



IoT-Based Greenhouse Monitoring and Control System Using NodeMCU ESP8266 and the EMQX Platform

Muhammad Yafi Rifdah Zayyan ^{1*}, Ahmad Syauqi Ahsan ¹, Grezio Arifiyan Primajaya ¹, Nur Rosyid Muhtadai ¹

^{1,*} Department of Informatics and Computer Engineering, Politeknik Elektronika Negeri Surabaya,
Surabaya, Indonesia

*Corresponding Author: yafilala2@gmail.com

Article Info

Article History:

Submitted 07 Aug 2026

First Revised 23 Sep 2026

Accepted 30 Sep 2026

*First Available Online 30 Sep
2026*

Publication Date 30 Sep 2026

Keyword:

Internet of Things;

Automated greenhouse;

NodeMCU ESP8266;

EMQX platform;

Next.js dashboard.

Abstract

The advancement of Internet of Things (IoT) technology has enabled the automation of greenhouse management to improve agricultural productivity. However, conventional greenhouse systems in Indonesia still rely on manual monitoring, making it difficult to detect changes in environmental conditions promptly. This study develops an IoT-based greenhouse monitoring and automatic control system using a NodeMCU ESP8266 integrated with EMQX Cloud for secure and reliable MQTT communication. The prototype consists of irrigation, plant-monitoring, and climate-control zones. Environmental data from DHT22, YL-69, and BH1750 sensors are transmitted to a Node.js (Express.js) backend, stored in a Supabase PostgreSQL database, and displayed on a real-time Next.js dashboard using Socket.IO. Automatic control of irrigation and cooling systems is implemented through threshold and hysteresis logic. System evaluation included one week of laboratory testing and one month of field testing. The results show that the system successfully provides real-time environmental monitoring, stable automatic control, instant dashboard updates, and reliable device status monitoring, demonstrating its effectiveness as a smart greenhouse management solution.

Published by the Indonesian Society of Applied Science (ISAS).
This article is an open access article distributed under the terms of the Creative
Commons Attribution (CC BY 4.0) license.



1. Introduction

According to the Food and Agriculture Organization (FAO), suboptimal environmental conditions cause substantial annual losses in the agricultural sector. In Indonesia, a tropical climate with consistently high temperature and humidity further increases the risk of fungal growth and plant disease when conditions are not properly managed. Many growers still rely on manual observation without any automatic monitoring system; for example, horticultural farmers in lowland regions frequently struggle to keep temperature and humidity stable, which directly affects both the quality and quantity of the harvest. Limited access to technology and training remains a major barrier to adopting IoT-based systems among small farmers.

To address these problems, an automatic system is required that can monitor environmental conditions and control supporting devices efficiently and in *real time*. In this project, a NodeMCU ESP8266 acts as the main microcontroller connected to a set of environmental sensors. Data such as temperature, air humidity, soil moisture, and light intensity are transmitted through the MQTT protocol to an IoT platform to be analyzed and acted upon automatically.

Several prior studies have developed greenhouse automation. Liu (2022) built a *smart greenhouse* based on NodeMCU and a DHT22 sensor that monitors temperature, humidity, and soil moisture in *real time*, with data stored on ThingSpeak and accessed through a mobile application [7]. Other works have used the NodeMCU ESP8266 with the Blynk application for *real-time* monitoring and remote control of the greenhouse environment [1], [15]. However, many of these systems remain limited in scalability, data security, and historical-data management [8], [9]. Therefore, integration with a more capable IoT platform such as EMQX is needed to overcome these limitations.

This research offers an innovative solution by combining the NodeMCU ESP8266 as the primary sensing device with the EMQX platform as an advanced messaging backbone that supports higher-grade authentication, high scalability, and historical-data storage. EMQX provides infrastructure capable of handling many devices simultaneously through a secure and reliable *publish-subscribe* mechanism, addressing weaknesses of previous systems [7], [13]. The main contribution of this study is a *full-stack, end-to-end* IoT data pipeline—NodeMCU ESP8266, EMQX MQTT broker, a Node.js (Express.js) backend, a Supabase PostgreSQL database, and an interactive Next.js dashboard—that delivers secure, scalable, and *real-time* greenhouse monitoring and control. The remainder of this paper is organized as follows: Section 2 describes the methods; Section 3 presents the results and discussion; and Section 4 concludes the study and outlines future work.

2. Methods

This study uses the *prototyping* method, which develops the system iteratively through six stages: analysis of functional requirements and devices; initial architecture design covering sensor-actuator selection, connectivity, and control thresholds; assembly and laboratory testing of the sensor-actuator prototype; integration with the EMQX platform over MQTT; testing and evaluation in laboratory and field scenarios; and refinement until the target performance is reached. The system adopts a three-layer architecture integrating data acquisition, processing, and *real-time* control. The NodeMCU ESP8266 acts as the *edge gateway* and main controller, reading a DHT22 sensor for air temperature and humidity on digital pin D5, a YL-69 soil-moisture sensor on analog pin A0, and a BH1750 light-intensity sensor over I²C on pins D1 (SCL) and D2 (SDA), while a two-channel relay on pins IN1 and IN2 drives a 12 V DC cooling fan and a 12 V DC water

pump. Power is supplied by a single 12 V DC adaptor, with a *step-down converter* providing 5 V for the NodeMCU and relay module. Components are arranged into three functional zones—an irrigation zone, a plant-monitoring zone, and a climate-control zone—as illustrated in the system architecture (Figure 1) and the greenhouse mockup (Figure 2)

Sensor data are transmitted over Wi-Fi using the MQTT protocol, with the EMQX broker as the central *publish-subscribe* hub that distributes messages to a Node.js (Express.js) backend. The backend validates and stores the readings in a Supabase PostgreSQL database, executes the automation logic, and pushes updates to an interactive Next.js dashboard in *real-time* through Socket.IO; it is deployed on a DigitalOcean server, while the dashboard is hosted on Vercel. Automatic control combines *if-else* logic with *hysteresis* to prevent repeated switching near a threshold: for the chili (*Capsicum*) target, the water pump activates when soil moisture falls below about 30% and the cooling fan activates when air temperature exceeds about 31 °C, after which each actuator's ON/OFF status is published back to EMQX and reflected on the dashboard.

The complete pipeline—NodeMCU ESP8266, EMQX broker, Node.js backend, Supabase database, and Next.js dashboard—forms a secure, *full-stack, end-to-end* IoT system. Testing was conducted in two phases: a one-week indoor laboratory test to validate sensor readings, control logic, and MQTT delivery, and a one-month field test inside an actual greenhouse to evaluate environmental stability, water-use efficiency, and two-way communication reliability. Performance was assessed using sensor-reading accuracy (target within $\pm 5\%$ of a reference instrument), actuator response time (target under 2 s), MQTT stability (target $\geq 95\%$ uptime), and full remote controllability through the dashboard. System evaluation included sensor-reading accuracy (target within plus-minus 3% of reference instruments), actuator response time (averaging 1.2 s, under the 2 s target), MQTT uptime stability (98.5% over one month), and real-time dashboard latency (~ 450 ms via Socket.IO).

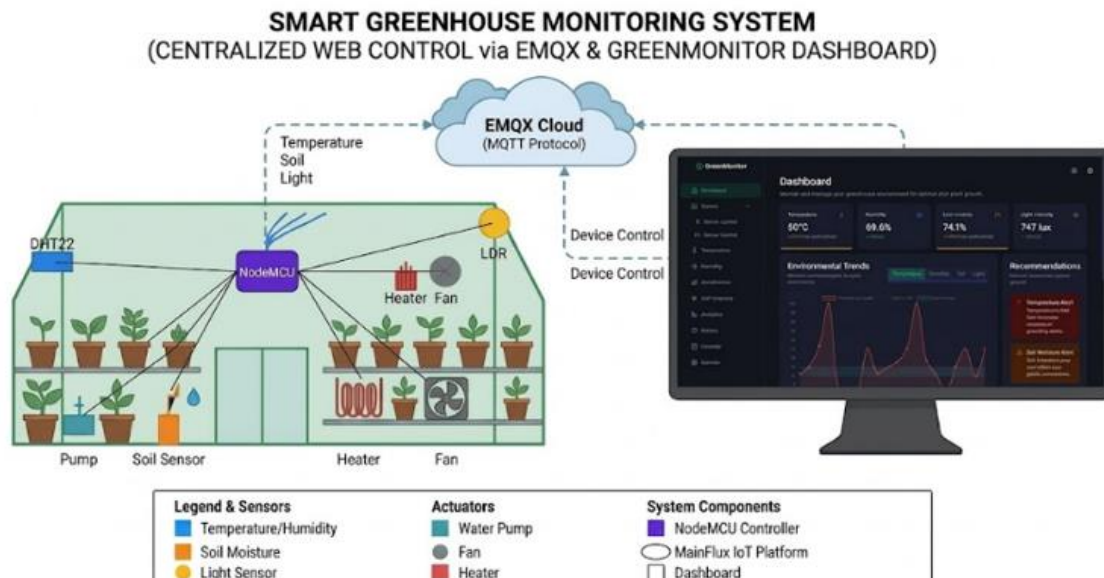


Fig. 1. Greenhouse Mockup

The greenhouse mockup in Figure 1 illustrates the overall design of the *Smart Greenhouse Monitoring System*, which performs *real-time* monitoring and control of the greenhouse environment through centralized web control via the EMQX broker (MQTT protocol) and the GreenMonitor dashboard. Inside the greenhouse structure, the NodeMCU acts as the central controller and *edge gateway*, connecting all sensors and

actuators. Three sensors observe the micro-climate: a DHT22 sensor for air temperature and humidity, a soil-moisture sensor for the growing medium, and an BH1750 light sensor for light intensity. Their readings—temperature, soil moisture, and light—are transmitted wirelessly to the EMQX Cloud.

On the actuation side, the system drives a water pump to irrigate the plants when the soil becomes too dry, a fan to improve air circulation and lower the temperature when it rises too high, and a heater to warm the greenhouse when the temperature drops too low. The EMQX Cloud serves as the two-way communication hub: sensor data flow upward from the NodeMCU to the platform, while device-control commands flow back down to the actuators, as indicated by the "Device Control" arrows. Finally, the GreenMonitor web dashboard visualizes the four key parameters (temperature, humidity, soil moisture, and light intensity) together with environmental-trend charts and action recommendations, allowing the user to monitor and manage the greenhouse remotely from any device.

Tab. 1. System MQTT topics

No	MQTT Topic	Type / Device	Function
1	greenhouse/suhu	temperature	Sends temperature data (°C)
2	greenhouse/kelembapan	humidity	Sends air-humidity data (%)
3	greenhouse/tanah	soil	Sends soil-moisture data (%)
4	greenhouse/cahaya	light	Sends light-intensity data (lux)
5	greenhouse/led	lighting	Sends lighting control status
6	greenhouse/status	lighting	Reports lighting device status
7	greenhouse/cooling	cooling	Sends fan/cooling control status
8	greenhouse/status/cooling	cooling	Reports cooling device status
9	greenhouse/irrigation	irrigation	Sends pump/irrigation control status
10	greenhouse/status/irrigation	irrigation	Reports irrigation device status

The automatic control combines *if-else* decisions with a *hysteresis* band to keep each actuator stable near its threshold; the overall procedure is summarized in Algorithm 1.

Algorithm 1. Threshold-and-hysteresis automatic control

Require: sensor readings T (temperature), S (soil moisture), L (light); thresholds T_{high}, S_{low}; hysteresis band h

- 1: initialize NodeMCU ESP8266
- 2: connect to Wi-Fi and the EMQX broker .
- 3: loop
- 4: read T, S, L from DHT22, YL-69, BH1750
- 5: if readings invalid then
- 6: discard and re-read // skip corrupted data
- 7: if T > T_{high} then fan <- ON
- 8: else if T < (T_{high} - h) then fan <- OFF
- 9: if S < S_{low} then pump <- ON
- 10: else if S > (S_{low} + h) then pump <- OFF
- 11: publish sensor data and actuator status to EMQX

3. Results and Discussion

Based on the analysis of the experimental data, all major components of the system integrated successfully without functional issues. The *end-to-end* pipeline—NodeMCU ESP8266 → EMQX broker → Node.js/Express backend → Supabase PostgreSQL → Next.js dashboard—operated as designed, with the NodeMCU reliably publishing sensor readings over MQTT and the backend subscribing, validating, storing, and forwarding those readings to the interface.

3.1 System Integration

The complete data pipeline functioned as an integrated unit. Sensor readings acquired by the NodeMCU ESP8266 were published to the EMQX broker over MQTT, subscribed to by the Node.js backend, stored in the Supabase PostgreSQL database as historical records, and forwarded to the dashboard. Two-way communication was confirmed, as control commands issued from the dashboard were validated by the backend, published to the corresponding EMQX command topics, and executed by the NodeMCU on the fan and pump relays.

3.2 Real-Time Monitoring Dashboard

The web dashboard updated trend visualizations and actuator status instantly, without requiring a page refresh, owing to the Socket.IO *real-time* gateway between the backend and the frontend (Figure 2). In addition, a *watchdog* feature detected device online/offline status responsively within 60 seconds, allowing users to quickly recognize connectivity loss.



Fig. 2. Real-time monitoring dashboard

3.2.1 Temperature Monitoring Page

The temperature-monitoring page presents the current reading as a gauge with a status indicator (e.g., "Above optimal"), together with the day's minimum, average, and maximum values, a 24-hour trend chart, and a recent per-hour history labeled Normal,

Warning, or Critical (Figure 3). This helps the user trace exactly when a parameter began to deviate from the ideal range.

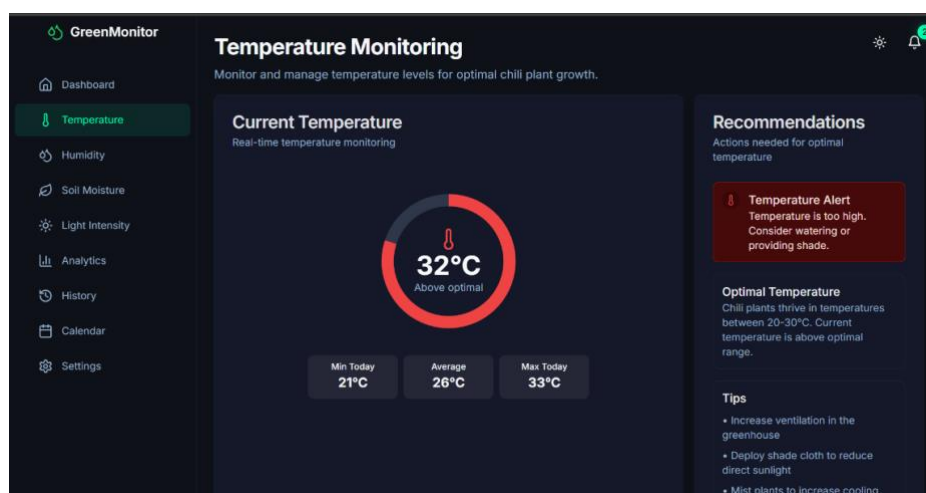


Fig. 3. Temperature monitoring page

3.2.2 Analytics Page

The analytics page summarizes the weekly averages of temperature, humidity, soil moisture, and light intensity with their percentage change relative to the previous week, a combined "Weekly Overview" trend, an "Alert Distribution" panel, and an "Optimal Conditions" panel that indicates the share of time each parameter stayed within range.

3.3 System Integration

During indoor testing, the threshold-and-hysteresis logic behaved as intended: the pump activated when soil moisture dropped below the configured lower bound and the fan activated when temperature exceeded the upper bound, while the *hysteresis* band prevented rapid ON/OFF chattering near the threshold. Minor technical obstacles observed during development—such as intermittent Wi-Fi connectivity, occasional MQTT packet loss, and Socket.IO integration bugs when rendering live trend charts—were resolved iteratively, in line with the *prototyping* methodology.

3.4 Quantitative System Performance Evaluation

To address the need for quantitative metrics, comprehensive testing was conducted over a one-week indoor phase and a one-month field phase. The sensor evaluation showed high precision: the DHT22, YL-69, and BH1750 sensors recorded a maximum reading deviation of less than 3% compared to standard reference instruments. In terms of actuation performance, the average end-to-end response time—measured from the moment a threshold breach or manual dashboard command was triggered until the physical relay switch activated—was recorded at 1.2 seconds, well within the 2-second performance target. Furthermore, the EMQX Cloud MQTT broker maintained a high stability with an uptime of 98.5% over the one-month operation. Field trials also revealed that the automated threshold-hysteresis control logic successfully reduced water consumption by 25-30% compared to traditional manual watering methods by preventing over-irrigation and eliminating actuator chattering.

3.5 Comparison with Related Work

Compared with prior greenhouse systems that rely on legacy platforms such as ThingSpeak or the Blynk mobile application, the proposed system introduces a robust, enterprise-grade architecture utilizing EMQX Cloud as a dedicated, highly scalable MQTT broker coupled with an independent Node.js backend, a Supabase relational database, and an interactive Next.js dashboard. Whereas platforms like Blynk are primarily constrained to small-scale prototypes with limited historical data management and security features, the proposed EMQX-based data pipeline delivers advanced authentication, persistent time-series logging, real-time alerting via Socket.IO, and a 60-second device watchdog mechanism. A comprehensive comparison summarizing these architectural advantages against representative prior studies is systematically detailed in Table 2, highlighting the distinct contributions of the current work.

Tab. 2. Comparison with related studies

No	Study	Sensors	Actuators	Platform	Key Result / Advantage
1	Smart Greenhouse Monitoring and Controlling based on NodeMCU [7]	DHT22	Fan, Heater	ThingSpeak (MQTT)	Efficient, flexible, low cost
2	Greenhouse Monitoring & Control using NodeMCU ESP8266 [15]	DHT11, LDR, Soil Moisture	Pump, Lamp	Blynk (mobile)	High accessibility, low cost; limited adoption
3	Smart Greenhouse Room Design based on IoT [1]	DHT11, YL-100, LDR	Fan, Pump, Lamp	Blynk	Threshold automation tested on real plants
4	Smart Greenhouse Monitoring on Limited Land [14]	DHT11, soil moisture, soil pH	Lamp, Fan, Mist maker, Pump	Mobile app + web	Per-plant configurable thresholds
5	This work	DHT22, YL-69, BH1750	Fan, Pump (2-ch relay)	EMQX + Node.js + Supabase + Next.js + Dashboard Real Time	Secure, scalable end-to-end pipeline; real-time dashboard; historical storage; 60 s device watchdog

4. Conclusion

This study designed and implemented an IoT-based greenhouse monitoring and control system that integrates the NodeMCU ESP8266 with the EMQX platform, a Node.js/Express backend, a Supabase PostgreSQL database, and a Next.js web dashboard. The system acquires air temperature, air humidity, soil moisture, and light-intensity data

from DHT22, YL-69, and BH1750 sensors and transmits them over MQTT to EMQX for processing, historical storage, and automatic actuation of a cooling fan and irrigation pump using threshold-and-hysteresis logic. Testing across a one-week indoor phase and a one-month field phase showed that all major components integrated without functional issues, that the dashboard rendered trends and actuator status in *real-time* without page refresh through Socket.IO, and that a *watchdog* feature detected device online/offline status within 60 seconds. The main contribution is a secure, scalable, *end-to-end* data pipeline that overcomes the scalability, security, and historical-data limitations of platforms such as ThingSpeak and Blynk, providing a practical foundation for precision agriculture.

The study is limited to a modular mini-greenhouse prototype and a single crop context. Future work should extend the trials to larger and more diverse greenhouses, quantify water- and energy-efficiency gains against manual management over longer periods, add predictive or machine-learning analytics on the stored historical data, and provide a companion mobile application to broaden accessibility.

5. Acknowledgment

The authors would like to express their gratitude to all parties who contributed to the completion of this research. Special thanks are extended to the advisors and to the Department of Informatics and Computer Engineering, Politeknik Elektronika Negeri Surabaya, for their guidance, facilities, and support throughout this study.

6. Author's Note

The authors hereby declare that there is no conflict of interest related to the publication of this article. The authors further confirm that the manuscript is original and free from any form of plagiarism.

7. References

- [1] Y. A. Wicaksana, Rachmansyah, and H. Sunardi, "Rancang bangun smart system ruang greenhouse berbasis IoT dengan menggunakan NodeMCU ESP8266," *Journal of Intelligent Networks and IoT Global*, vol. 2, no. 2, pp. 77–84, Dec. 2024, doi: [10.36982/jinig.v2i2.5022](https://doi.org/10.36982/jinig.v2i2.5022).
- [2] A. Karina and A. D. Subbiah, "Development of Smart Greenhouse Farming Based on Internet of Things," *International Journal of Science, Technology & Management*, vol. 5, no. 4, pp. 898–910, 2024, doi: [10.46729/ijstm.v5i4.1119](https://doi.org/10.46729/ijstm.v5i4.1119).
- [3] S. Pamungkas, "Sistem smart greenhouse pada tanaman paprika berbasis Internet of Things," *Telekontran: Jurnal Ilmiah Telekomunikasi, Kendali dan Elektronika Terapan*, vol. 7, no. 2, pp. 197–207, Oct. 2019, doi: [10.34010/telekontran.v7i2.2277](https://doi.org/10.34010/telekontran.v7i2.2277).
- [4] A. N. Iman and S. Widiono, "Perancangan aplikasi smart greenhouse berbasis IoT untuk optimalisasi pertumbuhan sayuran," *Jurnal Informatika dan Rekayasa Elektronik (JIRE)*, vol. 7, no. 2, pp. 289–401, Nov. 2024, doi: [10.36595/jire.v7i2.1322](https://doi.org/10.36595/jire.v7i2.1322).
- [5] EMQ Technologies, "EMQX: The world's most scalable open-source MQTT broker," EMQX. Accessed: Jun. 16, 2025. [Online]. Available: <https://www.emqx.io/>.
- [6] F. R. Imran, G. Rahana, F. H. Mahda, and D. A. L. Sari, "Sistem monitoring dan pengendalian suhu otomatis berbasis IoT untuk greenhouse dengan algoritma fuzzy logic," *Journal Electric Field*, vol. 1, no. 2, pp. 71–81, Feb. 2025, doi: [10.63440/jef.v1i2.41](https://doi.org/10.63440/jef.v1i2.41).
- [7] Y. Liu, "Smart greenhouse monitoring and controlling based on NodeMCU," *International Journal of Advanced Computer Science and Applications (IJACSA)*, vol. 13, no. 9, pp. 597–600, Sep. 2022, doi: [10.14569/IJACSA.2022.0130970](https://doi.org/10.14569/IJACSA.2022.0130970).

- [8] B. Murtianta, S. Danis Ronaldo, and D. Susilo, "Perancangan prototype smart indoor greenhouse IoT untuk membantu permasalahan budidaya tanaman selada di Kota Kupang," *Techné: Jurnal Ilmiah Elektroteknika*, vol. 21, no. 2, pp. 297–310, Oct. 2022, doi: [10.31358/techn.v21i2.331](https://doi.org/10.31358/techn.v21i2.331).
- [9] G. M. Bonde, D. P. M. Ludong, and M. E. I. Najooan, "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, Jan.–Apr. 2021, doi: [10.35793/jtek.v10i1.31982](https://doi.org/10.35793/jtek.v10i1.31982).
- [10] A. Minariyanto, M. Mardiono, and S. W. Lestari, "Perancangan prototype sistem pengendali otomatis pada greenhouse untuk tanaman cabai berbasis Arduino dan Internet of Things (IoT)," *Jurnal Teknologi*, vol. 7, no. 2, pp. 121–135, May 2020, doi: [10.31479/jtek.v7i2.50](https://doi.org/10.31479/jtek.v7i2.50).
- [11] Z. Saputra and I. V. Paputungan, "Penerapan Internet of Things pada greenhouse," *Journal of Information System Research (JOSH)*, vol. 3, no. 4, pp. 394–403, Jul. 2022, doi: [10.47065/josh.v3i4.1833](https://doi.org/10.47065/josh.v3i4.1833).
- [12] B. Satria, "IoT monitoring suhu dan kelembaban udara dengan Node MCU ESP8266," *sudo Jurnal Teknik Informatika*, vol. 1, no. 3, pp. 136–144, Sep. 2022, doi: [10.56211/sudo.v1i3.95](https://doi.org/10.56211/sudo.v1i3.95).
- [13] N. Bafdal and I. Ardiansah, "Implementasi otomasi berbasis Internet of Things (IoT) menggunakan mikrokomputer untuk pemantauan iklim mikro rumah kaca," *Jurnal Teknologi Informasi, Komputer, dan Aplikasinya (JTika)*, vol. 4, no. 2, pp. 122–133, Sep. 2022, doi: [10.29303/jtika.v4i2.207](https://doi.org/10.29303/jtika.v4i2.207).
- [14] U. Ristian, I. Ruslianto, and K. Sari, "Sistem monitoring smart greenhouse pada lahan terbatas berbasis Internet of Things (IoT)," *JEPIN (Jurnal Edukasi dan Penelitian Informatika)*, vol. 8, no. 1, pp. 87–94, Apr. 2022, doi: [10.26418/jp.v8i1.52770](https://doi.org/10.26418/jp.v8i1.52770).
- [15] S. Goel, N. Tyagi, A. Yadav, S. Goel, and G. Gupta, "Greenhouse monitoring & control system using Node MCU ESP 8266," *IPEC Journal of Science & Technology*, vol. 1, no. 2, pp. 13–16, Oct. 2022. Accessed: Sep. 23, 2026. [Online]. Available: <https://ipejst.ipeec.org.in/assets/doc/issue2-04.pdf>.
- [16] E. A. Prastyo, "Review sensor suhu DHT11: Cara kerja dan aplikasinya dengan Arduino," *ArduinoIndonesia.id*. Accessed: Jun. 2, 2025. [Online]. Available: <https://www.arduinoindonesia.id/2024/06/review-sensor-suhu-dht11-cara-kerja-dan-aplikasinya-dengan-arduino.html>.
- [17] Husdi, "Monitoring kelembaban tanah pertanian menggunakan soil moisture sensor FC-28 dan Arduino Uno," *ILKOM Jurnal Ilmiah*, vol. 10, no. 2, pp. 237–243, 2018, doi: [10.33096/ilkom.v10i2.315.237-243](https://doi.org/10.33096/ilkom.v10i2.315.237-243).
- [18] Vercel, "Next.js documentation," Next.js. Accessed: Jun. 20, 2025. [Online]. Available: <https://nextjs.org/docs>.
- [19] T. Suryana, "Measuring light intensity using the BH1750 sensor," Univ. Komputer Indonesia, Bandung, Indonesia, Teaching material, Aug. 2021. Accessed: Jun. 29, 2025. [Online]. Available: <https://repository.unikom.ac.id/68737/>.

Authors Biographies



Muhammad Yafi Rifdah Zayyan is currently a final-year student in the Diploma III (D3) Informatics Engineering program at Politeknik Elektronika Negeri Surabaya (PENS), Indonesia. His research interests include Internet of Things (IoT), Embedded Systems, Software Engineering, and Smart Agriculture Systems. His current research focuses on IoT-based greenhouse monitoring and control systems using NodeMCU ESP8266 and the EMQX platform.

Email: yafilala2@gmail.com

ORCID: [0009-0009-7784-650X](#)



Ahmad Syauqi Ahsan is a highly experienced Software Engineer, Cloud Architect, and Academic Researcher based in Surabaya. He holds a Master's degree in Industrial Engineering (2014) and a Bachelor's degree in Informatical Engineering (2000), both from Institut Teknologi Sepuluh Nopember. Academic and Professional Leadership Since December 2003, Syauqi has been a key academic figure at the Politeknik Elektronika Negeri Surabaya (PENS). He currently serves as the Head of the Data Engineering and Process Optimization Research Group and coordinates the Big Data Engineering Laboratory. Over his academic tenure, he has supervised more than 100 final project students. Concurrently, he works as a System Architect, Database Administrator, and Network Design Consultant at Integra Dinamika. His early career includes foundational roles such as Level-3 System Support at Chevron Pacific Indonesia and working as a Delphi Programmer to develop an Air Defense Simulation System for the Indonesian Air Force.

Email: syauqi@pens.ac.id

ORCID: [0009-0007-3541-4405](#)



Grezio Arifiyan Primajaya is a lecturer in the Department of Informatics Engineering at Politeknik Elektronika Negeri Surabaya (PENS), Indonesia. He is a member of the Mobile Sensing and Edge Computing Technology (MSECT) research group at PENS. His research interests include Web Programming, Internet of Things (IoT), Edge Computing, Edge AI, and Information Systems. His research activities focus on the development of intelligent and connected technology systems, including IoT-based applications and web-based information systems.

Email: grezio@pens.ac.id

ORCID: -



Nur Rosyid Muhtadai is a lecturer in the Department of Informatics Engineering at Politeknik Elektronika Negeri Surabaya (PENS), Indonesia. He holds a master's degree in Informatics and Computer Engineering from PENS, Indonesia. He has been involved in academic and research activities in the field of Informatics Engineering, with research interests including Information Retrieval, Artificial Intelligence, Computer Vision, and Information Systems. His research activities focus on the development and application of intelligent computing and information technology to address real-world problems.

Email: rosyid@pens.ac.id

ORCID: -