedia Computer Science*, 165(1), 322-327.
22. Karami, M.; McMorrow, G.V.; and Wang, L. (2018). Continuous monitoring of indoor environmental quality using an arduino-based data acquisition system. *Journal of Building Engineering*, 19(1), 412-419.
23. Sani, A.B.B.A.; and Beauty, P.E. (2019). Smart framework for environmental pollution monitoring and control system using IoT-based technology. *Sensors. Transducers*, 229(1), 84-93.
24. Saidi, T.; Zaim, O.; Moufid, M.; El Bari, N.; Ionescu, R.; and Bouchikhi, B. (2018). Exhaled breath analysis using electronic nose and gas chromatography-mass spectrometry for non-invasive diagnosis of chronic kidney disease, diabetes mellitus and healthy subjects. *Sensors and actuators B: Chemical*, 257(1), 178-188.
25. Küçükhüseyin Ö. (2021). CO_2 monitoring and indoor air quality. *The REHVA European HVAC Journal*, 58(1), 54-59.
26. Concas, F.; Mineraud, J.; Lagerspetz, E.; Varjonen, S.; Liu, X.; Puolamäki, K.; and Tarkoma, S. (2021). Low-cost outdoor air quality monitoring and sensor calibration: A survey and critical analysis. *ACM Transactions on Sensor Networks (TOSN)*, 17(2), 1-44.
27. Tryner J.; Mehaffy J.; Miller-Lionberg D.; and Volckens J.(2020). Effects of aerosol type and simulated aging on performance of low-cost PM sensors. *Journal of Aerosol Science*, 150 (1), 1-11.
28. Ouimette, J.; Arnott, W. P.; Laven, P.; Whitwell, R.; Radhakrishnan, N.; Dhaniyala, S.; ... and Volckens, J. (2024). Fundamentals of low-cost aerosol sensor design and operation. *Aerosol Science and Technology,* 58(1), 1-15.
29. Feng, Z.; Zheng, L.; Zhang, X.; Liu, J.; Xue, N.; AND Wang, M. (2024). Evaluation and calibration of low-cost particulate matter sensors for respirable coal mine dust monitoring. *Aerosol Science and Technology*, 58(2), 158-169.
30. Perlmutt, L.D.; and Cromar, K.R. (2019). Evaluation of the air quality index as a risk communication tool. *Journal of Environmental Health*, 81(6), 8-15.

31. Anderson, K.; and Gaston, K.J. (2013). Lightweight unmanned aerial vehicles will revolutionize spatial ecology. *Frontiers in Ecology and the Environment*, 11(3), 138-146.
32. Pereira, P.F.; and Ramos, N.M. (2021). The impact of mechanical ventilation operation strategies on indoor CO2 concentration and air exchange rates in residential buildings. *Indoor and Built Environment*, 30(9), 1516-1530.
33. Ihamji M.; Abdelmounim E.; Zbitou J.; Bennis H.; and Latrach M. (2019). A Compact Miniature Fractal Planar Antenna for RFID Readers. *Indonesian Journal of Electrical Engineering and Computer Science*, 15(1), 300-305.
34. AnnRoseela J.; and Godhavari T. (2020). Biometric and RFID based authentication system for exam paper leakages detection using IoT technology. *Indonesian Journal of Electrical Engineering and Computer Science*, 20(3), 1271-1277.
35. Al Haromainy, M.M.; Damaliana, A.T.; Najaf, A.R.E.; and Permatasari, R. (2022). Hyper smart cart as hypermart business process improvement in minimizing in-efficiency at the cashier. *RIGGS: Journal of Artificial Intelligence and Digital Business*, 1(1), 07-12.
36. El-Beqqal M.; Azizi M.; and Lanet J.L. (2020). Multimodal access control system combining RFID, fingerprint and facial recognition. *Indonesian Journal of Electrical Engineering and Computer Science*, 20(1), 405-413.
37. Vagaš M.; Galajdová A.; Šimšík D.; and Onofrejová D. (2019). Wireless data acquisition from automated workplaces based on RFID technology. *IFAC-PapersOnLine*, 52(27), 299-304.
38. Kah Wen M.K.; Ahmad N. binti; and binti Ruslan S.H. (2020). Arduino based outing and attendance system for boarding school students. *Indonesian Journal of Electrical Engineering and Computer Science*, 20(2), 1053-1061.
39. Odat, E.; Shamma, J.S.; and Claudel, C. (2017). Vehicle classification and speed estimation using combined passive infrared/ultrasonic sensors. *IEEE transactions on intelligent transportation systems*, 19(5), 1593-1606.
40. Wicaksono, M.F.; and Rahmatya, M.D. (2022). IoT for Residential Monitoring Using ESP8266 and ESP-NOW Protocol. *Jurnal Ilmiah Teknik Elektro Komputer Dan Informatika*, 8(1), 93-106.
41. Nuanmeesri, S.; and Poomhiran, L. (2021). Optimization Shortest One-Way Path for Energy Saving Auto Robot Collecting Floating Garbage using Fast Approximate Nearest Neighbor Search. *International Journal on Advanced Science, Engineering and Information Technology*, 11(2), 457-464.
42. Benny, L.; and Soori, P.K. (2017). Prototype of parking finder application for intelligent parking system. *International Journal on Advanced Science, Engineering and Information Technology*, 7(4), 1185-1190.
43. Surantha, N.; and Wicakson, W.R. (2020). Design of smart home security system using object recognition and PIR sensor. *Procedia Computer Science*, 135, 465-472.
44. Surantha, N.; and Wicaksono, W.R. (2018). Design of smart home security system using object recognition and pir sensor. *Procedia Computer Science*, 135(1), 465-472.

45. Jacob, M.; and Mani, P. (2018). A reference model for testing internet of things based applications. *Journal of Engineering Science and Technology*, 13(8), 2504-2519.
46. Alathari, B.; Kadhim, M.F.; Al-Khammasi, S.; and Ali, N.S. (2019). A framework implementation of surveillance tracking system based on pir motion sensors. *Indonesian Journal of Electrical Engineering and Computer Science*, 13(1), 235-242.
47. Murad, M.; Bayat, O.; and Marhoon, H.M. (2021). Design and implementation of a smart home system with two levels of security based on IoT technology. *Indonesian Journal of Electrical Engineering and Computer Science*, 21(1), 546-557.
48. Murad, A.; Bayat, O.; and Marhoon, H.M. (2021). Implementation of rover tank firefighting robot for closed areas based on arduino microcontroller. *Indonesian Journal of Electrical Engineering and Computer Science*, 21(1), 56-63.
49. Miikki, K.; Karakoç, A.; Rafiee, M.; Lee, D.W.; Vapaavuori, J.; Tersteegen, J.; Lemetti, L.; and Paltakari, J. (2021). An open-source camera system for experimental measurements. *SoftwareX*, 14, 100688.
50. Mouy, X.; Black, M.; Cox, K.; Qualley, J.; Mireault, C.; Dosso, S.; and Juanes, F. (2020). FishCam: A low-cost open source autonomous camera for aquatic research. *HardwareX*, 8, e00110.
51. Ganesh, K.V.S.S.; Jeyanth, S.P.S.; and Bevi, A.R. (2022). IOT based portable heart rate and SpO2 pulse oximeter. *HardwareX*, 11(1), 1-10.
52. Llata, J.R.; Sarabia, E.G., and Oria, J.P. (2002). Three-dimensional robotic vision using ultrasonic sensors. *Journal of Intelligent and Robotic Systems*, 33(1), 267-284.
53. Al-Turjman, F.; Nawaz, M.H.; and Ulusar, U.D. (2020). Intelligence in the internet of medical things era: A systematic review of current and future trends. *Computer Communications*, 150(1), 644-660.
54. Oladele, T.O.; Ogundokun, R.O.; Adegun, A.A.; Adeniyi, E.A.; and Ajanaku, A.T. (2021). Development of an Inventory Management System Using Association Rule. *Indonesian Journal of Electrical Engineering and Computer Science*, 21(3), 1868-1876.
55. Pramesti, A.A.; Arifudin, R.; and Sugiharti, E. (2016). Expert system for determination of type lenses glasses using forward chaining method. *Scientific Journal of Informatics*, 3(2), 177-188.
56. Gupta, V.; Sharma, M.; Pachauri, R.K.; and Babu, K.D. (2021). A low-cost real-time IOT enabled data acquisition system for monitoring of PV system. *Energy Sources, Part A: Recovery, Utilization, and Environmental Effects*, 43(20), 2529-2543.
57. Sharma, P.; Bourgeois, M.; Elmore, D.; Granger, D.; Lipschutz, M.E.; Ma, X.; and Vogt, S. (2000). PRIME lab AMS performance, upgrades and research applications. *Nuclear Instruments and Methods in Physics Research Section B: Beam Interactions with Materials and Atoms*, 172(1-4), 112-123.