Smart Home System Using Internet Of Things (Iot) Based On Arduino Mega And Esp32

Muttaqin Hardiwansyah, Monika Faswia Fahmi, Puteri Nurul Ma'rifah, Achmad Zain Nur, Safriudin Rifandi, Muhhammad Iyan Putra Pratama.
Department of Electrical Enginering University of Trunojoyo Madura, Bangkalan, Indonesia
muttaqin.hardiwansyah@trunojoyo.ac.id, monika.faswiaf@trunojoyo.ac.id,
puteri.marifah@trunojoyo.ac.id, azain.nur@trunojoyo.ac.id, safriudin.rifandi@trunojoyo.ac.id.

***Abstract*—Smart Home systems based on the Internet of Things (IoT) offer modern solutions to enhance efficiency, comfort, and security in managing household appliances. This research aims to develop a smart home system using Arduino Mega as the main controller and the ESP32 module as a Wi-Fi network connector. The system is designed to control devices such as lights, fans, automatic doors, and a plant watering system through an Apache-based web server connected to a MySQL database. The methods used include the design of a smart home miniature hardware prototype, the development of a web interface for manual and automatic control, as well as the integration of a voice command feature based on Google Text-to-Speech. Testing was conducted by measuring Quality of Service (QoS) parameters such as delay, throughput, packet loss, and jitter, along with functional testing of devices through manual, automatic, and voice command operations. The results show that the system can operate properly with an accuracy rate of 100%, an average delay of 4–5 seconds for device control, and an average QoS index of 1.5. Some minor delays were observed, mainly due to the use of a web server and internet connection. With affordable development costs, this system provides a practical solution for small-scale IoT-based smart home implementation.**

Keywords—Smart Home, Internet of Things, Arduino Mega, ESP32, Web Server, MySQL Database, Quality of Service.

I. INTRODUCTION

Technological advancements are rapidly transforming industry and society. In the Industry 4.0 era, many devices operate via computer systems, notably in smart homes, enhancing productivity and usability. The Internet of Things (IoT) enables remote control of lighting, temperature, and security, improving comfort, energy efficiency, and safety. IoT technology in smart homes integrates advanced technology into daily life, creating smarter, safer, and more efficient environments [1].

In big cities, empty homes attract thieves during holidays, who employ tactics like asking for directions or disguising themselves. An IoT-based security system enables real-time monitoring and door control, providing immediate alerts about break-ins. It is also more affordable than traditional security systems, making it an economical choice for home security [2].

In today's globalized world, advancements in science and technology, especially automation, significantly benefit humanity. Innovations like the Internet of Things (IoT) allow remote control of devices in Smart Homes, improving efficiency in functions such as security, lighting, and gas leak detection. Homeowners often struggle to monitor their homes remotely, risking high electricity bills if lights are left on. Many modern homes still use traditional keys that are easy to duplicate, increasing security risks [3].

Numerous studies on smart homes exist. Notably, Fawzan Masikul and Fikiana Prasetyovati from the University of Muhammadiyah Ponologo developed a Raspberry Pi-based system for remote appliance control. Marina, Ida, Edwinant, and Angi created a smart home for remote light control via the Blynk app. Fatur Zaini Rachman from Balikpapan State University designed an IoT-based smart home prototype accessible from various devices. The author aims to create another smart home prototype using the Ousea PLC as the controller. The equipment used in this prototype includes a power supply, on/off switch, photocell sensor, PIR sensor, Ouseal PLC, ESP8266 module, relay module, lights, fan, socket, and servo motor [4].

In the 2024 Digital Transformation Technology (Digitech) journal with the title "Designing an internet-based agricultural system based on the Internet of Things" by Rich Hansel Harefa and Hendra Gunawan [5], discusses the development of an internet-based agricultural system using the Internet of Things (IoT). This system utilizes various sensors, such as the YL-69 floor sensor and the DHT-11 air sensor, along with the NodeMCU ESP-32 microcontroller to monitor real-time environmental conditions in agricultural settings.

In the 2023 Journal of Information System Research, Supiyandi discusses an IoT smart home system that integrates and controls home devices for energy and water savings. Controlled by an Arduino microcontroller and various sensors, it enables remote monitoring and control via SMS and the Internet. This system enhances security, efficiency, and resident comfort. With remote monitoring and control capabilities, residents can manage household appliances more effectively and respond to real-time environmental conditions [6].

Paul K.A Windesi [7] in their research states that the presence of COVID-19 has made the role of technology increasingly important. Social distancing requires many business processes and activities to be conducted online. Home Automation can help simplify human tasks by leveraging the Internet. This journal discusses the development of a prototype for Internet of Things (IoT)-based home automation using NodeMCU ESP32.

B. Balaji [8] proposed a domestic automation system using IoT and Arduino, where Arduino Uno acts as a web server to control appliances via Wi-Fi. The system consists of hardware and software modules, where the Arduino UNO is used as a web server and interface for all hardware

settings, controlling home appliances through a mobile or web application.

Winarno [9] proposed the design and construction of a smart house prototype based on IoT using ESP8266, enabling control of household devices via Wi-Fi to improve home security and efficiency. This system consists of light control, air humidity and room temperature monitoring, detection of human movement in unoccupied homes, early detection of fire sources to prevent fires, and detection of LPG gas leaks in homes.

Automation in households, enabled by IoT-based smart home systems, reflects the growing reliance on Internet connectivity, as discussed by Kumar [10]. This device features a mobile app for equipment control, using IoT with the ESP8266 Wi-Fi module and Arduino IDE, and displays output on an LCD. It enhances interactivity and efficiency, employing the ATMega328 in the Arduino UNO alongside the ESP8622 for connections like theft alarms, leak detection, automatic water filling, and sensor-activated lights and fans. Users control functions via the app, receiving alert notifications. The system is low power, secure, cost-effective, and energy-efficient. Motion detection activates lights and fans, while alerts for theft, water levels, and gas leaks are provided through the app and buzzer notifications..

A 2021 IEEE study on smart homes using ESP8266, MQTT, and Node-RED emphasized the demand for efficient energy use and remote monitoring of home conditions [11], including device security and control. Key challenges include high costs, privacy issues, and the proprietary nature of existing solutions. The design employs NodeMCU ESP8266 with MQTT and Node-RED, utilizing a Raspberry Pi 3B+ as an MQTT broker for sensor communication. It integrates DHT 22 sensors for real-time data display on the Node-RED dashboard and allows device control via an MQTT-supporting interface. This affordable, open-source solution enhances energy efficiency and can be expanded with additional sensors and devices to cover home setups comprehensively.

Sarishma, Chamoli, and Sangwan [12] developed a low-cost smart home automation system using ESP8266 and IoT, enabling remote control and monitoring of devices via Amazon EC2. The model employs IPv6 for unique device addressing and integrates sensors to collect environmental data through MQTT, with storage on the ThingSpeak Cloud Platform. The prototype successfully demonstrates data collection and user interface control for smart homes. This interface enables remote control of home devices, addressing security, privacy, and integration challenges. The research presents a cost-effective smart home automation solution for diverse users, enhancing remote monitoring, energy efficiency, and convenience. Utilizing affordable devices like ESP8266 and ThingSpeak Cloud, it offers a low-cost alternative to pricier commercial options. Additionally, the paper emphasizes the need for thorough guidance for homeowners to understand system operation and safeguard their data.

The research contribution is miniature hardware prototype smart home systems based on the internet of things (IoT) to control devices such as lights, fans, automatic doors, and a plant watering.

II. MATERIAL AND METHOD

A. Methodology

This research employs a five-stage methodology: preparation, system planning, tool design, results analysis, and evaluation, as illustrated in Figure 1. Preparation identifies the topic and conducts a literature review. System planning outlines the smart home framework, including hardware and software requirements. Tool design develops the prototype using hardware (Arduino Mega, ESP32, sensors, actuators) and control software. Testing assesses performance across various control methods, measuring Quality of Service parameters. Evaluation concludes with an assessment of the system's effectiveness and limitations, documented in the thesis report.

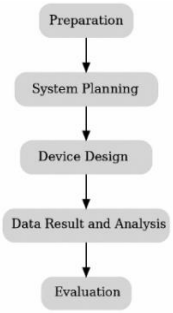


Fig. 1 Research Methodology

B. System Design

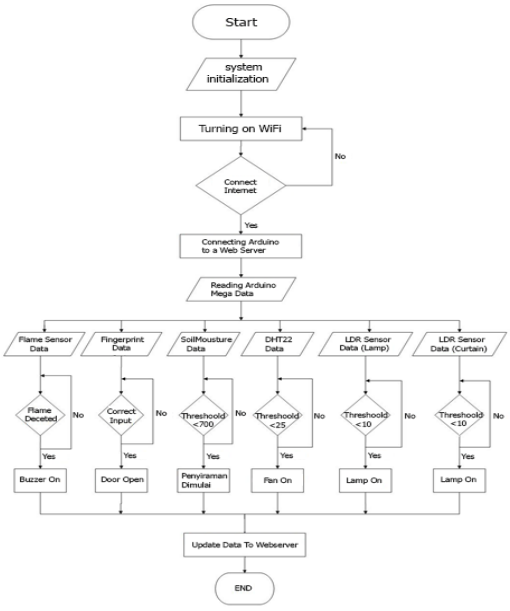


Fig. 2. Automation System Flowchart

The determination of threshold values for each sensor refers to several journal references. For the soil moisture sensor, the soil is considered dry if the moisture level is below 700 RH [13]. For the DHT22 temperature sensor, the room temperature is considered hot if it exceeds 25°C [14]. For the LDR light sensor, the condition is considered dark if the light intensity is below 10 lux [15]. These thresholds are used as a reference to automatically activate devices in a smart home system.

The Smart Home system flowchart shown above explains the sequence of steps in the system control process. The process begins with the initial condition where the system is ready for use. The user accesses the user interface (UI) on the Web Server device via Android to control the system. This interface connects the device to the WiFi network. Once connected, the user enters control commands that are forwarded to the sensor. The sensors detect environmental conditions and send data to the Arduino Mega 2560. Based on this data, the Arduino controls the lights and other devices. The system checks whether the devices are functioning according to the commands. If so, the system issues the corresponding command output, activating or deactivating the devices. The process ends when the devices are functioning according to the commands, and the system returns to the initial state, ready to receive the next command.

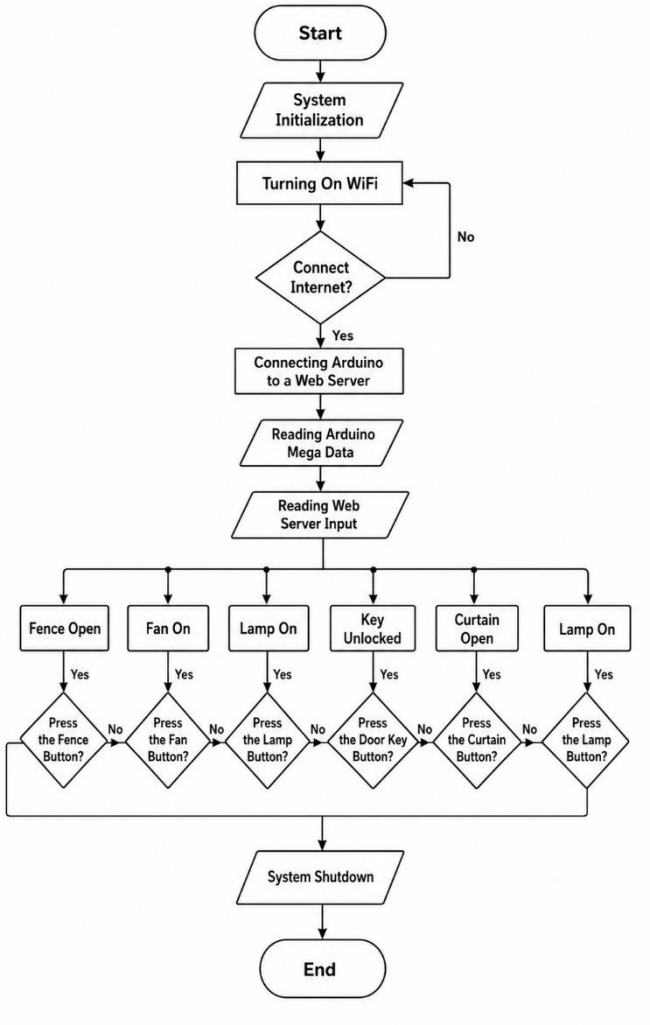


Fig. 3. Control System Flowchart

Block diagram illustrates the data and control flow in an Arduino Mega 2560-based system that supports automatic monitoring and control via a web server.

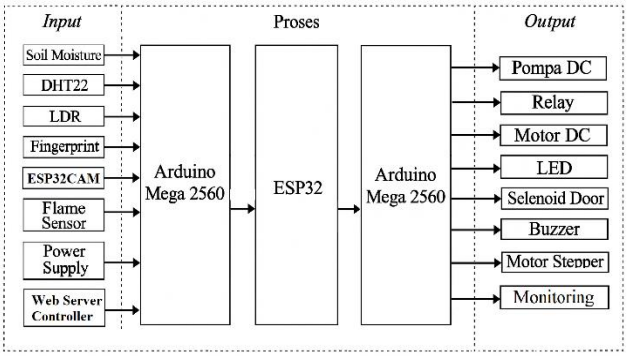


Fig. 4. Block Diagram

C. *Hardware Design*

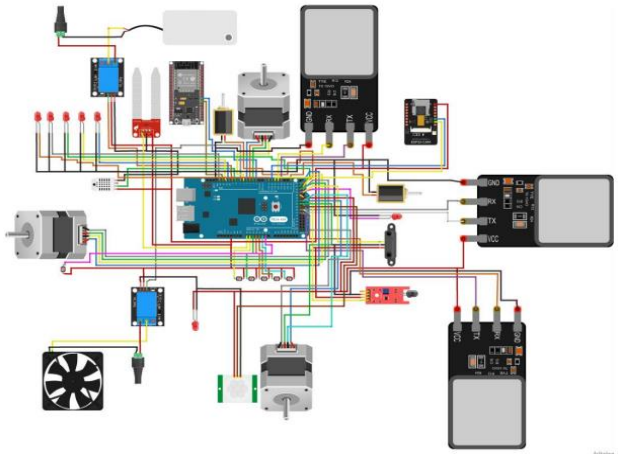


Fig. 5. Hardware Circuit Diagram

The image above shows the hardware components of a Smart Home system, with the Arduino Mega serving as the I/O and central processing unit for operating various home automation devices.

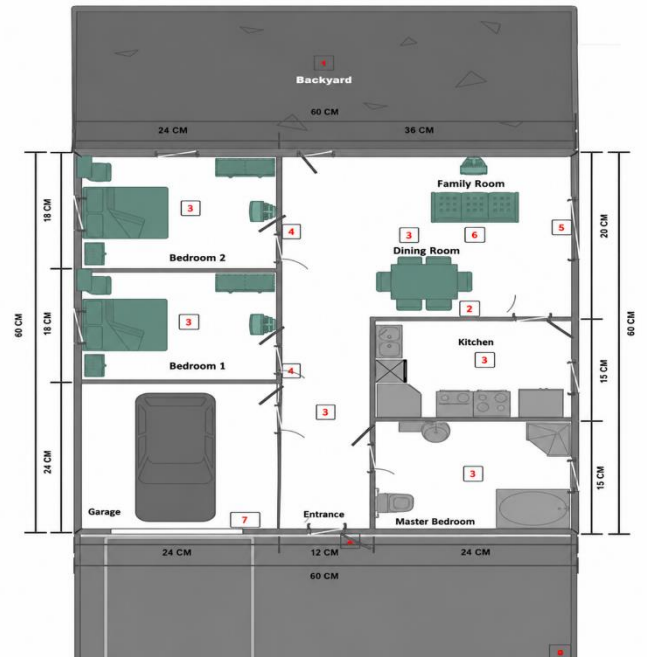


Fig. 6. Smart Home Miniature Floor Plan

On the floor plan, there are numbers representing various Smart Home features that will be installed. These numbers serve as markers for each feature. A detailed explanation of these numbers is as follows:

1. Automatic Plant Watering System
2. Automatic Fan
3. Automatic Lights
4. Fingerprint Security Door
5. Automatic Curtains
6. Automatic Fire Alarm
7. Automatic Garage
8. CCTV

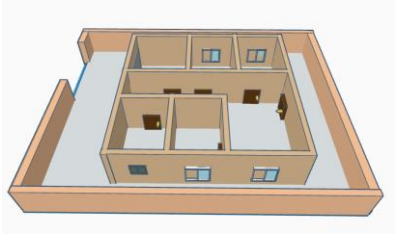


Fig. 7. Miniature Floor Plan Right View

D. *System Software Design*

E. *HTTP Method*

HTTP Method is a mechanism for client-server communication using HTTP, where the GET method retrieves device status (e.g., lamp ON/OFF) from ESP32, and the POST method sends commands to change device status (e.g., turn lamp ON/OFF), with ESP32 executing commands via a relay and responding accordingly [16]. This research aims to automate Smart Home device management through a technology-based system, utilizing sensors to detect device status and position, triggering appropriate actions to manage devices as needed.

III. Result And Discussion

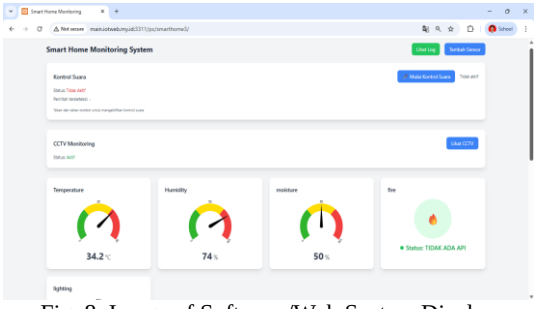


Fig. 8. Image of Software/Web System Display

Table 1. System Control Voltage Test Results

Device	Condition	Voltage (V)
Stepper Motor	Idle	12 V
	Moving	6.02 V
Solenoid	Idle	0.01 V
	Active	12 V
DC Motor	Idle	0.01 V
	Running	3.03 V
Pump	Idle	0.01 V
	Running	11.02 V
LED	OFF	0 V
	ON	3.01 V
Flame	No Fire	3.01 V
Sensor	Fire Detected	2.03 V
DHT11	Idle	3.03 V
Fingerprint	Idle	0 V
	In Use	3.03 V
LDR	Dark	0.01 V
	Bright	3.00 V
Soil Moisture	Dry	2.09 V
	Wet	1.02 V

Based on the test table, all devices in the smart home system respond well to commands from the web server. Changes in the conditions of the sensors and actuators produce voltages as expected, with clear differences between idle and active conditions, indicating that the control and output functions of the devices are working correctly.

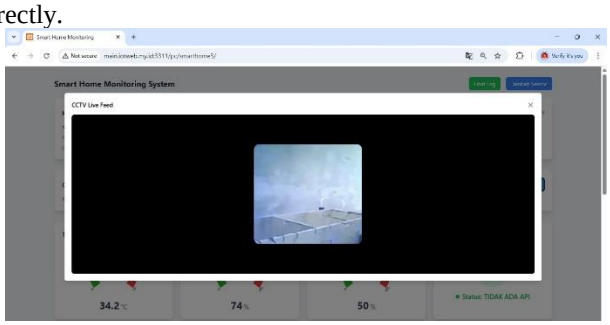


Fig. 7. CCTV Camera Results from ESP32CAM

Although the image quality of the ESP32CAM is not as high as that of a professional camera, it is sufficient for observing the surrounding conditions, detecting the presence of people, or monitoring activities inside a room. The video delay is also relatively low, enabling near-real-time monitoring.

A. *Quality Of Service (QoS)*

To determine the quality of service (QoS) of the developed smart home system, testing was conducted based on several QoS parameters, namely throughput, packet loss, delay, jitter, and accuracy. The system was tested by attempting to control the device 30 times for each feature, then recording the results.

Table 2. Web Server Statistics Using Wireshark Software

Measurement	Captured	Displayed
Packets	138	51 (37.0%)
Time span (s)	6.982	6.281
Average pps	19.08	08.01
Average packet size (B)	326	415
Bytes	44,936	21,162 (47.1%)
Average bytes/s	6,436	3,369
Average bits/s	51 k	26 k

B. *Throughput*

The higher the throughput value, the better the system performance is considered to be, as data can be transmitted quickly without significant obstacles. The formula for calculating throughput can be expressed as follows:

$$Thoughput = \frac{Total\,Data\ (bytes)}{Transmission\ Time\ (s)} \quad (1)$$

The data that will be used in the calculation is the data in the displayed column, because that column is data that has been filtered to the web server so that the throughput value calculation is:

$$Throughput = \frac{21162}{6.281} = 3369.208\ bytes/s$$

$$= 3369.208\ bytes\ x\ 8$$

$$= 26953,664\ bits/s$$

$$= 26\ Kbits/s$$

Based on these calculations, the result obtained from TP (Throughput) is 26 Kbits/s. This is in accordance with the table.

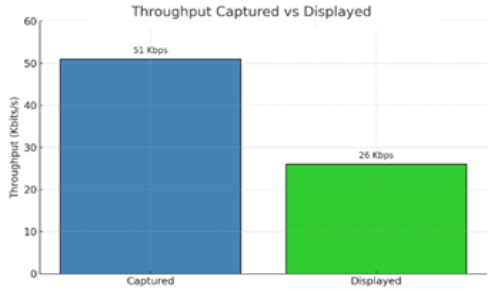


Fig. 8. Throughput Testing Graph

C. Packet Loss

Of the 51 experiments conducted, there was 1 data point that was not captured (error). Therefore, the packet loss calculation formula is as follows:

$$Packet\ Loss = \frac{Total\ Packet\ Loss}{Total\ packages\ sent}\ x\ 100 \quad (2)$$

The result of 51 trials with 1 packet loss led to the conclusion that the calculation was 1.96% packet loss.

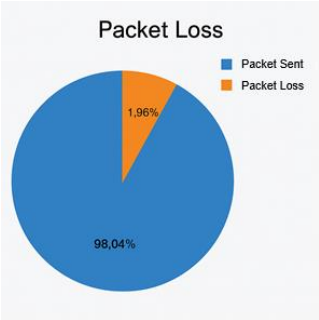


Fig. 9. Packet Loss Test Graph

D. Delay

TABLE I. TABLE 3. TABLE OF AVG. DELAY TIME TEST RESULTS IN SECONDS

No	Garage (s)	Door (s)	Curtain (s)	Lamp (s)	Fan (s)	Pump (s)
1	4.32	4.11	4.92	3.75	5.98	6.55
2	6.27	2.56	7.05	4.29	6.96	6.09
3	3.44	4.37	5.06	6.06	6.47	3.82
4	6.69	3.35	4.36	2.99	5.17	5,65
5	3.71	5.22	4.25	3.85	2.74	5.92
6	6.62	3.25	3.83	2.91	3.66	4.7
7	5.74	5.86	3.54	5.92	5.59	4.09
8	3.41	2.55	4.72	4.21	5.82	6.13
9	3.64	3.61	4.68	4.07	3.98	4.72
10	3.52	2.08	2.98	3.72	2.35	5.47
11	3.48	2.26	2.67	3.55	2.29	14.9
12	3.12	3.42	2.55	2.62	5.27	2.91
13	3.71	4.15	5.92	4.18	2.33	5.07
14	2.82	6.04	3.72	4.14	2.42	2.99
15	3.11	2.62	4.93	5.23	6.8	4.06
16	3.63	3.31	4.22	3.67	3.82	2.62
17	4.57	4.02	3.76	5.09	4.57	3.92
18	5.13	3.74	3.25	3.61	2.6	2.24
19	3.03	4.35	4.06	4.18	5.06	11.1
20	4.89	2.77	3.53	4.96	5.13	4.06
21	3.9	3.81	5.14	4.19	5.01	5.35
22	4.06	3.58	4.6	5.24	3.48	5.57
23	5.81	3.26	3.42	4.38	5.49	6.21
24	4.06	2.99	4.79	2.04	3.13	4.33
25	3.11	3.61	2.87	4.23	5.35	2.3
26	3.82	3.08	4.4	4.3	4.6	4.71
27	4.45	4.11	4.05	3.75	4.21	3.88
28	4.75	2.75	3.77	4.05	3.25	5.22
29	3.55	4.04	3.8	3.92	5.18	4.11
30	4.1	3.16	4.1	3.56	4.06	3.95

Based on the delay test results from 30 trials on each device in the smart home system, the average delay was calculated using the following formula:

$$Delay = Time2 - Time1$$

$$Total\ Delay = Delay\ 1 + Delay\ 2 + \cdots + Delay\ 30$$

$$(3)\ Delay\ Rata - rata = \frac{\sum_{i=1}^{n} Delay_i}{n}$$

Through this process, the average delay value for each device and the overall level were obtained. The results are shown in the following graph.

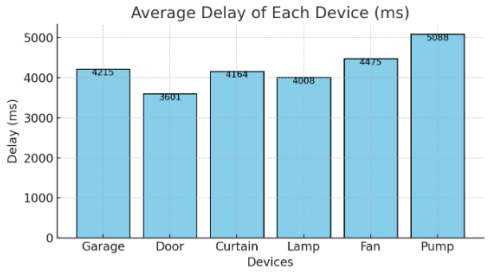


Fig. 10. Delay Testing Graph

E. Jitter

Jitter refers to the inconsistency in delay (latency) between data packets. The following is the formula for calculating jitter:

$$Delay_n = Delay\ 2$$
$$Delay_{n-1} = Delay\ 1$$

$$(4)\ Jitter = Delay_n - Delay_{n-1}$$

Through this process, the average jitter value for each device and the overall level were obtained. The results are shown in the following Graph:

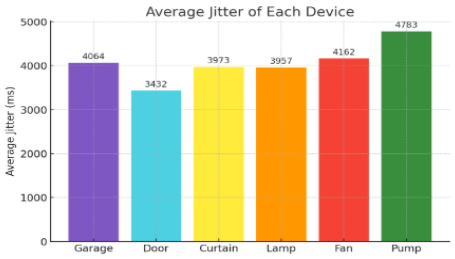


Fig. 11. Jitter Test Graph

DISCUSSION

Based on the results of testing the Quality of Service (QoS) parameters, which include Throughput, Delay, Jitter, and Packet Loss. The test results will use standard parameters in accordance with the TIPHON standard from ETSI [17]. The following is a table classifying QoS parameters based on the TIPHON standard:

TABLE 4. QOS PARAMETER STANDARDS BASED ON TIPHON STANDARDS

Parameter	**Excellent (4)**	**Good (3)**	**Fair (2)**	**Poor (1)**
Throughput	> 1 Mbps	512 Kbps – 1 Mbps	256 Kbps – 512 Kbps	< 256 Kbps
Delay	< 150 ms	150 – 300 ms	300 – 450 ms	> 450 ms
Jitter	< 20 ms	20 – 50 ms	50 – 75 ms	> 75 ms
Packet Loss	< 1%	1% – 2.5%	2.5% – 5%	> 5%

By referring to the standard table above, we can assess the quality level of the test results that have been carried out, both in terms of efficiency and reliability. The results of the Quality of Service (QoS) parameter comparison and quality level categorization are presented in the following table:

TABLE 5. AVERAGE QUALITY OF SERVICE (QOS) INDEX

QoS Parameter	Result	Index
Throughput	26 Kbit/s	1
Packet Loss	1.96%	3
Delay	4263 ms	1
Jitter	4060 ms	1
Average		1.5

IV. CONCLUSION

This smart home system can be considered "Poor" to "Fair" category based on its QoS results: Throughput 26 Kbit/s, Packet Loss 1.96%, Delay 4263 ms, Jitter 4060 ms. The average score is 1.5. This is due to very high delay and jitter values and low throughput, which significantly reduce the quality of data communication in the system. The only parameter showing relatively good performance is packet loss, but it is insufficient to improve the overall score.

REFERENCES

[1] A. Mulyana, S. Rahmawati, R. Rahman, and Z. Nur Permatasari, "ALAT PENGONTROL PERANGKAT ELEKTRONIK BERBASIS IOT MENGGUNAKAN BLYNK DAN GOOGLE ASSISTANT," *JCS-TECH*, vol. 2, no. 1, pp. 30–35, May 2022, doi: 10.54840/jcstech.v2i1.24.

[2] W. Istiana and R. P. Cahyono, "SISTEM KEAMANAN PINTU RUMAH BERBASIS Internet Of Things (IoT) Dengan ESP8266," vol. 2, 2022.

[3] K. Ikhsan and B. Setyawan, "Rancang Bangun Smart Home Berbasis Internet Of Things (IOT) Menggunakan Arduino Mega 2560," vol. 12, 2024.

[4] A. M. Nando and E. Fitriani, "Smart Home Menggunakan PLC Outseal Berbasis Internet Of Things," *AMPERE*, vol. 8, no. 1, pp. 22–31, Jun. 2023, doi: 10.31851/ampere.v8i1.9314.

[5] R. H. Harefa and H. Gunawan, "Perancangan Smart Agriculture System Berbasis Internet of Things," vol. 4, no. 1, 2024.

[6] S. Supiyandi, C. Rizal, M. Iqbal, M. N. H. Siregar, and M. Eka, "Smart Home Berbasis Internet of Things (IoT) Dalam Mengendalikan dan Monitoring Keamanan Rumah," *josh*, vol. 4, no. 4, pp. 1302–1307, Jul. 2023, doi: 10.47065/josh.v4i4.3822.

[7] P. K. A. Windesi, M. R. Sampebua, and R. M. Kmurawak, "IOT-BASED HOME AUTOMATION USING NODEMCU ESP8266," *J. Ris. Inform.*, vol. 4, no. 4, pp. 391–396, Sep. 2022, doi: 10.34288/jri.v4i4.431.

[8] B. Balaji, R. Priya, and R. Revathy, "Domestic Automation System Using Internet of Things and Arduino," in *2020 International Conference on System, Computation, Automation and Networking (ICSCAN)*, Pondicherry, India: IEEE, Jul. 2020, pp. 1–4. doi: 10.1109/ICSCAN49426.2020.9262421.

[9] Adi Winarno and M. Affandi, "Design and Construction of Smart House Prototype Based Internet of Things (Iot) Using Esp8266," *Buana Electrical, Science & Technology*, vol. 4, no. 1, pp. 11–14, Mar. 2022, doi: 10.36456/best.vol4.no1.5447.

[10] Saravana Kumar. K, Mohamed Faizal, Dheena. A, and Dr. N. Muruganantham, "SMART HOME USING IOT," *International Journal of Scientific Development and Research (IJSDR)*, vol. 6, no. 4, p. 1, Apr. 2021.

[11] P. Macheso, T. D. Manda, S. Chisale, N. Dzupire, J. Mlatho, and D. Mukanyiligira, "Design of ESP8266 Smart Home Using MQTT and Node-RED," in *2021 International Conference on Artificial Intelligence and Smart Systems (ICAIS)*, Coimbatore, India: IEEE, Mar. 2021, pp. 502–505. doi: 10.1109/ICAIS50930.2021.9396027.

[12] S. Sarishma, S. Chamoli, S. Sangwan, and V. Fore, "Smart Home Automation using ESP8266 and Internet of Things," *SSRN Journal*, 2020, doi: 10.2139/ssrn.3701427.

[13] I. Setyowati, D. Novianto, and E. Purnomo, "Preliminary design and soil moisture sensor yl-69 calibration for implementation of smart irrigation," *J. Phys.: Conf. Ser.*, vol. 1517, no. 1, p. 012078, Apr. 2020, doi: 10.1088/1742-6596/1517/1/012078.

[14] W. Adhiwibowo, A. F. Daru, and A. M. Hirzan, "Temperature and Humidity Monitoring Using DHT22 Sensor and Cayenne API," *JT*, vol. 17, no. 2, pp. 209–214, Jan. 2020, doi: 10.26623/transformatika.v17i2.1820.

[15] P. Madhavi, S. V. Satyanarayana, N. Shirisha, N. Chandana, M. Amulya, and M. Ankitha, "Automatic Night Lamp Using LDR," *IJERA*, vol. 15, no. 1, pp. 97–100, Jan. 2025, doi: 10.9790/9622-150197100.

[16] I. R. Nugraha, W. H. N. Putra, and E. Setiawan, "A Comparative Study of HTTP and MQTT for IoT Applications in Hydroponics," *J. RESTI (Rekayasa Sist. Teknol. Inf.)*, vol. 8, no. 1, pp. 119–126, Feb. 2024, doi: 10.29207/resti.v8i1.5561.

[17] ETSI, "Telecommunications and Internet Protocol Harmonization Over Networks (TIPHON); End to End Quality of Service in TIPHON Systems; Part 1: General Aspects of Quality of Service (QoS)," vol. 2.