ISSN 2089-8673 (Print) | ISSN 2548-4265 (Online)
Volume 13, Issue 3, December 2024

IMPLEMENTATION OF A WEB-BASED MASTER-SLAVE ARCHITECTURE FOR GREENHOUSE MONITORING SYSTEMS IN GRAPE CULTIVATION

Hirzen Hasfani[1], Uray Ristian[2], Uray Syaziman Kesuma Wijaya[3]

[1,2,3] Department of Computer System Engineering, Tanjungpura University

email: hirzen.hasfani@siskom.untan.ac.id[1], eristian@siskom.untan.ac.id[2], h1051201051@student.untan.ac.id[3]

Abstract

The Internet of Things (IoT) technology enables electronic devices to connect to the Internet for real-time data collection and analysis. In greenhouses, IoT is used to monitor critical environmental parameters such as soil moisture, temperature, and humidity to optimize plant care, especially for grape cultivation. However, IoT systems that rely on Wi-Fi networks often face issues like packet loss and transmission delays due to network congestion. This project presents the development and testing of a prototype grape plant monitoring system utilizing a master-slave architecture designed to mitigate these challenges. The system leverages the ESP-NOW protocol to reduce dependence on Wi-Fi networks by allowing direct, low-power communication between sensor nodes. The prototype was tested in a real-world greenhouse environment, where it successfully monitored soil moisture and other environmental parameters while addressing common issues associated with wireless data transmission. Testing results demonstrated that the system achieved an average delay of 1,546.65 ms, jitter of 120.56 ms, and a packet loss rate of just 0.07% across 88,815 data transmissions. These findings confirm the system's reliability and effectiveness for real-time monitoring, offering a practical and scalable solution for improving data transmission stability in controlled agricultural environments. The prototype demonstrates the feasibility of implementing a more efficient communication for IoT-based smart farming systems.

Keywords : Internet of Things, Master-Slave, Packet Loss, Delay

Received: 12-08-2024 | **Revised:** 11-10-2024 | **Accepted:** 15-10-2024
DOI: https://doi.org/10.23887/janapati.v13i3.84105

INTRODUCTION

The Internet of Things (IoT) technology is an innovation that enables electronic devices to connect to the Internet, allowing for real-time data collection and analysis [1][2][3]. The application of IoT has rapidly expanded across various sectors, including agriculture [4]. In agriculture, IoT is used to monitor soil conditions through moisture sensors [5]. The data collected enables the activation of automatic irrigation systems, which aid in watering and plant care, thereby enhancing productivity [6]. Additionally, IoT-based monitoring systems are also implemented in greenhouse management.

A greenhouse is a constructed structure designed to regulate and modify environmental parameters to create ideal conditions for plant cultivation [7][8]. In the context of grape cultivation in tropical climates, which tend to be challenging, the use of greenhouses has emerged as a promising solution [9]. Greenhouses provide essential control over environmental conditions, overcoming the challenges faced by certain crops in climates that are less conducive to growth [10][11]. By utilizing greenhouses supported by technology, farmers can create a more controlled and optimal environment for plant growth in tropical climates, particularly for grape cultivation.

The monitoring system for grape plants in greenhouses faces challenges such as packet loss and delay, particularly when a large volume of data is transmitted simultaneously from each node. The limitations of WiFi networks in handling multiple simultaneous internet access requests result in packet loss and high delays due to inefficient data transmission management [12]. Therefore, a solution capable of managing data transmission more efficiently is needed to maintain the system's optimal performance. One potential solution to address this issue is the implementation of a master-slave architecture in the monitoring system. In this approach, one node functions as the master, responsible for collecting data from other nodes acting as slaves. The master then transmits the aggregated data to the central server.

Master-slave is a communication model in which one device or process has direct control over one or more other devices [13]. In the master-slave paradigm, there are two main entities: the master and the slaves [14]. The master node functions as the primary manager that collects and organizes data, while the slave nodes are responsible for reading data from sensors [15]. By implementing a master-slave architecture, sensor data can be centralized at the master node, which reduces direct dependency on the WiFi access point and minimizes the potential for packet loss and delay. The master node is responsible for sending the centralized data to the server, thereby reducing the network load for transmitting sensor data. This approach ensures more efficient data transmission and improves the performance of the grape plant monitoring system in the greenhouse.

Numerous previous studies have discussed monitoring systems and master-slave architecture. One study explored the use of a master-slave model for controlling lights with pyroelectric infrared (PIR) sensors to conserve electricity. This system was installed in multiple zones using Wireless Sensor Networks (WSN) [16]. Another study investigated the use of master-slave architecture in a robotic arm, where communication between the master and slave nodes was conducted using the HC-05 Bluetooth module. This setup allowed the movement of the master arm by the user to be mirrored by the slave robotic arm. The data collected from the master and slave controllers were stored in Firebase Cloud and displayed through the MIT App Inventor on Android devices [17]. Another study examined a monitoring system for grape plants without employing a master-slave architecture. This system measured soil moisture, air temperature, air humidity, and water pH levels, and included a scheduled automatic irrigation mechanism. The data from these sensors were stored in a web-based application, Sipunggur, enabling real-time monitoring of the plant's environmental parameters [9].

In this project, a master-slave architecture is utilized for the collection of sensor data from multiple slave nodes. The slave nodes function by reading data from the sensors and transmitting the collected data to the master node [18]. To reduce dependency on WiFi networks, the ESP-NOW protocol is used for communication between the slave nodes and the master node [19]. The ESP-NOW protocol allows data to be transmitted wirelessly from the slave nodes to the master node without passing through the WiFi access point, thereby reducing the load on the WiFi network and enhancing communication efficiency [20][21]. Subsequently, the master node, which is connected to the WiFi access point, is responsible for sending all the collected sensor data to the central server. This approach ensures more efficient sensor data transmission and reduces the potential dependency on the WiFi network, which can lead to packet loss and delay.

METHOD

This project was conducted at the IoT Smart Farming Lab, FMIPA Untan, where grape plants are cultivated in multiple pots. The monitoring system developed utilizes a master-slave communication model, specifically designed to efficiently collect and manage data from various nodes distributed throughout the greenhouse environment. In this configuration, the master node is responsible for monitoring key environmental parameters, such as air temperature and humidity around the plants. Additionally, the master node gathers soil moisture data from slave nodes placed among the plant pots, ensuring comprehensive monitoring.

The system architecture employs the ESP-NOW protocol for communication between the master node and the slave nodes. ESP-NOW is a low-power, peer-to-peer communication protocol known for its ability to transmit data quickly and reliably without requiring an internet connection. This feature is particularly advantageous in scenarios where minimizing dependence on Wi-Fi networks is crucial, such as in controlled agricultural environments like a greenhouse. However, since the Wi-Fi module on the master node is fully utilized for communication with the slave nodes, the master node cannot directly connect to a Wi-Fi access point to send data to the server.

To overcome this limitation, a serial node was added to the system as an intermediary, physically connected to the master node via RX and TX pins. The serial node serves as a critical bridge, receiving the collected data from the master node and then forwarding it to the central server. This design ensures a continuous and uninterrupted flow of sensor data, including essential parameters such as air temperature, humidity, and soil moisture levels, to the server. The uninterrupted data transmission is vital for enabling real-time monitoring of plant conditions, accessible through a web interface, and providing a user-friendly and responsive platform for managing the greenhouse environment.

The combination of the master-slave architecture with the ESP-NOW communication protocol significantly optimizes this system for use

ISSN 2089-8673 (Print) | ISSN 2548-4265 (Online)
Volume 13, Issue 3, December 2024

in environments with numerous sensor nodes, such as greenhouses. By aggregating all sensor data at the master node and then transmitting it via the serial node to the server, the system effectively reduces the risk of data packet loss and delay. These issues are common in traditional Wi-Fi-based IoT systems, where network congestion and inefficient data management can lead to substantial performance degradation. A schematic overview of the grapevine monitoring system based on the master-slave architecture can be seen in Figure 1.

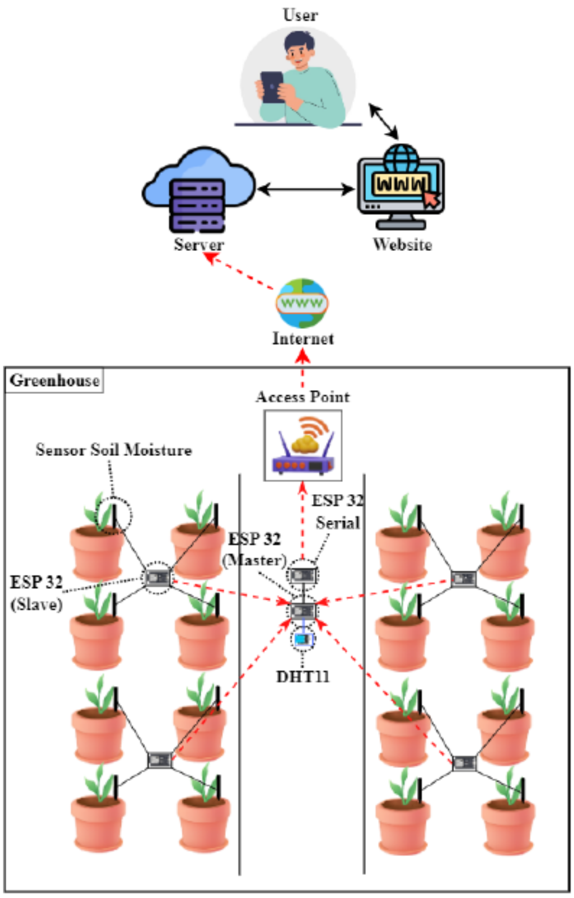

Figure 1. Grapevine Monitoring System Description Based on Master-Slave Architecture

Design of Communication System Between Master and Slave Nodes

The communication system design between the master and slave nodes is implemented using the ESP-NOW protocol. This protocol enables multiple devices to communicate without relying on Wi-Fi, utilizing a low-power 2.4GHz wireless connectivity concept. To associate the master and slave nodes, they must be connected through their respective MAC addresses, where the master's MAC address is registered on the slave node. Sensor data transmission from the slave node to the master node is done sequentially by each slave node, ensuring no data collision occurs. By leveraging the ESP-NOW protocol, this system can be implemented in various environmental conditions without the need for complex Wi-Fi network infrastructure. The design of the master-slave communication system is illustrated in Figure 2.

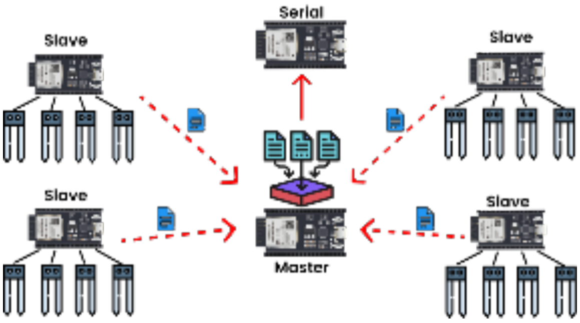

Figure 2. The Design of a Master-Slave Communication System

Hardware Design

In the hardware design, there are slave nodes and a master node, both utilizing the ESP32 as the primary platform. Each slave node is equipped with a soil moisture sensor to measure the soil moisture in each grape plant pot. These sensors continuously monitor soil moisture and wirelessly transmit the data to the master node using the ESP-NOW protocol. This protocol ensures efficient and reliable communication between the slave and master nodes without requiring Wi-Fi connectivity. The system monitoring design for the slave nodes is illustrated in Figure 3.

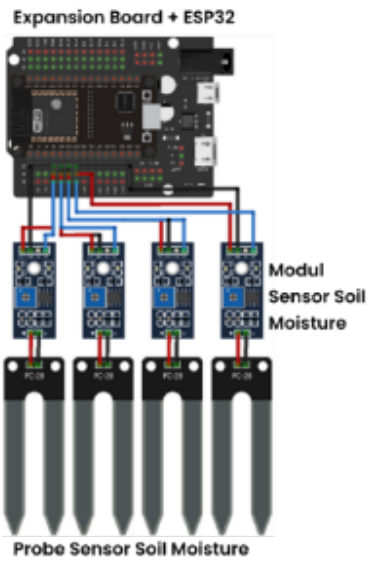

Figure 3. Design of the Monitoring System on Slave Nodes

In the master node design, a DHT11 sensor is included to measure the temperature and humidity within the greenhouse. Additionally, a serial node is connected to the master node via a cable, serving to forward data from the master node to the server. This setup is necessary because when the master node is connected to the slave nodes using the ESP-NOW protocol, the Wi-Fi module on the master node is fully utilized for this communication, preventing it from connecting to Wi-Fi through an access point. Therefore, the serial node acts as an intermediary, ensuring that data can be sent to the server without disrupting the communication between the master and slave nodes. The system monitoring design for the master node is illustrated in Figure 4.

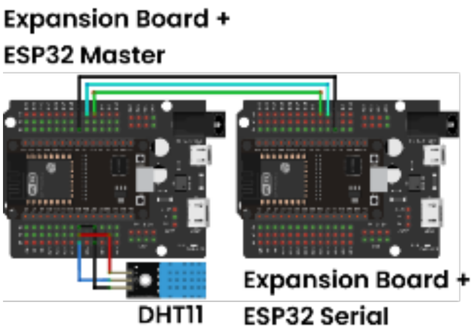

Figure 4. Design of the Monitoring System on Master Nodes

Software Design

The software design in this research aims to serve as the bridge between the slave nodes, the master node (an ESP32), and the server. Based on the hardware design, the conceptualized software functions to process information related to data transmission, including delay, jitter, and packet loss from the master node to the server. Sensor data from the slave nodes is first sent to the master node, where four soil moisture sensors on each slave node measure soil moisture (SM). This data is collected at the slave node and then transmitted to the master node.

In addition to receiving data from the slave nodes, the master node is equipped with a DHT11 sensor to measure the temperature (TU) and humidity (HU) in the greenhouse. Once all the data is collected at the master node, it is forwarded to the serial node for transmission to the server. The data sent to the server is stored in a database and displayed on the sensor data dashboard, along with information on sensor data transmission. The flowchart for reading soil moisture sensors on the slave nodes is shown in Figure 5. The flowchart for reading the air temperature and humidity sensors on the master node is shown in Figure 6.

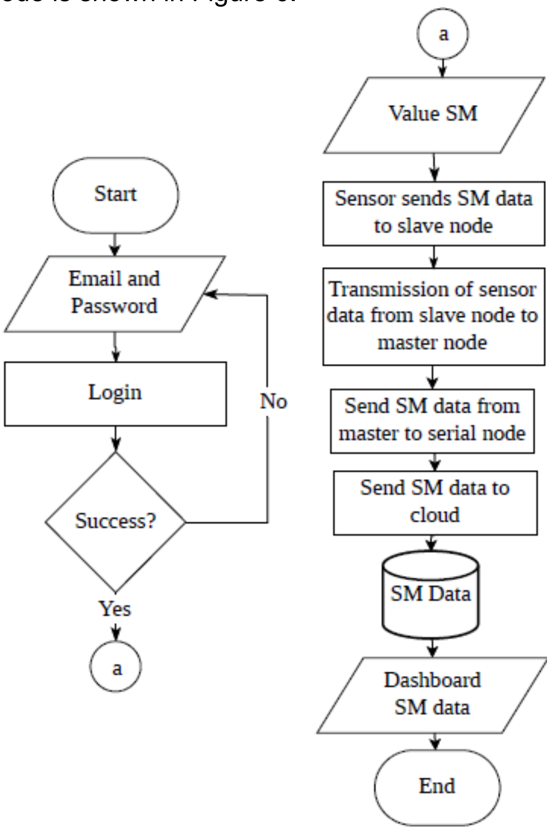

Figure 5. Flowchart of Soil Moisture (SM) Sensor Reading on Slave Node

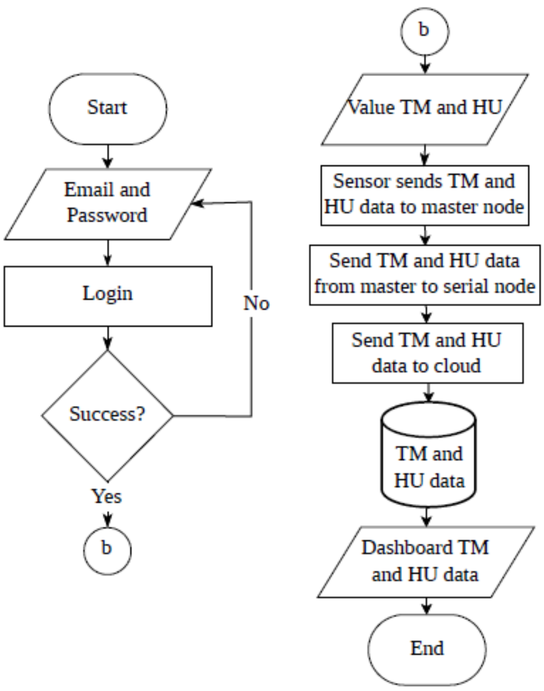

Figure 6. Flowchart of Air Temperature (TM) and Humidity (HU) Sensor Reading on Master Node

Figure 7. Implementation of the System on Slave Node

RESULT AND DISCUSSION

In this greenhouse monitoring system, the implementation was carried out on both slave and master nodes. There are four slave nodes, each equipped with four soil moisture sensors connected to an ESP32 microcontroller. These sensors are designed to measure the soil moisture levels in the grape plant pots, providing critical data for accurately monitoring the plants' conditions. Once the sensors are inserted into the soil, the system automatically begins continuous soil moisture readings, which are then processed by the ESP32 on each slave node to ensure data quality and consistency.

After the soil moisture data is collected, each slave node wirelessly transmits the data to the master node in a sequential manner. This staggered transmission process is carefully designed to ensure that data is transmitted without interruption or delay, reducing the risk of interference that can occur in wireless communication. This is crucial for maintaining the integrity of the transmitted data and ensuring that the information received by the master node is always accurate and up-to-date. Implementation of the system on the slave node is illustrated in Figure 7.

On the master node, a DHT11 sensor is used to measure air temperature and humidity within the greenhouse. Additionally, a serial node functions as an intermediary for transmitting the collected sensor data from the master node. This is necessary because when the master node is in the process of receiving sensor data from the slave node using the ESP-NOW protocol, it cannot directly send data to a server. To address this issue, the serial node is used by connecting the master node through cables to RX and TX pins. The implementation of the system on the master node is illustrated in Figure 8.

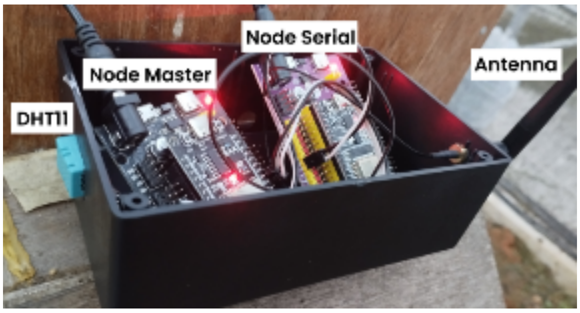

Figure 8. Implementation of the System on Master Node

In implementing the communication system between master and slave nodes, data is transmitted from slave nodes and received by the master node using the ESP-NOW protocol. This protocol enables direct communication between ESP devices without requiring a Wi-Fi network or internet connection. Each slave node must register the MAC address of the master node to ensure proper communication, allowing the slave nodes to sequentially transmit sensor data wirelessly to the master node. Once all data is collected at the master node, it is forwarded to a serial node connected to Wi-Fi via an access point, enabling the serial node to transmit the data to a server. The implementation of communication between master and slave nodes is illustrated in Figure 9.

ISSN 2089-8673 (Print) | ISSN 2548-4265 (Online)
Volume 13, Issue 3, December 2024

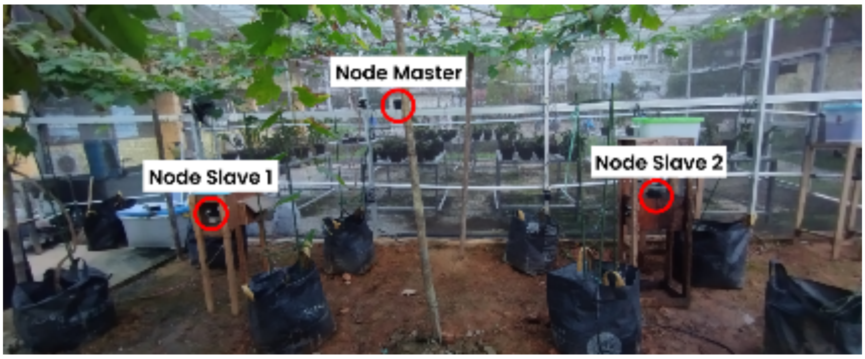

Figure 9. Implementation of Communication Between Master and Slave Nodes

A web-based application is used to assist in monitoring the grapevine greenhouse, aiming to create an optimal environment for plant growth. The application provides real-time soil moisture data for each grapevine pot, as well as air temperature and humidity data within the greenhouse. The web dashboard for the grapevine monitoring system is shown in Figure 10.

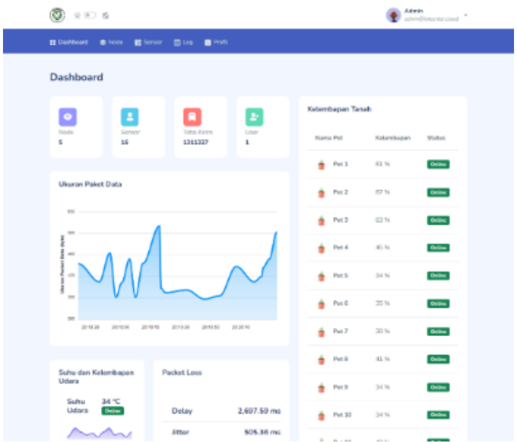

Figure 10. Web Dashboard for the Grapevine Monitoring System

Additionally, the application monitors system parameters such as delay, jitter, and packet loss within the grapevine monitoring system. This ensures that the data transmitted by sensors is received accurately and promptly, supporting better decision-making in greenhouse management. The log page for checking delay, jitter, and packet loss is shown in Figure 11.

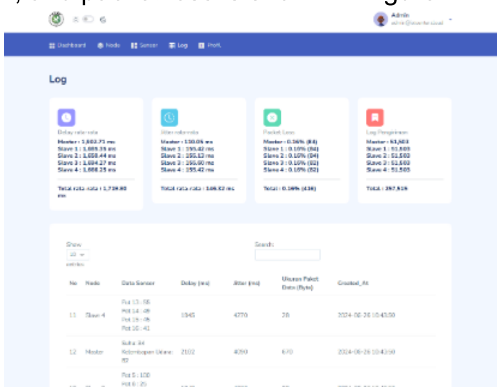

Figure 11. The Delivery Log Page Interface

In the testing scheme, a slave node transmits soil moisture sensor data to a master node at 250 ms intervals. The master node retrieves data from the slave node every 2000 ms. Data transmission occurs wirelessly via the ESP-NOW protocol. After collecting data from all slave nodes, the master node processes it into a single batch and sends it to a serial node through a wired connection using RX and TX pins. The serial node then transmits the data to the server. Delay measurement starts from the sensor reading until the data becomes accessible on the web server. The delay and jitter testing graph on the master node, recorded every half hour, is shown in Figure 12.

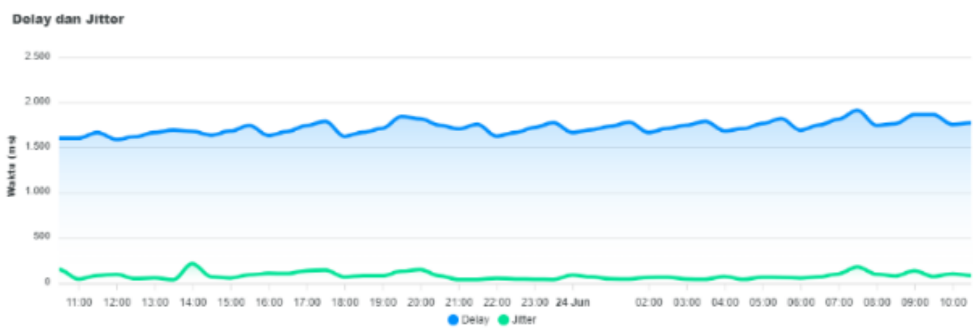

Figure 12. Delay and Jitter Testing Graph on the Master Node

ISSN 2089-8673 (Print) | ISSN 2548-4265 (Online)
Volume 13, Issue 3, December 2024

Figure 13. Graph of Transmission Count and Packet Loss Testing on the Master Node

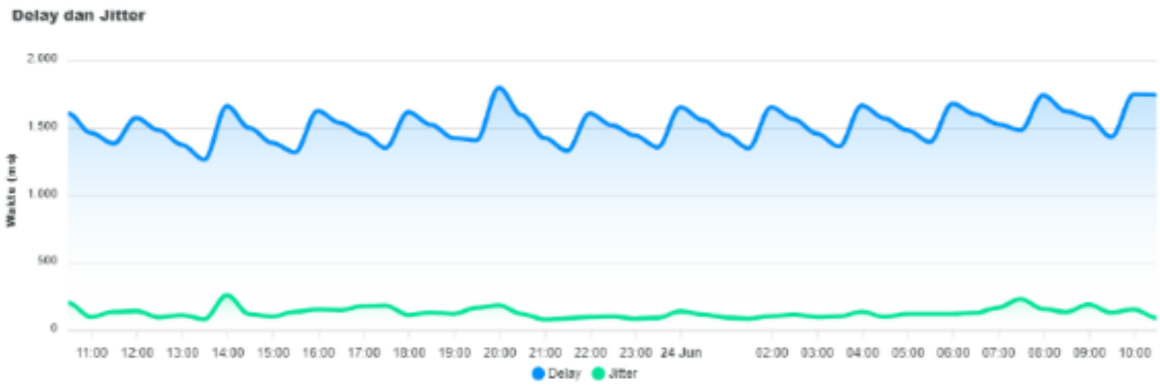

Figure 14. Delay and Jitter Testing Graph on the First Slave Node

Test results of the grapevine monitoring system on the master node over one day demonstrate relatively stable performance. On average, the delay was recorded at 1,724.75 ms, with minor fluctuations primarily attributed to the conditions and performance of the WiFi network in use. Despite slight variations in delay values, the system consistently maintained stability in data transmission. The average jitter was measured at 83.52 ms, indicating minimal variation in packet transmission times, which did not affect the accuracy of the transmitted data. The stability and consistency observed in this testing phase ensured accurate and timely sensor data transmission, a critical factor for real-time monitoring of grapevine conditions. A graph showing test results, including data sent and packet loss on the master node every half hour, can be seen in Figure 13.

In the one-day packet loss test on the master node, out of a total of 17,763 sensor data packets received, there were 13 packet losses, equivalent to about 0.07%. This very low packet loss rate indicates that the monitoring system has extremely high reliability in data transmission from the master node. The graph showing the delay and jitter test results on the first slave node can be seen in Figure 14.

In the delay and jitter tests conducted on the first slave node, the results indicate that the average delay was recorded at 1,513.27 ms, with an average jitter of 129.70 ms. Although the jitter value on the first slave node was higher compared to that recorded on the master node, this fluctuation did not significantly impact the overall system performance. Additionally, the packet loss test results on the first slave node were consistent with those observed on the master node, showing a packet loss of 0.07% out of a total of 17,763 packets transmitted. This extremely low packet loss rate underscores the system's high reliability in data transmission, which is a critical aspect of real-time environmental monitoring. Despite minor differences in performance between the first slave node and the master node, the system as a whole demonstrated strong stability. A graph of delay and jitter testing on the second slave node can be seen in Figure 15.

ISSN 2089-8673 (Print) | ISSN 2548-4265 (Online)
Volume 13, Issue 3, December 2024

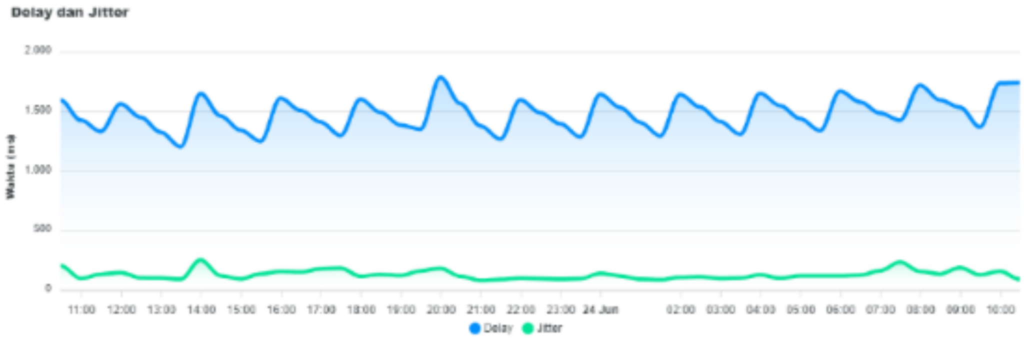

Figure 15. Delay and Jitter Testing Graph on the Second Slave Node

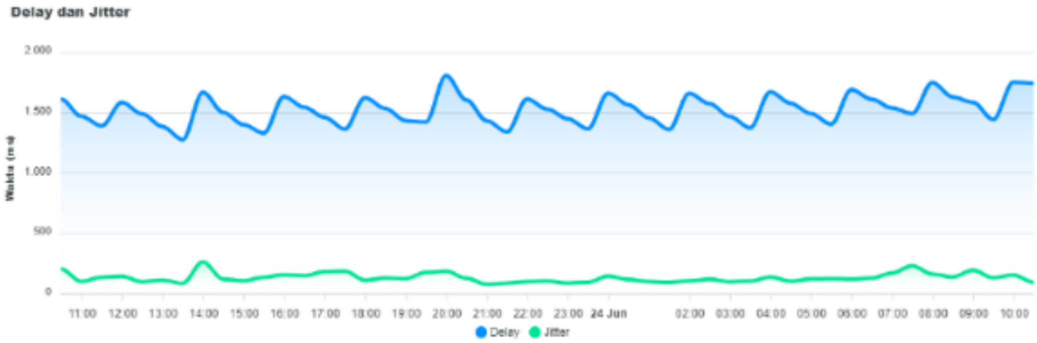

Figure 16. Delay and Jitter Testing Graph on the Third Slave Node

In delay and jitter testing on the second slave node, the graph of average delay every half hour over one day showed a nearly identical pattern. The average delay was recorded at 1,478.72 ms, with an average jitter of 129.64 ms. This variation in average delay was due to the sequential data transmission process from sensor nodes to the master node, starting from the first slave node through to the fourth. For packet loss, the results were the same as the first slave node, at 0.07%, or 13 out of 17,763 received packets, since transmission to the server was handled by a single node—the master node. In delay and jitter testing on the third slave node, the graph of average delay every half hour over one day showed a nearly identical pattern, as shown in Figure 16.

The testing results for delay and jitter on the third slave node indicated that the average delay was 1,521.53 ms, with an average jitter of 130.19 ms. In terms of packet loss, 12 data packets were lost, which corresponds to approximately 0.07% of the total 17,763 data packets transmitted. A difference in packet loss was observed on the third slave node compared to the previous slave nodes. This variation was due to a temporary power outage that caused all nodes to shut down. When power was restored, the setup process on the third slave node was completed more quickly, resulting in one additional data loss compared to the other slave nodes. Although there was a slight difference in packet loss, this did not significantly impact the overall system performance. The graph of transmission count and packet loss testing on the third slave node can be seen in Figure 17.

In testing on the fourth slave node, the delay and jitter graph showed a similar pattern, with an average delay of 1,494.96 ms and an average jitter of 129.75 ms. Packet loss was the same as in the third slave node, with 12 data packets lost, or about 0.07% out of the 17,763 received data packets. The difference in packet loss compared to the master node, first slave node, and second slave node was due to a brief power outage, where the setup process on this slave node, like the third slave node, was quicker.

In the overall node testing, the total average delay was recorded at 1,546.65 ms, with an average jitter of 120.56 ms. The total average packet loss was 0.07%, or 63 data packets out of the 88,815 received. These results demonstrate that the master-slave-based greenhouse monitoring system for grape plants is both stable and reliable. The relatively low average delay and jitter indicate that data can be transmitted quickly and consistently, which is crucial for real-time monitoring.

Figure 17. Graph of Transmission Count and Packet Loss Testing on the Third Slave Node

The very low packet loss rate of 0.07% shows that nearly all data was successfully transmitted and received. These findings confirm the reliability and effectiveness of the monitoring system in collecting and transmitting real-time data on grape plant conditions, even with power interruptions that caused differences in node setup and slight variations in packet loss.

CONCLUSION

This project successfully demonstrates the feasibility and effectiveness of a prototype-based greenhouse grape plant monitoring system, utilizing a master-slave architecture and the ESP-NOW protocol for monitoring environmental conditions. The system reduces dependency on Wi-Fi access points for sensor data transmission. Testing results indicate an average delay of 1,546.65 ms (1.5 seconds), an average jitter of 120.56 ms, and a packet loss rate of 0.07% from a total of 88,815 data transmissions over one day. The variability in packet loss among slave nodes is attributed to electrical disturbances affecting the setup time of each node. The low packet loss reflects the system's reliability in maintaining consistent data transmission, while the measured delay and jitter demonstrate the system's capability to provide acceptable data delivery for real-time monitoring. These findings confirm that the monitoring system is practical, reliable, and capable of providing accurate real-time data in controlled agricultural environments, even with occasional electrical disturbances.

REFERENCES

[1] S. Dwiyatno, E. Krisnaningsih, D. R. Hidayat, and Sulistiyono, “Smart Agriculture Monitoring Penyiraman Tanaman Berbasis Internet of Things,” *Jurnal PROSISKO*, vol. 9, pp. 38–43, 2022.

[2] M. Yusuf and M. Sodik, “Penggunaan Teknologi Internet of Things (IoT) dalam Pengelolaan Fasilitas dan Infrastruktur Lembaga Pendidikan Islam,” *prophetik*, vol. 1, no. 2, pp. 65–82, 2023, doi: 10.26533/prophetik.v1i2.3233.

[3] R. Ramadani, “The Potential of the Internet of Things (IoT) as a Source of Official Statistics for Agriculture,” *semnasoffstat*, no. 1, pp. 161–166, 2023, doi: 10.34123/semnasoffstat.v2023i1.1900.

[4] A. Arifin and M. Rizal, “Implementasi Sistem Otomatisasi Perawatan Tanaman indoor berbasis Internet of Things (IoT),” *remik*, vol. 7, no. 2, pp. 935–945, Apr. 2023, doi: 10.33395/remik.v7i2.12277.

[5] A. Srivastava, D. K. Das, and R. Kumar, “Monitoring of Soil Parameters and Controlling of Soil Moisture through IoT based Smart Agriculture,” in *2020 IEEE Students’ Conference on Engineering and Systems, SCES 2020*, Institute of Electrical and Electronics Engineers Inc., Jul. 2020. doi: 10.1109/SCES50439.2020.9236764.

[6] E. A. Abioye *et al.*, “A review on monitoring and advanced control strategies for precision irrigation,” Jun. 01, 2020, *Elsevier B.V.* doi: 10.1016/j.compag.2020.105441.

[7] R. Burhanudin Baharsah, A. Budimansyah Purba, J. Mulyana, and C. Indra Grahana, “Penerapan Teknologi Internet of Think (IoT) untuk Smart Green House Berbasis Web Server dan Android Controller,” *JIPAKIF NUSANTARA (Jurnal Inovasi Pengembangan Aplikasi dan Keamanan Informasi Nusantara)*, vol. 1, no. 1, pp. 45–54, 2023, [Online]. Available: http://jurnal.edunovationresearch.org/

[8] G. M. Bonde, D. P. M. Ludong, and M. E. I. Najoan, “Smart Agricultural System in Greenhouse based on Internet of Things for Lettuce (Lactuca sativa L.),” *Jurnal Teknik Elektro dan Komputer*, vol. 10, no.

1, pp. 9–16, 2021, doi: 10.35793/jtek.v10i1.31982.

[9] I. Ruslianto, U. Ristian, and H. Hasfani, "Sistem Pintar Untuk Anggur (Sipunggur) pada Kawasan Tropis Berbasis Internet of Things (IoT)," *JEPIN (Jurnal Edukasi dan Penelitian Informatika)*, vol. 8, no. 1, pp. 121–127, 2022.

[10] P. R. Badu, "Greenhouse Technology for Controlled Environment Crop Production," *International Journal for Multidisciplinary Research (IJFMR)*, vol. 5, no. 5, pp. 1–13, 2023, [Online]. Available: www.ijfmr.com

[11] E. Tiara, I. Ruslianto, and Suhardi, "Sistem Pemantauan dan Kendali Kelembapan Tanah dan PH pada Tanaman Anggur Berbasis Android (Studi Kasus: Greenhouse FMIPA Untan)," *Coding : Jurnal Komputer dan Aplikasi*, vol. 11, no. 3, pp. 437–466, 2023.

[12] A. Winata *et al.*, "Optimisasi Kecepatan Internet: Strategi Untuk Meningkatkan Kinerja Jaringan," *Jurnal SITEBA*, vol. 1, pp. 1–18, 2023.

[13] A. Wajiansyah, H. Purwadi, A. Astagani, and S. Supriadi, "Implementation of Master-Slave Method on Multiprocessor-Based Embedded System: Case Study on Mobile Robot," *International Journal of Engineering & Technology*, vol. 7, no. 2, pp. 53–56, 2018, [Online]. Available: www.sciencepubco.com/index.php/IJET

[14] A. Wajiansyah, Supriadi, N. Ramadhan, R. Sandria, and L. M. D. Pratama, "Implementasi Master-slave Pada Embedded System Menggunakan Komunikasi RS-485," *elkha*, vol. 12, no. 1, pp. 26–31, 2020, Accessed: Jul. 25, 2024. [Online]. Available: https://media.neliti.com/media/publications/357650-implementasi-master-slave-pada-embedded-481255bb.pdf

[15] O. H. Abdelkader, H. Bouzebiba, D. Pena, and A. P. Aguiar, "Energy-Efficient IoT-Based Light Control System in Smart Indoor Agriculture," *Sensors*, vol. 23, no. 18, pp. 1–20, Sep. 2023, doi: 10.3390/s23187670.

[16] C.-T. Lee, L.-B. Chen, H.-M. Chu, C.-J. Hsieh, and W.-C. Liang, "An Internet of Things (IoT)-Based Master-Slave Regionalized Intelligent LED-Light-Controlling System," *Applied Sciences (Switzerland)*, vol. 12, no. 1, pp. 1–23, Jan. 2022, doi: 10.3390/app12010420.

[17] S. N. Zulkarnain, R. L. A. Shauri, M. H. Saidin, and A. Z. A. Zamanhuri, "IoT Monitoring of a Master-Slave Robot System using MIT App Inventor," *Journal of Electrical & Electronic Systems Research*, vol. 24, no. Apr2024, pp. 33–39, Apr. 2024, doi: 10.24191/jeesr.v24i1.005.

[18] S. Duobiene *et al.*, "Development of Wireless Sensor Network for Environment Monitoring and Its Implementation Using SSAIL Technology," *Sensors*, vol. 22, no. 14, pp. 1–17, Jul. 2022, doi: 10.3390/s22145343.

[19] V. Masalskyi, D. Čičiurėnas, A. Dzedzickis, U. Prentice, G. Braziulis, and V. Bučinskas, "Synchronization of Separate Sensors' Data Transferred through a Local Wi-Fi Network: A Use Case of Human-Gait Monitoring," *Future Internet*, vol. 16, no. 2, pp. 1–22, Feb. 2024, doi: 10.3390/fi16020036.

[20] R. Rizal Isnanto, Y. Eko Windarto, J. Imago Dei Gloriawan, and F. Noerdiyan Cesara, "Design of a Robot to Control Agricultural Soil Conditions using ESP-NOW Protocol," in *2020 5th International Conference on Informatics and Computing, ICIC 2020*, Institute of Electrical and Electronics Engineers Inc., Nov. 2020. doi: 10.1109/ICIC50835.2020.9288575.

[21] T. N. Hoang, S.-T. Van, and B.D.Nguyen, "ESP-NOW Based Decentralized Low Cost Voice Communication Systems For Buildings," *International Symposium on Electrical and Electronics Engineering (ISEE)*, pp. 108–112, 2019, doi: 10.1109/ISEE2.2019.8921062.