e-ISSN: 2614-8404
p-ISSN: 2776-3234

Development of Smart Plant Watering System Application Based on Internet of Things

Adnan Nuur Bachtiar[1*)], Mochammad Faisal[2], Muhammad Darwis[3]
[1]Informatics Engineering Study Program, Faculty of Engineering Sciences, Paramadina University
[2] Informatics Engineering Study Program, Faculty of Engineering Sciences, Paramadina University
[3] Informatics Engineering Study Program, Faculty of Engineering Sciences, Paramadina University
Email: [1]adnan.bachtiar@students.paramadina.ac.id, [2]mochammad.faisal@students.paramadina.ac.id ,
[3]muhammad.darwis@paramadina.ac.id

Abstract – In 2024, air pollution levels in Jakarta were recorded as the highest in Southeast Asia, based on various air quality monitoring reports. This condition has become an increasingly alarming environmental issue, with pollution levels frequently exceeding the safe threshold set by the World Health Organization (WHO). One of the efforts by the Jakarta Provincial Government to address this problem is a free plant distribution program for the public. Plants play a crucial role in absorbing carbon dioxide, producing oxygen, and reducing airborne pollutant particles. However, plant maintenance—especially watering—poses its own challenges. Inefficient watering can cause plants to experience stress, wilt, or even die. With the advancement of technology, an automatic plant watering system based on the Internet of Things (IoT) presents a potential solution to improve the efficiency and sustainability of plant care. This study aims to develop a smart plant watering system application based on IoT that can automatically control watering based on real-time soil moisture levels. The system was applied to spider plants (Chlorophytum comosum) grown in pots with a diameter of 15 cm. By using an ESP32 microcontroller, a soil moisture sensor (Capacitive Soil Moisture Sensor v1.2), an air temperature and humidity sensor (DHT11), and a water pump, the system automatically activates watering when the soil moisture is below 55% and stops when it exceeds 65%. Sensor data is stored in a database and displayed through a web-based application for remote monitoring.

Keywords – Plant, Air Pollution, Internet of Things, Microcontroller, Soil Moisture

I. INTRODUCTION

Air pollution in Jakarta is an increasingly worrying environmental problem. Based on data from various sources, pollution levels in Jakarta often exceed the threshold set by the World Health Organization (WHO), mainly due to motor vehicle emissions and industrial activities [1], [2]. Poor air quality can have negative impacts on public health, such as increasing the risk of respiratory and cardiovascular diseases [3]. To address this problem, the DKI Jakarta Provincial Government (Pemprov) is planning a program to distribute free plants to the public to reduce air pollution in Jakarta [4].

Characteristics of plants that can reduce air pollution are green plants that can reduce CO2 in the air through photosynthesis. And to anticipate or adapt to global warming, among other things, it can be minimized through planting new plants [5]. Planting plants has many benefits, both for the environment and for humans themselves. Plants produce oxygen, absorb carbon dioxide, and help regulate air temperature. In addition, plants are also a source of food, medicine, and fuel. Historically, humans have depended on plants for their survival [6].

Some plants that can reduce air pollution such as Mother-in-law's Tongue (Sansevieria) which has a unique shape and can absorb carbon monoxide and carbon dioxide [7]. The Mother-in-law's Tongue (Sansevieria) plant can reduce carbon monoxide gas levels by 22.21 ppm [8], then there are the spider plant and the golden pothos (Scindapsus Aureus) which have been proven to have leaf surface characteristics that can collect PM2.5 indoors, microorganisms in the phyllosphere can detoxify PM2.5 [9].

Regular watering of plants is essential to ensure optimal plant growth. Water plays a role in the process of photosynthesis, helps transport nutrients, and maintains soil moisture so that plant roots remain healthy [10]. Plants that lack water can experience stress, wilt, and even die if left for a long period of time [11]. In addition, irregular or excessive watering can also cause other problems, such as plants being susceptible to disease due to being given too much water [12]. Therefore, an efficient watering method is needed that is in accordance with the needs of each type of plant to maintain ecosystem balance and environmental sustainability.

To ensure optimal plant watering, we can utilize technology to support the desired plant growth. One of them is by implementing an automatic plant watering system based on the Internet of Things (IoT) [13]. The Internet of Things (IoT) is one technology that can help to water plants intelligently. The Internet of Things (IoT) itself is a technology that connects many devices that are connected to each other using the internet to be able to communicate, connect, control, and exchange information with other devices [14].

Previous research has been conducted to build a smart farming system using the Internet of Things in the form of automatic plant watering, especially for chili plants (Capsicum annum L)[15] . Researchers use NodeMCU as the main processing unit, Driver Relay to control the water pump, and LCD (Liquid Crystal Display) to display soil conditions and pump status. Researchers get test results that the system works well. This study uses NodeMCU as the main unit for the main processing to read sensors, display data to the LCD screen, and drive the water pump.

The next research is an automatic watering system based on the Internet of Things (IoT) which aims to facilitate the care of ornamental plants, especially the golden pothos plant [16]. This study also uses NodeMCU as a link between hardware and systems. The sensors used are soil moisture sensors and air temperature sensors (DHT11), then integrating the system with Telegram to provide notifications to users regarding plant conditions and watering status. Furthermore, research conducted by Amsar Yunan (2022) entitled "Plant Watering Techniques Using Microcontrollers Based on the Internet of Things" also integrates the system with Telegram to inform users about the results of soil moisture sensor readings regarding dry or wet soil moisture conditions. Users set the soil moisture reading if the soil moisture is <65% then the soil will be declared dry, and the pump will turn on to water the plants. Then the pump will turn off when the soil moisture sensor reading gets a humidity result of >66% [17].

Previous research entitled "Implementation of Smart Monitoring and Control System for Eggplant Plants Based on Internet of Things with Drip Irrigation Method", the author used NodeMCU V3 microcontroller as an actuator and water pump controller, then used FC-28 Soil Moisture sensor to measure soil moisture, and DHT11 sensor to measure temperature and air humidity [18]. The author set the water pump on and off with reference if the soil moisture is below 50% then the pump will turn on, then if the soil moisture is at 60% then the pump will turn off. The system is integrated with an android-based application called Blynk then the data will be stored in the Thingspeak database. Similar research was also conducted by M. Iqbal Hasani (2023) where the author carried out automatic watering using a microcontroller and water level sensor to measure soil moisture [19]. then the system is integrated into an android-based application called Blynk so that users can monitor from anywhere. In addition, the author applies the Naive Bayes method for data analysis and decision making in plant watering where the accuracy level of the data analysis is 94.3%. "Implementation of the Internet of Things (IoT) in a Mobile-Based Plant Watering Automation System"

Based on the explanations above, a study entitled "Development of Smart Plant Watering System Application Based on Internet of Things" was conducted. This study aims to apply Internet of Things (IoT) technology to improve the efficiency of plant watering. IoT enables automation of plant watering by considering soil moisture levels in real-time, thus optimizing water usage.

This study also has advantages over previous studies, especially in terms of data presentation and system flexibility. The application developed uses a PHP-based web platform, so the interface can be adjusted to user needs without relying on third-party platforms such as Blynk or Telegram. In addition, this system records sensor data in real-time into a database, allowing users to review the complete history of sensor readings. This approach provides added value in terms of data control and a more flexible user experience.

II. RESEARCH METHODOLOGY

The development of the smart plant watering system application was developed based on the IoT Design Methodology proposed by Arshdeep Bahga and Vijay Madisetti in their book entitled "Internet of Things A Hands-On Approach" [20], [21], the stages in the Methodology are as in Figure 1.

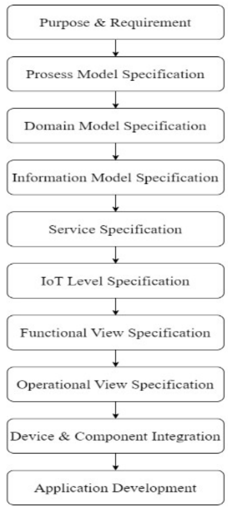

Fig 1. IoT Design Methodology

A. Purpose and Requirement

The purpose of this system is to automatically water plants using soil moisture reading data, as well as provide monitoring of temperature and humidity around the plants.

B. Process Model Specification

The interaction between the system and the user is modeled and described using Use Cases.

C. Domain Model Specification

In this stage, identify the entities and actors used in system development.

D. Information model Specification

In this stage, the data structure used by the system is explained. This structure includes attributes of data such as temperature, soil moisture, air humidity, pump status, and recording time. All are stored in the database and processed to be displayed in visual form.

E. Service Specification

System services are described based on Input and output. Services in this system are sensor data reading, automatic decision making in watering, sending data to the server, and automatic pump control.

F. IoT Level Specification

This system adopts a Hybrid/IoT Level 3 approach, where decision making and sensor reading are done locally on the ESP32, but data is sent to the Cloud Server.

G. Function View Specification

Grouping/organizing system functions into several modules, starting from the sensor module which contains soil moisture sensors and temperature and air humidity sensors, the control module which contains rules for controlling the pump on and off, the communication module, and the display module.

H. Operational View Specification

This stage explains the implementation of services and system management, where the system is set to read sensors in real-time and activate the pump when the activation requirements are met.

I. Device and Component Integration

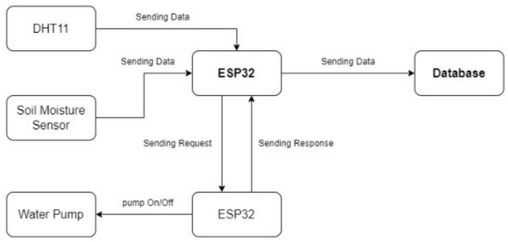

Fig 2. Schema Component

This stage combines hardware components, including sensors, microcontrollers, and actuators into an IoT system that is interconnected and runs automatically and in real-time.

J. Application Development

In this stage, application development and coding are carried out. In this stage, there are 2 codings, namely coding for the microcontroller and coding for the interface. Microcontroller coding is done using the C++ programming language on the Arduino IDE platform, where the ESP32 is programmed to be able to read the results of soil moisture sensors and DHT11 sensors, and make decisions in watering plants and send sensor readings and decisions to the server using the HTTP protocol. Then the interface coding uses the PHP programming language to display data on the server in real time to the user.

III. RESULTS AND DISCUSSION

A. Purpose and Requirement

The purpose of this research is to develop a smart plant watering system. Here are some things that underlie the research "Development of a Smart Plant Watering System Based on the Internet of Things"

a. Purpose

To develop an IoT-based watering automation system that uses soil moisture and temperature sensors to automatically control plant watering to ensure efficient water usage.

b. Behavior

This system measures soil moisture and temperature to determine irrigation needs, so water is only delivered when needed.

c. System Management Requirement

The system must be able to provide remote monitoring and control of irrigation through an easily accessible management interface. In addition, the system must be able to read the air temperature and humidity around the plants to provide additional information to the user.

d. Data Analysis Requirement

The system must be able to analyze soil moisture data to provide efficient irrigation recommendations. All are set in such a way as to provide minimum and maximum soil moisture thresholds.

e. Application Deployment Requirement

This application is implemented with a cloud server and can be accessed from any device connected to the internet where it can be accessed at any time. Then the system sends sensor data in real time.

f. Security

The application must have a user login feature to perform authentication and data protection to maintain data privacy and security.

B. Process Model Specification

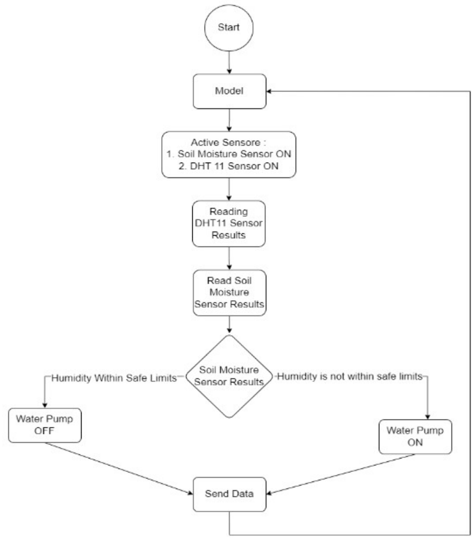

Fig3. Flowchart

In Figure 3. The system uses an automatic processing flow that utilizes a soil moisture sensor to automatically activate the water pump when soil moisture is below a certain threshold, then utilizes the DHT11 sensor to help monitor the air temperature and humidity around the plants.

C. Domain Model Specification

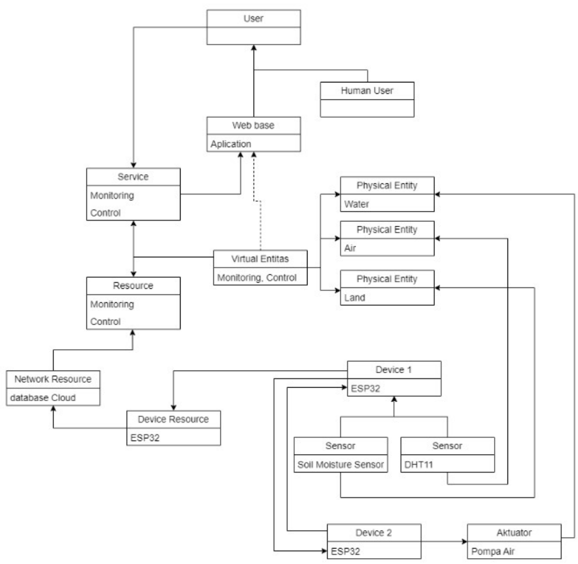

Fig 4. Domain Model Specification

e-ISSN: 2614-8404
p-ISSN: 2776-3234

In Figure 4. It is a Domain Model to describe the relationship between entities. In this stage, the important components of the system are determined and classified based on the five elements of the IoT domain model: Physical entities, virtual entities, devices, resources, and services.

a. Physical Entity
  - Land (Dependent variable)
    The main entity whose humidity is monitored and controlled by the system is using a soil moisture sensor.
  - Water
    The main entity used to water the soil if the sensor detects that the soil moisture is below the required safe threshold.
  - Air
    The entity used to find out the temperature and humidity of the air around the plant. This entity will be monitored by DHT11.

b. Virtual Entity
  - Monitoring
    Refers to the process of monitoring soil moisture levels and temperature through sensors.
  - Controlling
    Refers to the system's ability to turn the water pump on and off as needed based on soil moisture sensor data.
  - Alerting
    Refers to the system's ability to notify the user about the soil moisture level.

c. Device
  - Soil Moisture Sensor
    Sensors used to detect soil moisture levels in real-time.
  - DHT11 Sensor
    Sensors that measure air temperature and measure air humidity around the land to provide additional data in system needs analysis.
  - ESP32 Microcontroller
    The control unit that receives data from sensors, performs initial processing, and sends the data to the database.
  - Water pump
    A unit for pumping water so that it can be sprinkled onto areas of land that require watering.

d. Resource
  - ESP32
    Running Firmware that manages sensor readings and decision making locally.
  - Network
    The system uses a cloud database to store sensor data periodically and provides access to monitoring and visualization services, these services are supported by a PHP-based website interface hosted on a server that allows user interaction with the system.

e. Service
  - Controlling Service
    Monitor sensor data and regulate irrigation system conditions according to specified rules. A service that manages water pump activity based on data received from sensors.
  - Sensor Service
    Read humidity and temperature values, send data to cloud database, and perform real-time data analysis for monitoring via web.

D. Information Model Specification

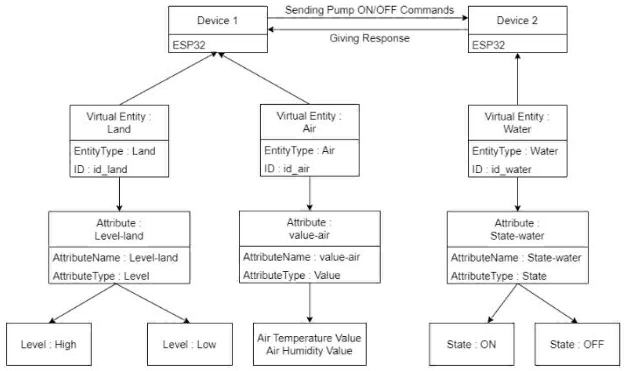

Fig 5. Information Model Specification

In Figure 5. there are 3 main virtual entities that represent the physical entities involved, namely Land, Air, and Water. The Land entity has a level-humidity attribute of type Level. Which level is obtained from the results of the Moisture Sensor reading then the reading results are sent to device 1. Then the Air Entity has a value-air attribute that contains information on temperature and humidity. Data is obtained from the DHT11 sensor which is sent to device 1. Finally, there is a Water entity of type state-air. This entity represents the status of the water pump controlled by device 2 with a value of ON or OFF. Where device 2 gets commands from device 1 from the results of reading the soil moisture sensor.

E. Service Specification

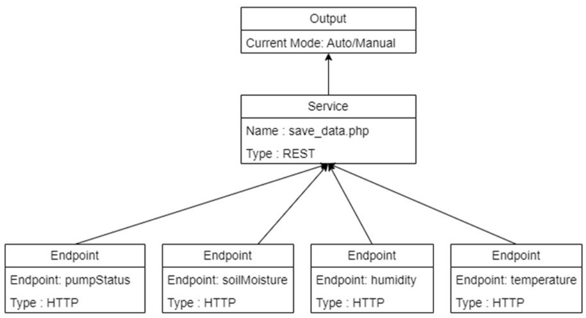

Fig 6. Service Specification

In Figure 6 is the Service Specification of the system, this system uses the HTTP protocol to send data from the device to the server. Where sensor reading data and pump status are sent via URL with parameters such as in the endpoint with the service name save_data.php. This allows data to be sent in real-time to the server automatically.

e-ISSN: 2614-8404
p-ISSN: 2776-3234

F. IoT Level Specification

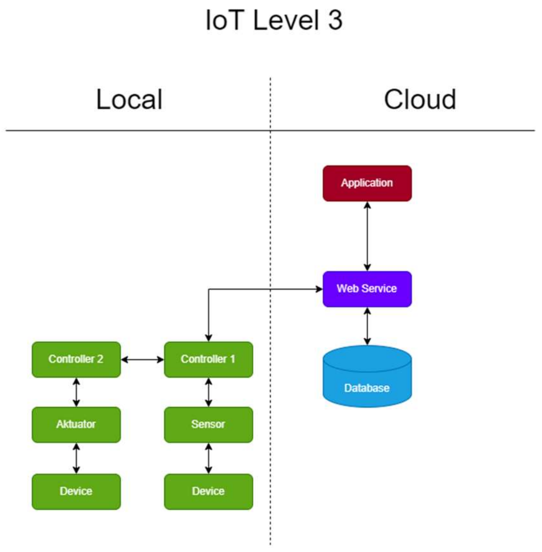

Fig 7. IoT Level Specification

In Figure 7, the author uses 2 controllers with the same tasks. Controller 1 reads the soil moisture sensor and DHT11 sensor, sends a command to controller 2 to turn on/off the pump depending on the soil moisture reading, then sends the data into the database. Controller 2 is tasked with receiving commands from controller 1 to turn on the Actuator and then sending a response back to controller 1. And the data is displayed in the web-based application that has been created.

G. Functional View Specification

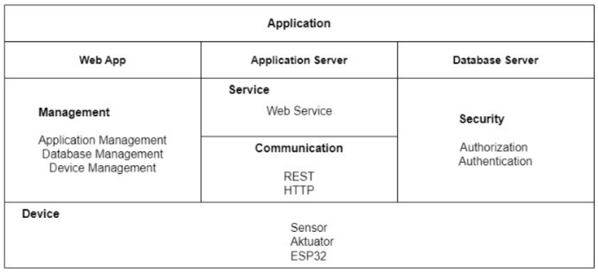

Fig 8. Function View Specification

In Figure 8, the division of roles and functions of each component. Where the device layer is a physical device such as sensors, actuators, and ESP32 with communication using REST/HTTP allows communication between the controller and the web service. Using a web-based application with the PHP programming language and MySQL database.

H. Operational View Specification

Table 1. Operational View Specification

Application	Database Service : MySQL Application Server : Website, PHP
Management	MySQL, Website, PHP
Security	Authentication : Databases Authorization : Database
Service	Web Service
Communication	REST, HTTP
Devices	Controller : ESP32 Sensor : Soil Moisture Sensor, DHT11 Aktuator : Water Pump

The following table 1 shows various operational options related to the deployment and operation of the IoT system. This stage is important to ensure that all system components can work in an integrated manner. MySQL is chosen as the database server because it is the most famous database service. The application runs on a web base with the PHP programming language.

I. Device and Component Integration

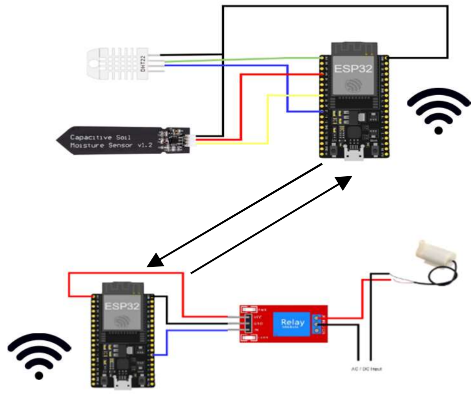

Fig 9. Device & Component Integration

Figure 9 shows a series of images in the development of this system. The author uses 2 controllers that communicate with each other. The first controller is tasked with reading the results of the soil moisture sensor and the temperature and air humidity sensor, where the safe soil humidity for the spider plant is between 30 - 50%. In the first controller determines whether the soil humidity is at a safe threshold or not, then the controller sends a command to the second controller to turn the water pump on or off. In sending the results of the sensor reading and pump execution, the first controller performs the task. The first controller sends data using REST HTTP into the cloud database that has been provided.

J. Application Development

At the application implementation stage, the author took several steps to implement the application starting from creating a web-based application using PHP and a database using MySQL, installing and configuring controllers and sensors, installing sensors and devices, and visualizing and analyzing data.

1. Application Development

In developing the application, the author developed a web-based application using the PHP programming language and Bootstrap 5 as the UI.

Fig 10. website appearance

Figure 10 shows the display results of the application, where users can control the reading results from the installed sensors, then can find out the statistics of plant soil moisture in real time.

2. Hardware assembly and integration

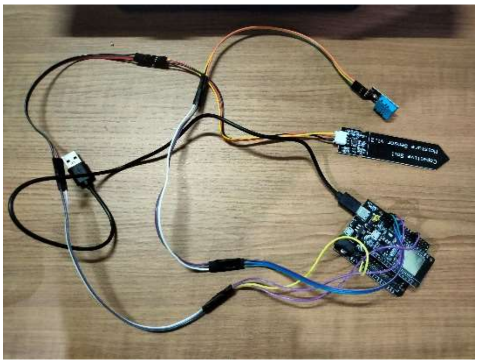

Fig 11. controller assembly 1

In figure 11 the hardware is assembled, controller 1 is connected to the soil moisture sensor and the DHT11 sensor.

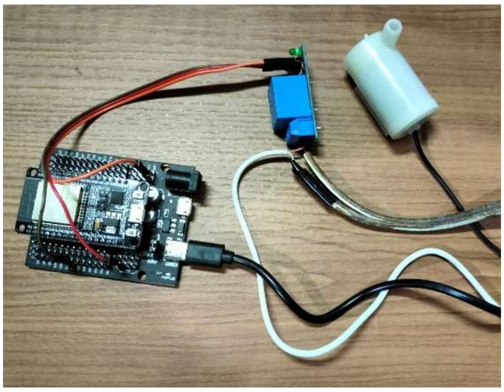

Fig 12. controller assembly 2

Then in Figure 12, controller 2 is connected to a relay, which will conduct electricity to turn on the water pump.

3. ESP32 Programming

ESP32 is programmed using the C++ programming language on the Arduino IDE platform, where Controller 1 is programmed to read the soil moisture sensor and DHT11 sensor then sends a command to controller 2 to turn the water pump on or off and after that controller 1 sends data to the database using HTTP. Controller 2 is programmed to receive commands from controller 1 to turn on the pump. Figure 13 is a script to send commands to turn the pump on and off by controller 1.

```
// Rute untuk menerima perintah dari ESP-A
server.on("/pump", HTTP_GET, []() {
    if (server.hasArg("status")) {
        String pumpStatus = server.arg("status");

        if (pumpStatus == "ON") {
            digitalWrite(RELAYPIN, HIGH);
            Serial.println("Pompa HIDUP!");
            delay(5000);  // Pompa hidup 5 detik
            digitalWrite(RELAYPIN, LOW);
            Serial.println("Pompa MATI!");
            server.send(200, "text/plain", "Pompa ON (5 detik)");
        } else if (pumpStatus == "OFF") {
            digitalWrite(RELAYPIN, LOW);
            Serial.println("Pompa MATI!");
            server.send(200, "text/plain", "Pompa OFF");
        } else {
            server.send(400, "text/plain", "Perintah tidak valid");
        }
    } else {
        server.send(400, "text/plain", "Parameter status tidak ditemukan");
    }
});

server.begin();
```

Fig 13. Script ESP32

K. Application Testing

This stage is carried out after the application has been successfully developed, system testing using blackbox testing to determine the success of the application function as desired. Some of the tests carried out are explained in table 2.

Table 2. Blackbox Testing

No	Description	Expected results	Results
1	Controller 1 sends sensor and pump reading data into the database.	Controller 1 can send data into the database, and the database can receive reading results from the controller.	Succeed
2	Controller 2 turns on the water pump	Controller 2 receives a command from controller 1 to turn on the water, and the controller turns on the water pump.	Succeed
3	Pump runs for 5 seconds	When controller 2 gets the on command, the pump will turn on for 5 seconds, then turn off again automatically.	Succeed
4	Data is displayed on the website	The website can display data that has been received by the database.	Succeed

IV. Conclusion

The use of the Internet of Things (IoT) system in automatic plant watering can be used by users to control the irrigation of their plants so that plants get water on time and according to plant needs by utilizing a microcontroller to read sensors and turn on and off the water pump in real-time then send data into a MySQL database to be displayed via a PHP-based website application. This system can be further developed by adding notification features, and adding sensors for plant maintenance..

References

[1] "NAFAS Indonesia." Accessed: Mar. 08, 2025. [Online]. Available: https://nafas.co.id/article/Bagaimana-dengan-kualitas-udara-Jakarta

[2] "Kualitas Udara Jakarta Terburuk Nomor 2 Dunia, Warga Disarankan Pakai Masker - Nasional Katadata.co.id." Accessed: Mar. 08, 2025. [Online]. Available: https://katadata.co.id/berita/nasional/66e795e7b56ff/kualitas-udara-jakarta-terburuk-nomor-2-dunia-warga-disarankan-pakai-masker

[3] R. Umah and E. Gusmira, "Dampak Pencemaran Udara Terhadap Kesehatan Masyarakat di Perkotaan," *Profit: Jurnal Manajemen, Bisnis dan Akuntansi*, vol. 3, no. 3, pp. 103–112, 2024.

[4] "Di Balik Keputusan Pemprov DKI Pakai Lidah Mertua sebagai Solusi Polusi – Jakarta Berketahanan." Accessed: Mar. 10, 2025. [Online]. Available: https://lingkunganhidup.jakarta.go.id/jakartaberketahanan/?p=16225

[5] M. D. Udin, I. Istiadi, and F. Rofii, "Aquascape Dengan Kontrol Fotosintesis Buatan Pada Tanaman Air Menggunakan Metode Kendali Logika Fuzzy," *Transmisi: Jurnal Ilmiah Teknik Elektro*, vol. 23, no. 3, pp. 103–111, 2021.

[6] G. Viroli, A. Kalmpourtzidou, and H. Cena, "Exploring benefits and barriers of plant-based diets: Health, environmental impact, food accessibility and acceptability," *Nutrients*, vol. 15, no. 22, p. 4723, 2023.

[7] B. Putra, K. Exaudi, R. Passarella, and P. Sari, "Pemantauan Penyerapan Gas Beracun Menggunakan Tanaman Sansevieria Pada Ruangan Tertutup," *Komputika : Jurnal Sistem Komputer*, vol. 11, pp. 95–102, Feb. 2022, doi: 10.34010/komputika.v11i1.4751.

[8] K. P. Cahyanti and D. A. A. Posmaningsih, "Tingkat kemampuan penyerapan tanaman sansevieria dalam menurunkan polutan karbon monoksida," *Jurnal Kesehatan Lingkungan (JKL)*, vol. 10, no. 1, 2020.

[9] A. D. Haq, A. R. Rahim, and L. A. D. Rahayu, "Potensi sirih gading (Epipremnum aureum) dan lili paris (Chlorophytum comosum) sebagai sarana fitoremediasi PM2. 5 di dalam ruangan," *Jurnal Kedokteran*, vol. 10, no. 1, pp. 347–354, 2021.

[10] A. Sumarianti, K. D. Jayanti, and Y. Tanari, "Pengaruh frekuensi penyiraman terhadap pertumbuhan dan hasil bawang merah (Allium cepa L.)," *Agrovigor: Jurnal Agroekoteknologi*, vol. 15, no. 1, pp. 39–43, 2022.

[11] C. A. Nugroho and A. W. Setiawan, "Pengaruh frekuensi penyiraman dan volume air terhadap pertumbuhan tanaman sawi pakcoy pada media tanam campuran arang sekam dan pupuk kandang," *AGRIUM: Jurnal Ilmu Pertanian*, vol. 25, no. 1, pp. 12–23, 2022.

[12] D. E. Nadindra and J. C. Chandra, "Sistem IoT penyiram tanaman otomatis berbasis Arduino dengan kontrol Telegram," *SKANIKA: Sistem Komputer dan Teknik Informatika*, vol. 5, no. 1, pp. 104–114, 2022.

[13] R. Alamsyah, E. Ryansyah, A. Y. Permana, and R. Mufidah, "Sistem Penyiraman Tanaman Otomatis Menggunakan Logika Fuzzy dengan Teknologi Internet of Things Berbasis ESP8266 dan Aplikasi Blynk," *Jurnal Informatika dan Teknik Elektro Terapan*, vol. 12, no. 2, 2024.

[14] A. Selay *et al.*, "Internet Of Things," *Karimah Tauhid*, vol. 1, no. 6, pp. 860–868, 2022.

[15] I. W. B. Darmawan, I. N. S. Kumara, and D. C. Khrisne, "Smart garden sebagai implementasi sistem kontrol dan monitoring tanaman berbasis teknologi cerdas," *Jurnal SPEKTRUM Vol*, vol. 8, no. 4, pp. 161–170, 2021.

[16] P. A. Wulandari, P. Rahima, and S. Hadi, "Rancang Bangun Sistem Penyiraman Otomatis Berbasis Internet of Things Pada Tanaman Hias Sirih Gading," *Jurnal Bumigora Information Technology (BITe)*, vol. 2, no. 2, pp. 77–85, 2020.

[17] A. Yunan, S. Safriati, and H. Hermalinda, "Teknik Penyiraman Tanaman Menggunakan Mikrokontroler Berbasis Internet of Things," *Journal of Information System Research (JOSH)*, vol. 3, no. 3, pp. 331–337, 2022.

[18] E. Anugrah, M. Hasbi, and M. P. Lukman, "Penerapan Sistem Monitoring Dan Kendali Pintar Untuk Tanaman Terung Berbasis Internet of Things Dengan Metode Penyiraman Irigasi Tetes," *Jurnal RESISTOR (Rekayasa Sistem Komputer)*, vol. 4, no. 2, pp. 204–212, 2021.

[19] M. I. Hasani and S. Wulandari, "Implementasi Internet of Things (IoT) Pada Sistem Otomatisasi Penyiraman Tanaman Berbasis Mobile," *ILKOMNIKA*, vol. 5, no. 3, pp. 149–161, 2023.

[20] M. Darwis, H. A. Al Banna, S. R. Aji, D. Khoirunnisa, and N. Natassa, "IoT Based Early Flood Detection System with Arduino and Ultrasonic Sensors in Flood-Prone Areas," *JURNAL TEKNIK INFORMATIKA*, vol. 16, no. 2, pp. 133–140, 2023.

[21] "Metodologi Desain IoT - Tutorial IoT untuk Pemula - Startertutorials." Accessed: May 07, 2025. [Online]. Available: https://www.startertutorials.com/blog/iot-design-methodology.html