1st BICONE (Bhinneka Conference)
Empowering Society 5.0: Education, Technology, and Social Transformation
Vol 1 No 1 , 2025

Development of a Remote Light Control System Using ESP32 Based on the Internet of Things (IoT)

Adytia Aries Nugroho[1], Abdul Haris Indrakusuma[2]
1. Information Technology Education, Science and Technology, Universitas Bhinneka PGRI, Indonesia
Email address : adytiaariesnugroho@gmail.com
2. Information Technology Education, Science and Technology, Universitas Bhinneka PGRI, Indonesia
Email address : abdulharisindrakusuma@gmail.com

***Abstract*— In the modern era, Internet of Things (IoT) technology has brought significant changes to various aspects of life, particularly in automation and remote control. One of the most common applications of IoT is the development of remote light control systems, which allow users to turn lights on and off from a distance via the internet. This study utilizes the ESP32 module, which is equipped with Wi-Fi and Bluetooth connectivity. The objective of this research is to develop a remote light control system using ESP32 based on IoT technology. The research stages included literature review to understand relevant theories and references, device and software design, system testing, and result analysis. The findings of this study indicate that the designed system successfully achieved the primary goal of real-time light control over the internet, with an average response time of 1.1 seconds.**

***Keywords*— Blynk, ESP32, Internet of Things, Light Control**

I. INTRODUCTION

In this modern era, Internet of Things (IoT) technology has brought significant changes to various aspects of life, especially in terms of automation and remote control. One of the popular applications of IoT technology is the development of a remote light control system, which allows users to remotely turn lights on and off using the internet network. By utilizing the ESP32 module, which is a microcontroller with Wi-Fi and Bluetooth connectivity capabilities, the system can be developed in a more user-friendly and cost-effective manner. According to research by [1], the use of ESP32 in IoT systems has been proven to control devices in real-time with low network levels.

A study by [2] developed an IoT-based room lighting control system application at Nurtanio University using an Android smartphone. This system allows users to turn lights on and off remotely through an Android app by utilizing an internet connection. However, the system has a limitation in the form of a fairly long delay time, which is about 60 seconds, for each command sent through Thingspeak's cloud servers using ESP8266. This shows the need to develop a more responsive system.

Based on the above background, this study aims to develop a remote light control system using an easy-to-use IoT-based ESP32. By utilizing the capabilities of the ESP32, this system is expected to be a solution to the needs of remote lighting control in the modern era.

II. LITERATURE REVIEW

A. Internet of Things

The Internet of Things (IoT) is a concept that connects various physical devices to the internet, making it possible to interact and share data with each other. This technology includes a wide range of devices such as sensors, cameras, smart home devices, vehicles, and medical devices equipped with internet connectivity [3]. The concept of IoT aims to create a network where devices communicate with each other and work automatically, collecting data from their environment to analyze and take specific actions.

B. Microcontrollers

A microcontroller is a computer system integrated into a single chip consisting of a processor, memory, and an input/output (I/O) unit. Microcontrollers are designed to control electronic devices autonomously by running programs that have been embedded in them. The ESP32 is one of the microcontrollers developed by Espressif Systems and is well-known for its capabilities in wireless connectivity, especially Wi-Fi and Bluetooth. The ESP32 has more advanced features, such as a dual-core processor, a greater number of GPIOs (General Purpose Input/Output), and better power management capabilities. These features make the ESP32 ideal for Internet of Things (IoT) applications, where connectivity and energy efficiency are key factors[4].

Figure 2. 1 ESP32 Pin (Source: upesy.com, 2023)

C. Relay

A relay is an electromechanical component that functions as an electronic switch, operated by an electrical signal. Relays allow control of high-voltage devices using low signals from microcontrollers or low-power control circuits. When a small current flows into the relay coil, the resulting magnetic field pulls the lever (armature), so that the switch contacts shift to connect or disconnect the main circuit. When the current is stopped, the magnetic field disappears, and the coil returns the lever to its original position, re-cutting the main circuit [5].

The ports on the relay have specific functions that support their operation. The NO (Normally Open) port connects to COM (Common) when the relay is active, while the NC (Normally Closed) port connects to COM when the relay is inactive. The COM port serves as the main terminal connected to NO or NC depending on the state of the relay. The IN (Input) port receives the control signal from the microcontroller to turn the relay on or off, while the VCC (Voltage at the Common Collector) port receives the relay operating voltage, usually 5V. The GND (Ground) port serves as ground or zero voltage [6].

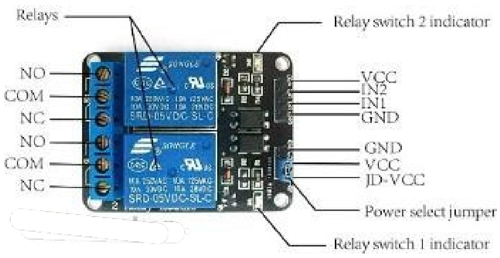

Figure 2. 2 Relay *Module* (Source: indomaker.com, 2022)

D. Platform Control IoT

An IoT control platform/application is software that allows users to control, monitor, and manage IoT devices through an easy-to-use interface, such as a mobile or web application. The platform serves as a link between IoT devices (such as microcontrollers) and users, enabling real-time two-way communication [6], blynk was chosen as the control platform in this study due to some of its advantages.

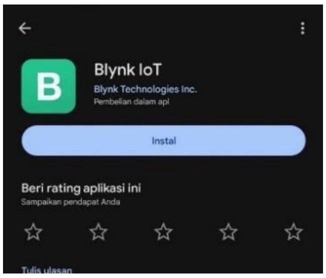

Figure 2. 3 Blynk App

E. Compiler

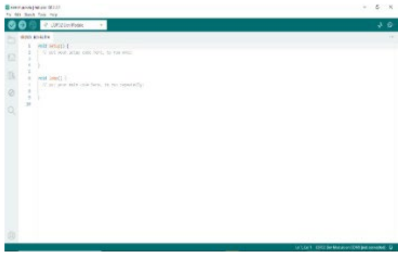

Figure 2. 4 Arduino IDE

F. Lamp

A lamp is a device that functions to produce light by converting electrical energy or other energy sources into light energy. Today, LED (Light Emitting Diode) lights are the main choice due to their advantages in terms of energy efficiency, long lifespan, and environmental friendliness. According to research by [7], LED lights consume less electrical power than incandescent or fluorescent lamps, making them more energy-efficient.

Figure 2. 5 Lights (Source: wilsoncables.com)

III. RESEARCH METHODOLOGY

This research is located in Tulungagung Regency, Bandung District and is further divided into four villages, namely Bulus Village, Ngepeh Village, Gandong Village and Suruhan Lor Village.

In this study, there are several stages carried out to achieve the desired final result, namely:

A compiler is a program used to translate source code written in a high-level programming language, such as C/C++, into machine code that can be executed by a microcontroller. The compilation process involves several stages, including parsing, optimization, and code generation, which ensures the program's code can run efficiently on the target device. In the context of ESP32-based IoT device development, compilers play a critical role in transforming program code into instructions that can be understood by microcontrollers.

In this study, the compiler used is Arduino IDE, Arduino IDE (Integrated Development Environment) is the official software used to write, compile, and upload code to Arduino microcontrollers. The Arduino IDE is designed with a simple and easy-to-use interface, making it suitable for both beginners and experienced developers [8]. The main reasons for using the Arduino IDE on the ESP32 in this study are its ease of use, extensive support for the ESP32, and the availability of abundant libraries.

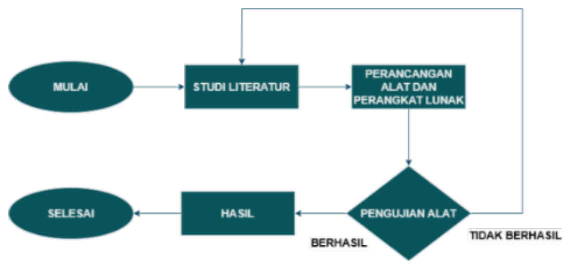

Figure 3. 1 Research Workflow Flowchart

The flowchart above explains the research flow from literature study, tool and software design, tool testing and to the results of the research,

A. *Mechanical design*

Mechanical design is the arrangement of mechanical components used in the manufacture of props (protoype). In this study, the researcher used wood board material as the basis for shaping the design of the tool. As shown in the image below, the system design consists of four light points, namely Lamp 1, Lamp 2, Lamp 3, and Lamp 4, each of which represents the position of the first to fourth lamp installation. In addition, the design also displays a microcontroller unit that functions as a system control center. This unit is an integration place between the power supply, relay module, and ESP32 microcontroller which is used to remotely control the light on and off via an internet connection.

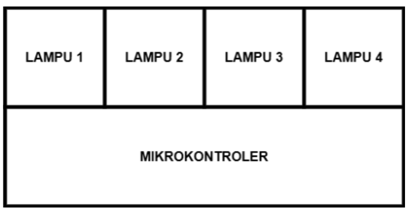

Figure 3. 2 Protoype *Design*

B. *Electrical design*

Electrical design includes several circuits that have specific functions and are integrated into a complete system. The tools in this study were controlled using ESP32 and Relay and lamps. The image below shows the pin configuration used on the ESP32 microcontroller. The VCC pin is connected to the positive voltage of the power supply, while the GND pin is connected to the negative voltage (ground). Several output pins are used to control the lights through a relay module that functions as an electronic switch. The pins used to control the relay are D12, D13, D14, and D27, which are connected to the input terminal (IN) on the relay module, respectively.

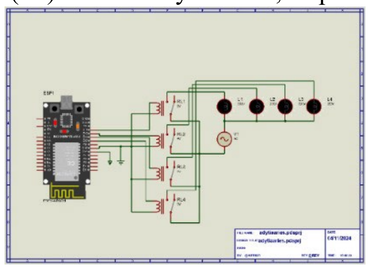

Figure 3. 3 Protoype Design

IV. RESEARCH RESULTS

A. *Mechanical design*

The image below shows the state of the prototype IoT- based light control system using ESP32 when all lights are off. This circumstance illustrates that the system has not received instructions or commands from the Blynk application to turn on the lights. At this stage, the ESP32 has been connected to the Wi-Fi network but has not received input from the user through the application, so all relays are in a state of inactivity and the electric current has not been delivered to the lights.

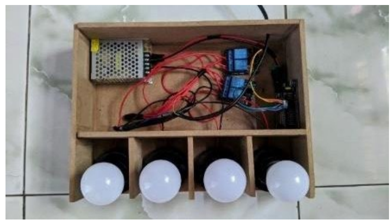

Figure 4. 1 Protoype design in the off state

Second, the image below shows the prototype system when it is on, i.e. after the ESP32 receives instructions from the Blynk application to activate one or more lights. In this condition, the digital signal from the ESP32 activates the relay module which then conducts an electric current to the lamp, so that the lamp lights on.

Figure 4. 1 *Prototype* control lights when the state is *on*

B. *Blynk App Configuration*

The initial configuration of the Blynk application is done by downloading the Blynk IoT application through official distribution platforms such as the Google Play Store for Android users, or the App Store for iOS users.

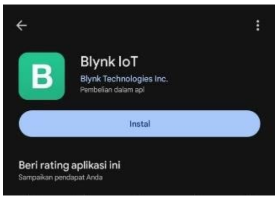

Figure 4. 2 Blynk App

Once the account is verified, the next stage is the creation of the project template. With the ESP32 hardware configuration and select WiFi in the connection section.

Figure 4. 3 *Device* Blynk

The next step is the addition of a datastream as a communication line between the ESP32 device and the Blynk application.

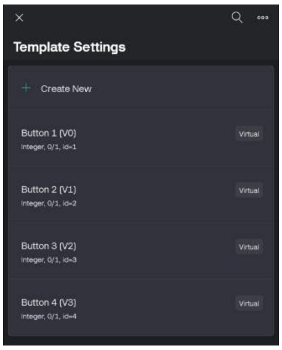

Figure 4. 4 *Blynk* Datastream

Once the datastream is completely configured, the user proceeds to design the controller interface through the Developer Zone menu

Figure 4. 5 *Button* pseudo-lamp

C. *Arduino IDE Configuration*

The first step is to add the ESP32 board to the Board Manager on the Arduino IDE. To do this, the user must go to the Tools menu > Board > Boards Manager, the user can search for the ESP32 by Espressif System board and install it with version 2.0.16.

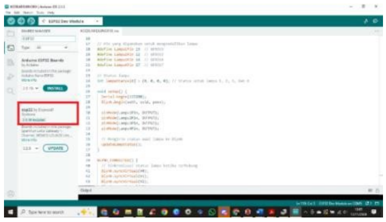

Figure 4. 6 *Board Manager* ESP32 by Espressif System

The second step is to install the Blynk library needed as a link between the ESP32 microcontroller and the Blynk app on a mobile device. Users can open the Tools menu > Manage Libraries and type the keyword "Blynk" in the search field. Next, select and install the Blynk by Volodymyr Shymanskyy library version 1.3.2.

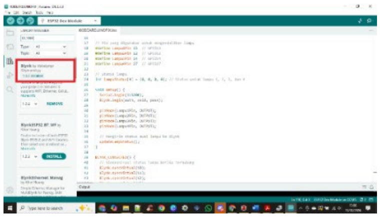

Figure 4. 7*Liibrary* Blynk by Volodymyr Shymanskyy

The third step is to write a program that has been prepared to control the lights through the WiFi network by utilizing the ESP32 Board. And the last step involves uploading the program code to ESP32.

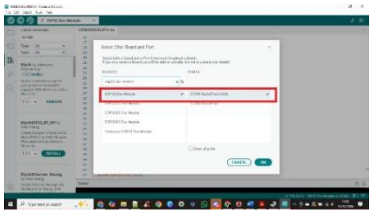

Figure 4. 8 ESP32 Module and Port

D. *Test System*

In the test stage of the light control system, make sure the prototype and device are connected to the internet, the test is carried out to measure the response time of the change in the status of the lamp from the on (ON) to off (OFF) condition and vice versa. This test aims to determine the time the system responds to commands given by the user through the Blynk application. Measurements are made using a timer to record the time difference from the command to the actual response on the light.

The results of this test are compiled based on the response time of each light to the ON and OFF commands. Each test is repeated to obtain accurate data. Below shows the results of the tests carried out in each village:

Based on the data displayed in Table 4.1, the system test was carried out in the Bulus Village area with an internet network speed of 11 Mbps. From the test results, the average system response time was 1.1 seconds.

Table 4. 1 Results of Protoype Lamp ON Testing in Bulus Village

Test to	Lamp 1	Lamp 2	Lamp 3	Lamp 4	Information

1	1 second	1.2 seconds	1.3 seconds	1.4 seconds	Succeed
2	1.2 seconds	1.1 seconds	1.2 seconds	1.1 seconds	Succeed
3	1 second	1.2 seconds	1 second	1 second	Succeed
4	1.1 seconds	1 second	1 second	1 second	Succeed
5	1 second	1.2 seconds	1.1 seconds	1.2 seconds	Succeed
6	1.3 seconds	1.2 seconds	1 second	1 second	Succeed
7	1.2 seconds	1 second	1 second	1 second	Succeed
8	1.1 seconds	1.1 seconds	1.1 seconds	1.1 seconds	Succeed
9	1.3 seconds	1.2 seconds	1.3 seconds	1.2 seconds	Succeed
10	1.3 seconds	1.1 seconds	1.2 seconds	1.1 seconds	Succeed

Based on the data displayed in Table 4.2, the system test was carried out in the Bulus Village area with an internet network speed of 11 Mbps. From the test results, an average system response time of 1.1 seconds was obtained.

Table 4. 2 Test Results of Protoype OFF Lights in Bulus Village

Test to	Lamp 1	Lamp 2	Lamp 3	Lamp 4	Information
1	0.9 seconds	1 second	1 second	1 second	Succeed
2	1 second	1.1 seconds	1.1 seconds	1.2 seconds	Succeed
3	1.2 seconds	1.3 seconds	1.3 seconds	1.4 seconds	Succeed
4	1.1 seconds	1.3 seconds	1.3 seconds	1.3 seconds	Succeed
5	1.3 seconds	1.1 seconds	1.1 seconds	1.2 seconds	Succeed
6	1.3 seconds	1.2 seconds	1.2 seconds	1.2 seconds	Succeed
7	1.4 seconds	1.2 seconds	1.2 seconds	1.3 seconds	Succeed
8	1.2 seconds	1.2 seconds	1.4 seconds	1.5 seconds	Succeed
9	1.3 seconds	1.1 seconds	1.3 seconds	1.4 seconds	Succeed
10	1 second	1.1 seconds	1.1 seconds	1 second	Succeed

Based on the data shown in Table 4.3, the system test was carried out in the Ngepeh Village area with an internet network condition that has a speed of 11 Mbps. From the test results, the average system response time was 1 second.

Table 4. 3 Results of Protoype Testing of ON Lamps in Ngepeh Village

Test to	Lamp 1	Lamp 2	Lamp 3	Lamp 4	Information
1	1.1 seconds	1 second	1.3 seconds	1.2 seconds	Succeed
2	1 second	1 second	1 second	1 second	Succeed
3	1 second	1 second	1 second	1.1 seconds	Succeed
4	1.2 seconds	1 second	1 second	1.1 seconds	Succeed
5	1.2 seconds	1.2 seconds	1.3 seconds	1.2 seconds	Succeed
6	1 second	1 second	1.1 seconds	1.1 seconds	Succeed
7	1.2 seconds	1.1 seconds	1 second	1.1 seconds	Succeed
8	1.1 seconds	1 second	1 second	1 second	Succeed
9	1 second	1 second	1 second	1.1 seconds	Succeed
10	1.1 seconds	1 second	1 second	1 second	Succeed

Based on the data shown in Table 4.4, the system test was carried out in the Ngepeh Village area with an internet network speed of 11 Mbps. From the test results, the average system response time was 1 second.

Table 4. 4 Results of Protoype Lamp OFF Testing in Ngepeh Village

Test to	Lamp 1	Lamp 2	Lamp 3	Lamp 4	Information
1	1.1 seconds	1 second	1 second	1 second	Succeed
2	1.1 seconds	1 second	1 second	1 second	Succeed
3	1.2 seconds	1.2 seconds	1.2 seconds	1 second	Succeed
4	1 second	1 second	1 second	1 second	Succeed
5	1 second	1.1 seconds	1 second	1.3 seconds	Succeed
6	1 second	1 second	1 second	1 second	Succeed
7	1.2 seconds	1.1 seconds	0.9 seconds	1.2 seconds	Succeed
8	1.1 seconds	1 second	1 second	1 second	Succeed
9	1 second	1 second	1.2 seconds	1.2 seconds	Succeed
10	1 second	1 second	1 second	1 second	Succeed

Based on the data shown in Table 4.5, the system test was carried out in the Gandong Village area with an internet network speed of 10 Mbps. From the test results, the average system response time was 1 second.

Table 4. 5 Results of Protoype Lamp ON Testing in Gandong Village

Test to	Lamp 1	Lamp 2	Lamp 3	Lamp 4	Information
1	1.2 seconds	1.2 seconds	1 second	1 second	Succeed
2	1.1 seconds	1 second	1 second	1 second	Succeed
3	1.2 seconds	1.2 seconds	1 second	1 second	Succeed
4	1 second	1 second	1 second	1.2 seconds	Succeed
5	1.1 seconds	1.1 seconds	1.1 seconds	1.3 seconds	Succeed
6	1 second	1 second	1 second	1 second	Succeed
7	1 second	1.2 seconds	1 second	1.2 seconds	Succeed
8	1.1 seconds	1 second	1.3 seconds	1.3 seconds	Succeed
9	1 second	1 second	1 second	1.3 seconds	Succeed
10	1 second	1 second	1.2 seconds	1 second	Succeed

Based on the data shown in Table 4.6, the system test was carried out in the Gandong Village area with an internet network speed of 10 Mbps. From the test results, the average system response time was 1 second.

Table 4. 6 Results of Protoype Lamp OFF Testing in Gandong Village

Test to	Lamp 1	Lamp 2	Lamp 3	Lamp 4	Information
1	1 second	1.2 seconds	1.1 seconds	1.2 seconds	Succeed
2	1.1 seconds	1 second	1 second	1.2 seconds	Succeed
3	1.1 seconds	1 second	1 second	1 second	Succeed
4	1.2 seconds	1 second	1 second	1 second	Succeed
5	1.2 seconds	1.2 seconds	1.3 seconds	1.2 seconds	Succeed
6	1 second	1 second	1 second	1 second	Succeed
7	1 second	1 second	1 second	1 second	Succeed
8	1.2 seconds	1.1 seconds	1.2 seconds	1.3 seconds	Succeed
9	1 second	1 second	1.2 seconds	1.2 seconds	Succeed
10	1.2 seconds	1.1 seconds	1.2 seconds	1.2 seconds	Succeed

Based on the data shown in Table 4.7, the system test was carried out in the Suruhan Lor Village area with internet network conditions that have a speed of 10 Mbps. From the test results, the average system response time was 1.1 second.

Table 4. 7 Results of Protoype Lamp ON Testing in Suruhan Lor Village

Test to	Lamp 1	Lamp 2	Lamp 3	Lamp 4	Information
1	1.1 seconds	1.2 seconds	1 second	1 second	Succeed
2	1.2 seconds	1.2 seconds	1.3 seconds	1 second	Succeed
3	1 second	1 second	1 second	1 second	Succeed
4	1.3 seconds	1.2 seconds	1 second	1.2 seconds	Succeed
5	1 second	1.2 seconds	1 second	1.3 seconds	Succeed
6	1 second	1 second	1 second	1 second	Succeed
7	1.2 seconds	1.2 seconds	1.1 seconds	1.2 seconds	Succeed
8	1.2 seconds	1.1 seconds	1.1 seconds	1.1 seconds	Succeed
9	1.1 seconds	1.3 seconds	1 second	1 second	Succeed
10	1.1 seconds	1 second	1 second	1.2 seconds	Succeed

Based on the data shown in Table 4.8, the system test was carried out in the Suruhan Lor Village area with an internet network speed of 10 Mbps. From the test results, the average system response time was 1.1 second.

Table 4. 8 Results of Protoype Lamp OFF Test in Suruhan Lor Village

Test to	Lamp 1	Lamp 2	Lamp 3	Lamp 4	Information
1	1.2 seconds	1.2 seconds	1.2 seconds	1.2 seconds	Succeed
2	1.1 seconds	1.1 seconds	1.1 seconds	1 second	Succeed
3	1 second	1 second	1 second	1 second	Succeed
4	1.2 seconds	1.2 seconds	1.2 seconds	1 second	Succeed
5	1 second	1 second	1 second	1 second	Succeed
6	1.1 seconds	1 second	1.1 seconds	1.1 seconds	Succeed
7	1 second	1.1 seconds	1.1 seconds	1 second	Succeed
8	1.1 seconds	1 second	1 second	1.3 seconds	Succeed
9	1.3 seconds	1 second	1 second	1 second	Succeed
10	1.3 seconds	1.3 seconds	1.3 seconds	1.2 seconds	Succeed

From the results of the tests carried out in each village, the overall remote light control has an average response time:

$$\text{Average} = \frac{\sum \textit{Response time in each village}}{\textit{Testing time}}$$

$$= \frac{\sum 1.1+1.1+1+1+1+1+1.1+1.1}{8}$$

$$= 1.1 \text{ second}$$

E. Analysis of test results

Based on the test results, the average response time of the light control system is 1.1 seconds. This shows that the remote light control using the ESP32 is able to provide a fairly fast response from previous research [2] which has a limitation in the form of a fairly long delay time, which is about 60 seconds, for each command sent through the Thingspeak cloud server using ESP8266. The first factors that affect the response speed of remote light control are the quality of the internet network, the internet network used both on the ESP32 and on smartphones greatly affects the response speed. Because if the internet network is interrupted, then the commands from the application received and executed by ESP32 cannot be processed.

Secondly from the ESP32 microcontroller, the processor and memory on the ESP32 also affect the data processing speed. If the ESP32 is overloaded with too much program, the data processing to respond becomes slow.

The last factor is the server condition of blynk, blynk is a cloud-based IoT control platform, so the response time also depends on how fast the blynk server processes the commands and sends them to the ESP32. If the server is congested or experiencing an outage, there may be a delay in receiving, processing and sending data to the ESP32 microcontrol.

CONCLUSION

Based on the results of the study, it can be concluded that the designed system has succeeded in fulfilling the main goal of real-time light control through the internet network. In addition, Blynk-based software is proven to be able to control the system remotely. The tests conducted showed that the system can operate stably in various internet network conditions with varying speeds, and has an average response time of 1.1 seconds.

For further development, it is suggested that some improvements and optimizations to this ESP32-based light control system are suggested. Namely the addition of a manual switch as an alternative to direct light control, manual switches can also increase ease of use, especially in the event of an internet connection or application interruption

REFERENCES

[1] M. Z. Mutaqin, M. E. Favian, and A. Kurniawan, “Sistem Smart Home Pemantauan dan Pengendalian Suhu Ruangan Menggunakan Arduino ESP32 Berbasis Green Energy,” *Jupiter : Journal of Computer & Information Technology*, vol. 5, no. 2, Art. no. 2, Aug. 2024, doi: 10.53990/jupiter.v5i2.355.

[2] I. Iswanto and G. Gandi, “Perancangan Dan Implementasi Sistem Kendali Lampu Ruangan Berbasis Iot (Internet Of Things) Android (Studi Kasus Universitas Nurtanio),” *Jurnal Teknologi Informasi dan Komunikasi*, vol. 8, no. 1, 2018, doi: 10.56244/fiki.v8i1.238.

[3] R. A. R. A. Mouha, “Internet of Things (IoT),” *Journal of Data Analysis and Information Processing*, vol. 09, no. 02, Art. no. 02, Apr. 2021, doi: 10.4236/jdaip.2021.92006.

[4] P. U. R. Rizdania, “Perbandingan Analisis Dan Implementasi Esp32 Dan Esp8266,” Nov. 2024.

[5] D. A. O. Turang, “Pengembangan Sistem Relay Pengendalian Dan Penghematan Pemakaian Lampu Berbasis Mobile,” 2015.

[6] S. P. Santosa and R. M. W. Nugroho, “Rancang Bangun Alat Pintu Geser Otomatis Menggunakan Motor DC 24 V,” vol. 9, no. 1, 2021.

[7] M. Nasir, Aswandi, Safriadi, N. Saputri, and N. Dwina, *INTERNET OF THINGS: Aplikasi dan Penerapan Mikrokontroler Arduino*. Penerbit Andi, 2024.

[8] A. S. Ismailov and Z. B. Jo, “Study of arduino microcontroller board,” *Science and Education*, vol. 3, no. 3, 2022.

[9] S. Anisah and A. D. Tarigan, “Efektivitas Pemamfaatan Lampu Penerangan Hemat Energi Pada Rumah Tinggal Masyarakat,” *Journal Of Electrical And System Control Engineering*, vol. 3, no. 2, Art. no. 2, Feb. 2020, doi: 10.31289/jesce.v3i2.3270.