ISSN 2355-3286

ULTIMA Computing

Jurnal Sistem Komputer

Engineering | Vol. 17 | No. 1 | Page 1 - 53 | June 2025 | ISSN 2355-3286

UMN

UNIVERSITAS
MULTIMEDIA
NUSANTARA

ISSN 2355-3286

EDITORIAL BOARD

EDITORIAL ADDRESS

Universitas Multimedia Nusantara (UMN)
Jl. Scientia Boulevard
Gading Serpong
Tangerang, Banten - 15811
Indonesia
Phone. (021) 5422 0808
Fax. (021) 5422 0800
Email : ultimacomputing@umn.ac.id

Ultima Computing : Jurnal Sistem Komputer is a Journal of Computer Engineering Study Program, Universitas Multimedia Nusantara which presents scientific research articles in the field of Computer Engineering and Electrical Engineering as well as current theoretical and practical issues, including Edge Computing, Internet-of-Things, Embedded Systems, Robotics, Control System, Network and Communication, System Integration, as well as other topics in the field of Computer Engineering and Electrical Engineering. The Ultima Computing : Jurnal Sistem Komputer is published regularly twice a year (June and December) and is jointly managed by the Computer Engineering and Electrical Engineering Study Program at Universitas Multimedia Nusantara.

Call for Papers

International Journal of New Media Technology (IJNMT) is a scholarly open access, peer-reviewed, and interdisciplinary journal focusing on theories, methods and implementations of new media technology. Topics include, but not limited to digital technology for creative industry, infrastructure technology, computing communication and networking, signal and image processing, intelligent system, control and embedded system, mobile and web based system, and robotics. IJNMT is published twice a year by Faculty of Engineering and Informatics of Universitas Multimedia Nusantara in cooperation with UMN Press.

Ultimatics : Jurnal Teknik Informatika is the Journal of the Informatics Study Program at Universitas Multimedia Nusantara which presents scientific research articles in the fields of Analysis and Design of Algorithm, Software Engineering, System and Network security, as well as the latest theoretical and practical issues, including Ubiquitous and Mobile Computing, Artificial Intelligence and Machine Learning, Algorithm Theory, World Wide Web, Cryptography, as well as other topics in the field of Informatics.

Ultima Computing : Jurnal Sistem Komputer is a Journal of Computer Engineering Study Program, Universitas Multimedia Nusantara which presents scientific research articles in the field of Computer Engineering and Electrical Engineering as well as current theoretical and practical issues, including Edge Computing, Internet-of-Things, Embedded Systems, Robotics, Control System, Network and Communication, System Integration, as well as other topics in the field of Computer Engineering and Electrical Engineering.

Ultima InfoSys : Jurnal Ilmu Sistem Informasi is a Journal of Information Systems Study Program at Universitas Multimedia Nusantara which presents scientific research articles in the field of Information Systems, as well as the latest theoretical and practical issues, including database systems, management information systems, system analysis and development, system project management information, programming, mobile information system, and other topics related to Information Systems.

ISSN 2355-3286

FOREWORD

ULTIMA Greetings!

Ultima Computing : Jurnal Sistem Komputer is a Journal of Computer Engineering and Electrical Engineering at Multimedia Nusantara University which presents scientific research articles in the field of Computer Systems as well as the latest theoretical and practical issues, including Edge Computing, Internet-of-Things, Embedded Systems, Robotics, Control Systems, Network and Communication, System Integration, and other topics in the field of Computer Engineering and Electrical Engineering.

In this June 2025 edition, Ultima Computing enters the 1st Edition of Volume 17. In this edition there are six scientific papers from researchers, academics and practitioners in the fields of Computer Engineering and Electrical Engineering. Some of the topics raised in this journal are: Real-Time Space Availability Detection in Smart Parking Systems using Infrared Sensor and Microcontroller ATmega 328p, Long Term Prediction of Extreme Weather with Long Short-Term Memory (LSTM) Model: Effect of Climate Change, Design of Greenhouse Prototype Controller and Monitor on Green Mustard Plants IoT, Design of a Nutrient and Environment Monitoring IoT Device in Vertical Hydroponic System, An Adaptive Stacking Approach for Monthly Rainfall Prediction with Hybrid Feature Selection, and Micro-Scale CPV Performance Enhancement through V-Trough Concentration and Passive Cooling.

On this occasion we would also like to invite the participation of our dear readers, researchers, academics, and practitioners, in the field of Engineering and Informatics, to submit quality scientific papers to: International Journal of New Media Technology (IJNMT), Ultimatics : Jurnal Teknik Informatics, Ultima Infosys: Journal of Information Systems and Ultima Computing: Journal of Computer Systems. Information regarding writing guidelines and templates, as well as other related information can be obtained through the email address ultimacomputing@umn.ac.id and the web page of our Journal here.

Finally, we would like to thank all contributors to this June 2025 Edition of Ultima Computing. We hope that scientific articles from research in this journal can be useful and contribute to the development of research and science in Indonesia.

June 2025,

Monica Pratiwi, S.ST., M.T.
Editor-in-Chief

ISSN 2355-3286

TABLE OF CONTENT

ISSN 2355-3286

Real-Time Space Availability Detection in Smart Parking Systems using Infrared Sensor and Microcontroller ATmega 328p

Fidel Lusiana Putri[1]
[1]Electrical Engineering, Faculty of Engineering, Universitas Negeri Semarang, Semarang, Indonesia
[1]fidellusiana@students.unnes.ac.id

Accepted January 22, 2025
Approved April 11, 2025

***Abstract*—In the growing digital era, vehicles have become a staple for daily mobility. This poses a significant parking problem, especially in urban areas with limited parking spaces. The search for parking spaces often leads to congestion, air pollution, fuel waste, and frustration for drivers. This research aims to develop an smart parking system using infrared sensors and an ATmega328p microcontroller to detect the availability of parking slots in real time. This system provides information to the driver through a Liquid Crystal Display (LCD) screen installed at the entrance of the parking area. The methods used in this research are direct testing and tool prototyping. The test results show that the parking system model successfully displays the availability status of each parking area in real-time on the display screen, with data obtained from infrared sensors and processed by the ATmega328p microcontroller.**

***Index Terms*—Automatic Parking Slot; Internet of Things (IoT); Infrared sensor; Traffic Congestion**

I. INTRODUCTION

In the current development of the digital era, vehicles have become a basic need for mobility in daily life [1-5]. This leads to crucial parking problems, affecting people's quality of life. With a limited number of parking spaces in almost every city, finding a parking space is often time-consuming and causes traffic congestion, air pollution, fuel waste, and frustration for drivers [5-11]. In addition, urban expansion leads to traffic congestion, where the search for parking spaces accounts for up to 30%-50% of congestion within cities [4][7][12-14]. Therefore, the availability of parking spaces is a major issue in urban areas, and awareness of its importance is increasing. So smart parking management systems that collect real-time information about parking locations that are accessible to the public are becoming a major focus [4][5][15-20].

According to Statista [21], it is estimated that the number of autonomous vehicles worldwide will increase to more than 54 million by 2024, with a global market value of nearly 62 billion US dollars by 2026. This indicates a growing need for efficient and innovative parking solutions. Smart parking systems can help overcome various problems arising from conventional parking by utilizing technologies such as the Internet of Things (IoT), sensors, and mobile applications [1-7][12][14-17][22][23].

Some research on automotive parking systems has been carried out, such as research conducted by S. Saravanan et al. [17] designed an IoT-based smart parking system. This system uses infrared sensors to detect available parking spaces and display the information on a mobile application, where users can reserve parking spaces and reduce traffic congestion. The system also includes an experimental setup with Infrared sensors, Arduino Uno, and Wireless Fidelity (Wi-Fi) modules.

Research by Md. Thoufiq Zumma et al. [24] designed an Arduino Uno-based smart car parking system. The system uses MobileNet Classifier to recognize available parking slots based on images taken from cameras that continuously monitor parking lot conditions. In addition, the system also involves components such as servo motors, IR sensors, fire alarms, and Liquid Crystal Display (LCD) that are all connected to Arduino.

Research by Soni et al. [25] also designed a smart parking system that uses ATmega 328p microcontroller technology. The parking system uses infrared sensors to detect the availability of parking slots and then display them on the LCD screen. Buzzers and Light Emitting Diode (LED) are also used as warnings if there are vehicles in and out.

From various literature conducted, the application of technology in parking management systems can be an effective solution to overcome this problem. One approach that can be taken is to develop a smart parking system that can detect the availability of parking spaces in real time and provide that information to the driver [4][5][8][15-20][22][24][25]. This system not only helps drivers find parking spaces quickly but also reduces traffic congestion caused by prolonged searches for parking spaces [5][6][12][20].

In this study [15] introduced a smart parking system that uses Arduino and infrared sensors to detect the presence of vehicles and display parking availability in real-time. Aimed at reducing traffic congestion and fuel consumption, the system uses LEDs to indicate slot status.

In [17], an Arduino Uno-based smart parking system was developed using IR sensors and a Telnet-based mobile app to monitor parking availability. However, the system lacks user-friendly features, object differentiation, and real-world validation. This highlights the need for a more scalable and intelligent parking system with improved accuracy and usability.

In [18], an ultrasonic-based parking assistance system uses Arduino Uno to improve safety and convenience by providing real-time visual and auditory feedback. While the system is low-cost and easy to implement, it has limitations such as inaccurate distance measurement and difficulty in detecting certain objects.

This research will be carried out with a direct test method and prototyping tools by utilizing infrared sensors to detect the availability of parking slots. Real-time information regarding the availability of parking slots will be displayed on the information screen installed at the entrance of the parking area. This allows drivers to know the status of parking availability before entering the parking area, so they can avoid full areas and find alternatives more efficiently.

The implementation of this automated parking system is expected to increase convenience and efficiency in parking management in urban areas, reduce the negative impact of searching for parking spaces, and support the creation of a more environmentally friendly and organized urban environment.

II. METHODOLOGY

The method used in this research is hands-on testing and prototyping which involves several stages, including overall system design through block diagrams, system flowcharts, and hardware and software system design. The initial stage involved designing a block diagram to provide an overview of the main components and their relationships, followed by designing a system flowchart that describes the operational steps. Next, the hardware and software design were carried out to ensure that the physical components and applications that support the operation of the system are designed as required.

A. System Block Diagram

The initial stage in this research is a system design that identifies the system components using a block diagram. The system is designed using electronic components shown in Figure 1 as a block diagram. These electronic components include an infrared sensor, Atmega328 microcontroller, servo motor, LCD and Power Supply.

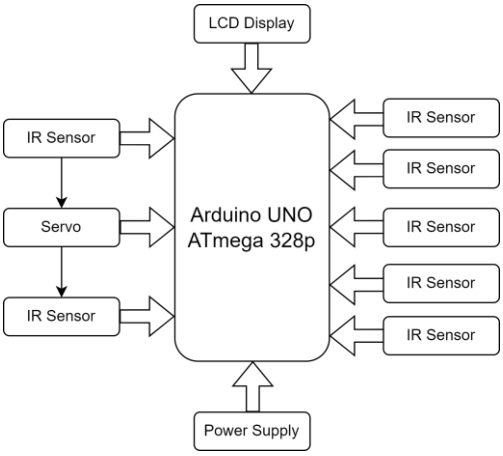


Fig. 1. System Block Diagram

The system uses infrared sensors to detect vehicles. The sensors are installed in each parking slot, entrance gate, and exit gate to monitor vehicle occupancy and vehicle entry and exit. Where the gate is controlled using a servo motor.

The main component of this parking system is the Arduino ATmega 328p microcontroller, which interprets input data from Infrared sensors based on instructions programmed and stored in its memory.

Here are the functions of the components in the block diagram [26]:

1. Power Supply: This circuit begins with a 5-volt, 2-ampere power source from an Arduino Uno. The 5-volt power supply has various components in the system.
2. Arduino UNO: The Arduino UNO board serves as the main controller for the system. It processes sensor data, controls servo motors, and displays information on the LCD.
3. Module I2C: This module interfaces with the 20×4 LCD (Liquid Crystal Display), simplifying the connection between the LCD and the Arduino.
4. 20×4 LCD: This LCD displays information about parking slots, including messages about slot availability and other system statuses.
5. Infrared Sensor: The system utilizes two infrared sensor, one positioned at the entrance and the other at the exit gate.
6. Mini Servo Motor: Mini servo motors control the opening and closing of gates, allowing or preventing vehicle entry based on data from the IR sensors.
7. Six Parking Slots: The system features six parking spaces, each equipped with an Infrared sensor to detect the presence of a car. These sensors are crucial for monitoring which spaces are occupied and which are available.

Overall, the Arduino microcontroller processes sensor data, controls gate access, and displays parking

slot information, making the parking system automated and efficient.

B. *Software Design*

This research utilizes the Arduino IDE software to write program code for the automatic parking system and slot availability management that has been designed to provide instructions to the system. For the Arduino Uno device to function properly, this program code must be entered into the Arduino Uno. To run the program in the Arduino Uno microcontroller, a USB drive, Arduino IDE software, and an ATmega328p Microcontroller are required. The following program code in C language serves to connect the various modules according to the condition of the components when used.

C. *System Flow Chart Design*

An illustration of the overall system flow chart is shown in Figure 2 below.

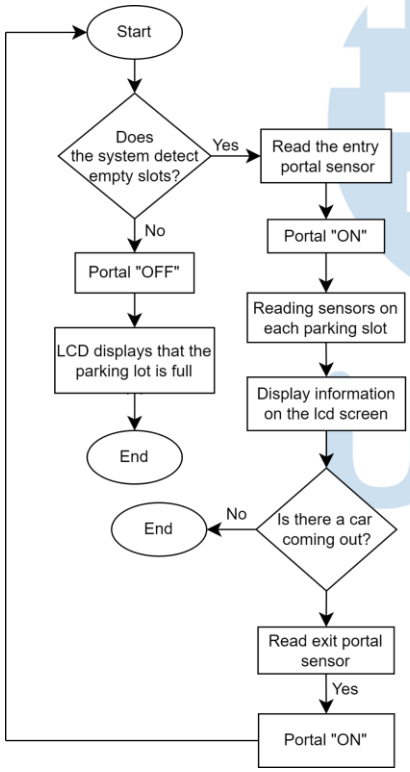


Fig. 2. System Flow Chart

In the flowchart system above, the Arduino will receive input from the infrared sensor at the entrance, and then the Arduino will process the input to drive the servo motor so that the door portal opens. However, if the parking slot is detected to be full, the servo motor will not work, so the entrance portal will not open. The Arduino will also receive input from the infrared sensors installed in each parking slot, and then the Arduino will process the input to move the servo motor so that the entrance portal opens. The system will continue to run if the Arduino is connected to the power supply.

D. *Hardware System Design*

The circuit scheme of the system is shown in Figure 3.

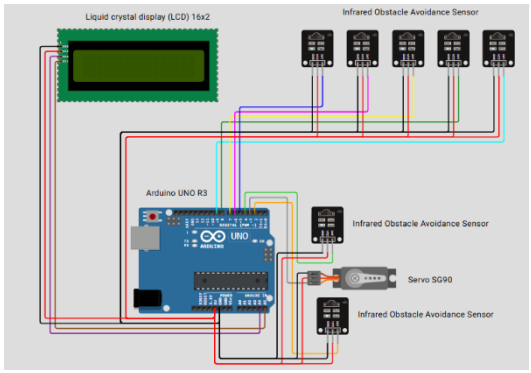


Fig. 3. Circuit Diagram

In Figure 3, the circuit configuration of the automatic parking system and parking slot availability is illustrated. This system utilizes an ATmega 328p microcontroller as the central controller. Arduino handles sensor data processing, servo motor control, and information display on the LCD. The microcontroller is connected to infrared sensors at the entrance and exit of the parking lot as well as at the five parking slots through digital pins 2-9 Arduino to detect the presence of cars. Besides being connected to the infrared sensor, the microcontroller is also connected to the I2C module and servo. The Arduino pin is connected to the I2C module on pins A4 and A5, then this I2C module is connected to the LCD to display parking slot availability information. The servo is connected to digital pin 3 of the Arduino to regulate the opening and closing of the parking bar. Based on the data collected from the Infrared sensor, the servo motor will be activated.

This innovative automated parking system is designed to improve efficiency and convenience for users. The system utilizes various reliable components in Table 1 below.

TABLE I. COMPONENTS OF PROTOTYPING

No.	Components	Amount
1	Arduino UNO	1
2	IR Sensor	7
3	Servo Motor	1
4	Jumper Cables	± 30
5	LCD	1

III. RESULTS AND ANALYSIS

Based on a series of methods, the results of the tool design are shown in Figure 4.

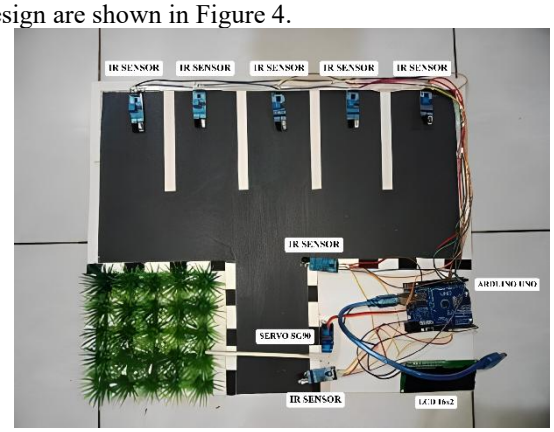


Fig. 4. Prototype Design

The design results of the parking system and the automatic parking slot availability using Arduino are presented through prototypes and simulations, as shown in Figure 4. The initial step of the simulation of the prototype of this tool starts when a car approaches the parking entrance, then two infrared sensors installed on both sides of the gate will detect it. The task of this infrared sensor is to detect objects at a predetermined distance.

This research utilizes the Arduino IDE software to write program code for the automatic parking system and slot availability management that has been designed to provide instructions to the system. In order for the Arduino Uno device to function properly, this program code must be entered into the Arduino Uno. To run the program in the Arduino Uno microcontroller, a USB drive, Arduino IDE software, and an ATmega328p Microcontroller are required. The following program code in C language serves to connect the various modules according to the condition of the components when used.

In the test, once the car is detected at the parking entrance, the servo will open. Then the Infrared sensors installed in each slot will detect the empty parking slots and display them on the LCD screen at the entrance. Thus, the driver can know the available spaces within the parking area with perfect location accuracy from the entrance. Moreover, once the vehicle occupies one of the slots, the LCD screen will reduce the number of empty spaces by one unit, providing information about the number of available spaces at the entrance.

Fig. 5. Initial Display on LCD

In Figure 5 above, the initial display on the LCD when the system starts running is shown. Once the automated parking system and automated parking slot availability have been designed, the next step is to conduct a thorough test of the device. This test involves various conditions to ensure that the device is functioning properly. The results of this test will be displayed on a 16x2 LCD screen that provides information about available parking slots. The test results will be displayed in Table 2 below.

TABLE II. IR SENSOR TESTING OF ENTRANCE AND EXIT DOORS

Test Conditions	IR Sensor	Servo	Gate	LCD Display
Car approaching entrance gate	Detected	Active	Open	"Availabl e Slots: 5"
Car approaching entrance gate (all slots full)	Detected	Inactive	Closed	“Parking Full”
Car moving away from the entrance gate	Not Detected	Inactive	Closed	"Availabl e Slots: 5"
Car approaching the exit gate	Detected	Active	Open	"Availabl e Slots: 5"
Car moving away from the exit gate	Not Detected	Inactive	Closed	"Availabl e Slots: 5"

Table 2 above shows the system response to vehicles approaching the entry and exit gates. When a car approaches the entrance gate and there is an available slot, the IR sensor detects the vehicle, activates the servo motor to open the bar, and the LCD screen displays "Slot Available: 5." If the parking lot is full, the system correctly identifies this and keeps the bar closed, with the LCD display showing "Parking Full." This ensures that vehicles are not allowed to enter when there are no slots available, to prevent overcapacity. The 'Parking Full' condition is shown in Figure 6 below.

Fig. 6. Full Slot Indication

As the vehicle approaches the exit gate, the IR sensor again detects the vehicle, and the servo motor activates to open the bar. The LCD display updates to the original slot availability and shows that the slot has become available after the vehicle exits. This real-time condition of available slots helps in managing parking efficiently, as the system dynamically adjusts the displayed information to reflect the current condition of the parking area.

Infrared sensor testing was also carried out on each parking slot which will be shown in table 3.

TABLE III. IR SENSOR TESTING OF EACH PARKING SLOT

No	Parking Slot	IR Sensor	Slot Condition	LCD Display
1	Slot 1	Detected	Occupied	Slot 1: Occupied
2	Slot 2	Detected	Occupied	Slot 1: Occupied
3	Slot 3	Detected	Occupied	Slot 1: Occupied
4	Slot 4	Detected	Occupied	Slot 1: Occupied
5	Slot 5	Detected	Occupied	Slot 1: Occupied

Table 3 evaluates the performance of the IR sensor in detecting the occupancy status of each parking slot. Each slot is tested for empty and occupied conditions. When the slot is empty, the IR sensor does not detect any vehicles, and the LCD screen displays the slot as empty. Conversely, when the slot is occupied, the IR sensor detects the presence of a vehicle, and the LCD screen updates to show the slot as occupied. This detailed and accurate detection of the status of each slot is essential to provide the driver with precise and up-to-date information about available parking spaces. Here is when the whole system is running and the parking slot detects the presence of a car:

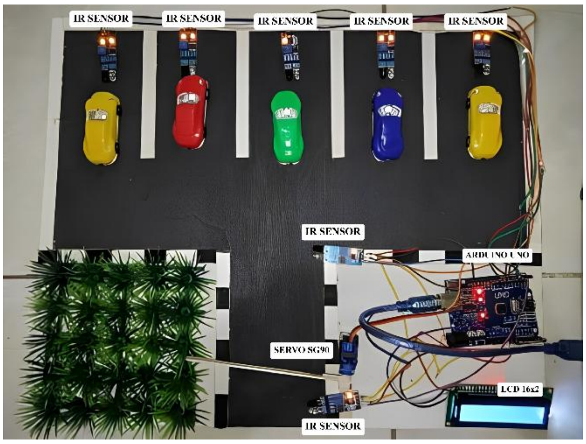


Fig. 7. Prototype When Running

Overall, the system functioned as expected, accurately detecting the presence of vehicles, controlling the servo motors to set the entry and exit bars, and providing real-time updates on the LCD screen regarding the availability of parking slots. This automated approach improves the efficiency of parking management, reduces the time drivers spend searching for available spaces, and helps reduce traffic congestion within the parking area.

Below is when the IR sensors one by one detect the presence of a car, the serial monitor display on the Arduino IDE software and also the LCD will change according to which part of the slot is empty and also filled.

```
Slot status: S1=1, S2=1, S3=0, S4=1, S5=1
Slot status: S1=1, S2=1, S3=0, S4=1, S5=1
Slot status: S1=1, S2=1, S3=0, S4=1, S5=1
Slot status: S1=1, S2=1, S3=0, S4=1, S5=1
Slot status: S1=1, S2=1, S3=0, S4=1, S5=1
Slot status: S1=1, S2=1, S3=0, S4=1, S5=1
Slot status: S1=1, S2=1, S3=0, S4=1, S5=1
Slot status: S1=1, S2=1, S3=0, S4=1, S5=1
Slot status: S1=1, S2=1, S3=0, S4=1, S5=1
Slot status: S1=1, S2=1, S3=0, S4=1, S5=1
Slot status: S1=1, S2=1, S3=0, S4=1, S5=1
Slot status: S1=1, S2=1, S3=0, S4=1, S5=1
Slot status: S1=1, S2=1, S3=0, S4=1, S5=1
Slot status: S1=1, S2=1, S3=0, S4=1, S5=1
Slot status: S1=1, S2=1, S3=0, S4=1, S5=1
Slot status: S1=1, S2=1, S3=0, S4=1, S5=1
Slot status: S1=1, S2=1, S3=0, S4=1, S5=1
Slot status: S1=1, S2=1, S3=0, S4=1, S5=1
Slot status: S1=1, S2=1, S3=0, S4=1, S5=1
Slot status: S1=1, S2=1, S3=0, S4=1, S5=1
Slot status: S1=1, S2=1, S3=0, S4=1, S5=1
Slot status: S1=1, S2=1, S3=0, S4=1, S5=1
Slot status: S1=1, S2=1, S3=0, S4=1, S5=1
Slot status: S1=1, S2=1, S3=0, S4=1, S5=1
Slot status: S1=1, S2=1, S3=0, S4=1, S5=1
Slot status: S1=1, S2=1, S3=0, S4=1, S5=1
```

Fig. 8. Serial Monitor When the Slot Detects a Car

IV. CONCLUSION

This research successfully designed and implemented a smart parking system using IoT technology and infrared sensors. The system is able to detect the availability of parking slots in real time and display the information on the LCD screen at the entrance of the parking area. Tests show that the system can detect vehicles passing through the infrared sensors, control servo motors to open and close the parking portals and provide accurate information about the availability of parking slots on the LCD. The system not only helps drivers find parking spaces quickly but also reduces traffic congestion caused by searching for parking spaces. Thus, the implementation of this automated parking system is expected to increase convenience and efficiency in parking management in urban areas and support the creation of a more environmentally friendly and organized urban environment. In the future, the development of mobile applications, the integration of payment systems, the use of other sensors, and further research can improve the accuracy, efficiency, and functionality of the system, as well as provide greater benefits to users and contribute to the creation of a smart, integrated, and sustainable urban environment.

REFERENCES

[1] M. R. M. Veeramanickam, B. Venkatesh, L. A. Bewoor, Y. W. Bhowte, K. Moholkar, and J. L. Bangare, "IoT based smart parking model using Arduino UNO with FCFS priority scheduling," Measurement: Sensors, vol. 24, pp. 1-7, Dec. 2022, doi: 10.1016/j.measen.2022.100524.

[2] H. T. Alhaq, E. Apriaskar, and D. Djuniadi, "Overspeed detection using arduino uno-based IR infrared sensor," Circuit: Jurnal Ilmiah Pendidikan Teknik Elektro, vol. 7, no. 2, p. 189, Aug. 2023, doi: 10.22373/crc.v7i2.16409.

[3] D. O. Deltania, D. Djuniadi, and E. Apriaskar, "Pengaturan lampu lalu lintas (traffic light) dengan sensor ultrasonik," Jetri?: Jurnal Ilmiah Teknik Elektro, vol. 19, no. 1, pp. 77-95, Aug. 2021, doi: 10.25105/jetri.v19i1.8660.

[4] J. Parmar, P. Das, and S. M. Dave, "Study on demand and characteristics of parking system in urban areas: a review," Journal of Traffic and Transportation Engineering (English Edition), vol. 7, no. 1, pp. 111-124, Feb. 2020, doi: 10.1016/j.jtte.2019.09.003.

[5] A. Ahlam-Rauf, L. Y. K. Wang, K. S. Bee, and W. R. Thang, "Architecture design of an IoT-based smart parking system," in 2023 International Conference on Future Internet of Things and Cloud (FiCloud), Institute of Electrical and Electronics Engineers Inc., 2023, pp. 136-141. doi: 10.1109/FiCloud58648.2023.00028.

[6] C. G. Hoehne, M. V. Chester, A. M. Fraser, and D. A. King, "Valley of the sun-drenched parking space: The growth, extent, and implications of parking infrastructure in Phoenix," Cities, vol. 89, pp. 186-198, Jun. 2019, doi: 10.1016/j.cities.2019.02.007.

[7] R. Ke, Y. Zhuang, Z. Pu, and Y. Wang, "A smart, efficient, and reliable parking surveillance system with edge artificial intelligence on IoT devices," IEEE Transactions on Intelligent Transportation Systems, vol. 22, no. 8, pp. 4962-4974, Aug. 2021, doi: 10.1109/TITS.2020.2984197.

[8] D. B. Mariappan, A. Parihar, and M. Kumar, "Ardino based IoT smart parking system," in 2021 3rd International Conference on Advances in Computing, Communication Control and Networking (ICACCCN), Institute of Electrical and Electronics Engineers Inc., 2021, pp. 651-656. doi: 10.1109/ICAC3N53548.2021.9725371.

[9] R. R. Weinberger, A. Millard-Ball, and R. C. Hampshire, "Parking search caused congestion: Where's all the fuss?," Transp Res Part C Emerg Technol, vol. 120, pp. 1-15, Nov. 2020, doi: 10.1016/j.trc.2020.102781.

[10] F. Al-Turjman and A. Malekloo, "Smart parking in IoT-enabled cities: a survey," Sustain Cities Soc, vol. 49, pp. 1-20, Aug. 2019, doi: 10.1016/j.scs.2019.101608.

[11] M. R. Islam et al., "Smart parking management system to reduce congestion in urban area," in 2020 2nd international conference on electrical, control and instrumentation engineering (ICECIE) , 2020, pp. 1-6.

[12] J. A. Verdejo, F. A. Pecina, E. Alba, and A. G. Arenas, "Optimal allocation of public parking spots in a smart city: problem characterisation and first algorithms," Journal of Experimental and Theoretical Artificial Intelligence, vol. 31, no. 4, pp. 575-597, Jul. 2019, doi: 10.1080/0952813X.2019.1591522.

[13] S. Shafiei, Z. Gu, H. Grzybowska, and C. Cai, "Impact of self-parking autonomous vehicles on urban traffic congestion," Transportation (Amst), vol. 50, no. 1, pp. 183-203, Feb. 2023, doi: 10.1007/s11116-021-10241-0.

[14] O. T. T. Kim, N. H. Tran, C. Pham, T. Leanh, M. T. Thai, and C. S. Hong, "Parking assignment: minimizing parking expenses and balancing parking demand among multiple parking lots," IEEE Transactions on Automation Science and Engineering, vol. 17, no. 3, pp. 1320-1331, Jul. 2020, doi: 10.1109/TASE.2019.2948200.

[15] O. Kula, R. Shabalala, and B. Monchusi, "A Parking ranging system Based on Microcontroller and Infrared Sensor," in International Conference on Electrical, Computer and Energy Technologies (ICECET 2023), Institute of Electrical and Electronics Engineers Inc., 2023, pp. 1-3. doi: 10.1109/ICECET58911.2023.10389407.

[16] W. A. Jabbar, L. Y. Tiew, and N. Y. Ali Shah, "Internet of things enabled parking management system using long range wide area network for smart city," Internet of Things and Cyber-Physical Systems, vol. 4, pp. 82-98, Jan. 2024, doi: 10.1016/j.iotcps.2023.09.001.

[17] S. Saravanan, A. Rishitha, M. Kalaiyarasi, K. V. Bhaskar Reddy, P. Chandrababu, and D. G. Kumar, "Identification of parking space availability by using Arduino Uno for Smart City," in Proceedings of the 2022 3rd International Conference on Communication, Computing and Industry 4.0, C2I4 2022, Institute of Electrical and Electronics Engineers Inc., 2022, pp. 1-5. doi: 10.1109/C2I456876.2022.10051427.

[18] Subathra, A. Arul Prakash, V. Ramalingam, and S. Vignesh, "Arduino Uno-powered parking guidance system with Ultrasonic Sensors," in 2023 1st International Conference on Optimization Techniques for Learning (ICOTL), Institute of Electrical and Electronics Engineers Inc., 2023, pp. 1-6. doi: 10.1109/ICOTL59758.2023.10435209.

[19] R. Palanisamy, C. Pandey, S. Goswami, V. Sharma, and J. Yadav, "Energy efficient low cost arduino based smart parking system with concept of counting," International Journal of Recent Technology and Engineering (IJRTE), vol. 8, no. 2S11, pp. 2551-2553, Sep. 2019, doi: 10.35940/ijrte.B1302.0982S1119.

[20] P. Dhanabalraj, L. Gopinath, G. M. Gowthaman, J. Jashva Sherin, and K. Kumar, "Car parking allocation system using Arduino," in Proceedings - International Conference on Artificial Intelligence and Smart Systems (ICAIS 2021), Institute of Electrical and Electronics Engineers Inc., Mar. 2021, pp. 1223-1227. doi: 10.1109/ICAIS50930.2021.9395865.

[21] M. Placek, "Autonomous vehicles worldwide - statistics & facts," Statisca. Accessed: Jun. 18, 2024. [Online]. Available: https://www.statista.com/topics/3573/autonomous-vehicle-technology/#topicOverview

[22] S. The, E. E. Mathias, J. F. Valesco, K. W. Noverianto, R.

Elta, and B. Juarto, "Accuracy evaluation of infrared and ultrasonic sensors for car parking systems with confusion matrix and RMSE," in 2023 3rd International Conference on Electronic and Electrical Engineering and Intelligent System: Responsible Technology for Sustainable Humanity (ICE3IS), Institute of Electrical and Electronics Engineers Inc., 2023, pp. 448-453. doi: 10.1109/ICE3IS59323.2023.10335280.

[23] R. Widyasari, M. Z. C. Candra, and S. Akbar, "IoT based smart parking system development authorized," in 2019 International Conference on Data and Software Engineering (ICoDSE), 2019, pp. 1-6. doi: 10.1109/ICECIE50279.2020.9309546.

[24] M. T. Zumma, K. B. M. Tahmiduzzaman, R. Khan, M. Rahman, A. A. Suhaeel, and M. F. Ahmed, "An approach to Arduino Uno-based smart car parking system," in 2023 14th International Conference on Computing Communication and Networking Technologies, ICCCNT 2023, Institute of Electrical and Electronics Engineers Inc., 2023. doi: 10.1109/ICCCNT56998.2023.10308179.

[25] L. Soni and A. Kaur, "Solving parking problems with Arduino smart car parking systems," in Proceedings of the 5th International Conference on Inventive Research in Computing Applications (ICIRCA), Institute of Electrical and Electronics Engineers Inc., 2023, pp. 1705-1710. doi: 10.1109/ICIRCA57980.2023.10220937.

[26] M. Ansar, "Automatic Car Parking System," Hackster.io. Accessed: Jun. 18, 2024. [Online].Available: https://www.hackster.io/embeddedlab786/automatic-car-parking-system-95a9dc

ISSN 2355-3286

Long Term Prediction of Extreme Weather with Long Short-Term Memory (LSTM) Model: Effect of Climate Change

Nurulita Purnama Putri[1], Adhi Harmoko Saputro[2]
[1,2]Department of Physics, Faculty of Mathematics and Science, Universitas Indonesia, Depok, Indonesia
[1]nurulita.purnama@ui.ac.id
[2]adhi.harmoko@ui.ac.id

Accepted January 11, 2025
Approved June 24, 2025

***Abstract* — Increasingly intense climate change has increased the frequency and intensity of extreme weather, making weather prediction critical for mitigation and adaptation. This research focuses on long-term prediction of extreme weather using the Long ShortTerm Memory (LSTM) model, as well as evaluating the influence of climate change on prediction accuracy. In this study, historical weather data is used to train and test an LSTM model combined with a RandomForestClassifier. Analysis was carried out using the Mean Squared Error (MSE) evaluation technique for 50 epochs and 8 trials at various threshold values (26, 29, 32, 35, 38, 41, 44, 47). The research results show that the LSTM model is able to predict extreme weather with an accuracy of up to 100%. Apart from that, this research also predicts daily rainfall in Bandung City through the process of data collection, preprocessing, normalization and evaluation using Root Mean Squared Error (RMSE) and Mean Absolute Error (MAE). This model produces an RMSE of 4.24 mm and an MAE of 2.72 mm, indicating quite good predictions. It is hoped that this research can make a significant contribution to the field of meteorology and can be developed further by adding parameters or other methods to improve the quality of predictions. Suggestions are given to increase the amount of data used to obtain better prediction results in the future.**

***Index Terms*— Weather Prediction; Long Short Term Memory (LSTM); Climate Change; Extreme Weather.**

I. INTRODUCTION

Technology is developing very quickly, producing new knowledge that is useful in areas such as education, economics, and meteorology [1]. These developments and changes aim to facilitate human activities [2]. Artificial intelligence has emerged as an algorithmic system designed and developed to have the ability to innovate in various fields, be it computers or machines, so that it is able to think like humans or even surpass them [3].

As the amount of data increases, accuracy also increases. Weather is the atmospheric conditions within a limited and specific time span [4]. Understanding the weather situation is very useful because it provides knowledge for early preventive action when extreme weather occurs [5]. The development of increasingly sophisticated technology opens up opportunities to increase the accuracy of weather predictions [6]. Weather prediction is an effort to understand the development of atmospheric conditions from the past, present and future to anticipate abnormal conditions [7].

Understanding weather analysis and prediction is critical, and research in this area continues to grow. The accuracy obtained using ensemble learning reached 81.21% with an MSE of 18.79%. With the decision tree algorithm forest method, accuracy reached 82.38% with an MSE of 17.62%, while the deep learning method produced an accuracy of 82.92% with an MSE of 17.08%. MSE (Mean Squared Error) is a metric used to measure the extent to which the model matches actual observation data, as well as how accurate the weather predictions produced by the model are [8]. Apart from that, the process of integrating weather radar monitoring throughout Indonesia still requires quite a long time [9].

Predicting extreme weather is a challenge for modeling experts in Indonesia and around the world [10]. Extreme weather is a complex problem due to the small probability of its occurrence, which causes the models developed to often have low accuracy [11]. The main aim of this research is to increase accuracy in predicting and analyzing extreme weather in order to provide more precise predictions in the future [12].

This research uses a deep learning method, namely the Long Short-Term Memory (LSTM) algorithm, which was further developed based on data. LSTM is an artificial neural network method that utilizes information from the past to be processed in the form of sequential data [13]. By using the LSTM algorithm, the accuracy of weather analysis can be improved, supporting various sectors in making more informed decisions. The high accuracy of extreme weather predictions can provide more accurate information, thereby reducing the risk of extreme weather impacts. The use of artificial intelligence technologies such offer learning and data processing with Random Forest

Classifier models for weather analysis and prediction offers the possibility of better planning. The results of this research have great potential to improve understanding of weather information, as well as the contribution of artificial intelligence technology with deep learning methods and LSTM algorithms in better weather analysis and prediction.

II. Literature Review

A. Extreme Weather and Climate Change

Extreme weather is an atmospheric event that is outside the historical norms of a region and can cause significant impacts on the environment and human life [14]. Phenomena such as storms, floods, heat waves and droughts are examples of extreme weather [15]. Global climate change, caused in large part by human activities such as burning fossil fuels and deforestation, has increased the frequency and intensity of extreme weather events [16]. Global warming, increasing concentrations of greenhouse gases, and changing rainfall patterns are some of the factors contributing to this climate change.

B. Long Short Term Memory (LSTM)

LSTM is another type of module of Recurrent Neural Network [17]. LSTM was created by Hochreiter and Schmidhuber [18]. Unlike a single hidden layer in an RNN, LSTM stores information in the control unit outside the normal flow of the RNN [19]. LSTM also has a chain structure like the RNN structure, the difference lies in the structure of the repetition module [20]. LSTM is suitable for problems that have long-term dependencies [21]. Considering long-term information is the inherent behavior of LSTM [22]. In the LSTM structure, there is a memory block which contains three gates (memory gate, forgetting gate, and output gate). On the other hand, LSTM also has a memory unit, which functions to control the transfer of information to the next stage [23].

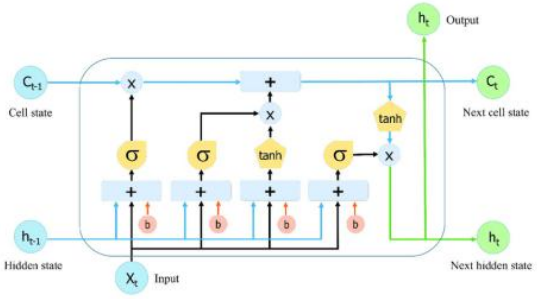


Fig. 1. LSTM Architecture

Here's how the LSTM architecture works from Figure 2 which is explained by the equation below [24]:

- Forget Gate
  The forget gate determines how much information was stored at a previous time or information that was removed.
  $$f_t = \sigma(w_f \times [h_{t-1}, x_t] + b_f) \quad (1)$$
  where:
  w_f: The weight of the foregate gate matrix
  x_t: Input data
  h_{t-1}: Output of the previous memory block
  b_f: Bias from foregate gate
  σ : Sigmoid function
- Input Gates
  The gate input determines how much information is obtained from $x_t$what is stored in the cell state c_t:
  $$i_t = \sigma(w_f \times [h_{t-1}, x_t] + b_i) \quad (2)$$
- Output Gate
  The gate output has a relationship to the output result h_t.
  $$\tilde{c}_t = tanh(w_c \times [h_{t-1}, x_t] + b_c) \quad (3)$$
  $$c_t = f_t \times c_{t-1} + i_t \times \tilde{c}_t \quad (4)$$
  $$h_t = o_t \times \tanh(c_t) \quad (5)$$
  From the equation above, it can be seen that the LSTM input not only produces output h_{t-1} from the hidden layer neurons in the last stage but also contains the memory unit values in the LSTM unit. LSTM can effectively avoid gradient loss, can remember long-term historical information, and is suitable for long-term time series data more effectively.

C. Weather Prediction Using LSTM

In the context of weather prediction, LSTM can utilize historical weather data to identify patterns and trends that can be used to predict future weather conditions [25]. By considering variables such as temperature, humidity, atmospheric pressure, and wind speed, LSTM can produce more accurate prediction models compared to traditional statistical methods [26]. LSTM's advantage in handling sequential data makes it a very useful tool in predicting extreme weather events that are influenced by many interrelated factors [27].

D. Physical Theory in Weather Prediction

Weather predictions are strongly influenced by the physical laws that govern atmospheric dynamics. Some relevant physical theories include:

- Ideal Gas Law: States that the pressure (P), volume (V), and temperature (T) of a gas are related to each other, which is very important in understanding atmospheric dynamics. The equation is [28]:
  $$PV = nRT \quad (6)$$
  Where(P) is the pressure , (V)is the volume, (n)is the number of moles of gas, (R)is the ideal gas constant, and (T)is the temperature in Kelvin.

- Laws of Thermodynamics: These principles govern the transfer of energy and phase changes of water in the atmosphere, such as condensation and evaporation, which influence cloud formation and precipitation. An example of an equation is [29]:

$$Q = mc\Delta T \quad (7)$$

where (Q)is the heat added or released, (m)is the mass, (c)is the heat capacity, and (ΔT) is a change in temperature.

- Newton's Laws of Motion: Used to model the movement of air in the atmosphere, including winds and vertical air currents that can influence the weather. The basic equation is [30]:

$$F = ma \quad (8)$$

where (F)is the force, (m)is the mass, and (a) is the acceleration.

E. Evaluation Criteria

To evaluate the performance of the model with the method used, the author uses mean absolute error (MAE) and root mean square error (RMSE) with the following equation:

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i' - y_i)^2} \quad (9)$$

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i' - y_i| \quad (10)$$

III. Research Method

This research included in the quantitative research category uses the Long Short-Term Memory (LSTM) algorithm. LSTM is a type of artificial neural network architecture, which is specifically designed to handle time sequence data processing.

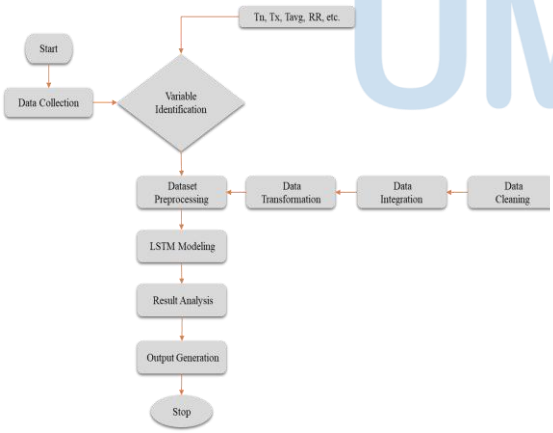


Fig. 2. Research Method

The stages in the research process are as follows:

- Data collection: At this stage, data will be collected before proceeding to the next process. Download data from 1 January 2023 to 31 May 2024 in Bogor City obtained from the BMKG website (http://dataonline.bmkg.go.id/dashboard_user).
- Variable identification: This stage involves identifying the variables to be analyzed in the research. Variables can be numeric or non-numeric data.
- Dataset pre-processing: This stage involves cleaning the data that has been collected by removing incomplete or irrelevant data. This includes deleting incomplete data, deleting duplicate data, and deleting data that is not relevant to the research topic.
- Data transformation: This stage involves converting the data into a format that can be used in analysis. This includes converting the data into a format that can be used in an LSTM model.
- Data integration: This stage involves combining data from various sources into a single dataset. This includes combining data from various sources, such as surveys, experiments, and observations.
- Data cleaning: This stage involves removing incomplete or irrelevant data from the dataset. This includes deleting incomplete data, deleting duplicate data, and deleting data that is not relevant to the research topic.
- LSTM Modeling: This stage involves using an LSTM model to analyze the data. The LSTM model is a type of model that can be used to analyze data that has a time structure.
- Result Analysis: This stage involves analyzing the results from the LSTM model to extract insights and conclusions. This includes using statistical analysis techniques to extract insights and conclusions from research results.
- Output Generation: This stage involves producing a report or presentation that describes the results of the research. This report or presentation can be used to share research results with others.

A. Data source

Weather data is taken from the BMKG website, Java Jarat Province climatology station Citeko Meteorological Station, Latitude -6.70000, Longitude 106.85000, Elevation 920. The following is the website address (https://dataonline.bmkg.go.id/home).

B. Population and Sample

The data population includes all daily weather data records, from 1 January 2023 to 31 May 2024, which reaches 1640 data. Data samples are Tn (minimum temperature), Tx (maximum temperature), Tavg (average temperature), RH_avg (average humidity), RR (rainfall), ss (duration of sunlight), ff_x (maximum wind speed), ddd_x (wind direction at maximum speed), ff_avg (average wind speed), ddd_car (most wind direction)

C. Data Retrieval Techniques

Online data collection, taken from the official website of the BMKG (Meteorology, Climatology and

Geophysics Agency) which can be used legally, by downloading the data periodically once a month from 1 January 2023 to 31 May 2024.

D. Data Retrieval Techniques

Weather data is taken from the BMKG website, Java Jarat Province climatology station Citeko Meteorological Station, Latitude -6.70000, Longitude 106.85000, Elevation 920. The following is the website address (https://dataonline.bmkg.go.id/home). The following data is an overview and explanation of the data analysis techniques that will be carried out in this research:

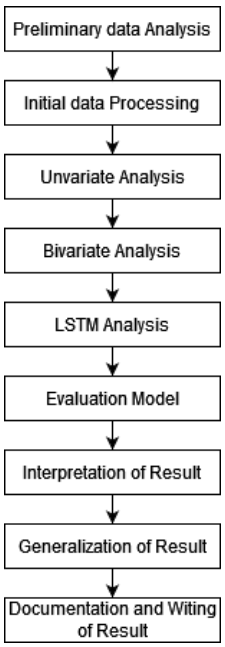


Fig. 3. Data Analysis Techniques

The following are the data analysis techniques for this research:

- Data analysis: to understand the characteristics of weather data.
- Data processing: ensuring data is in good condition.
- Univariate analysis: understanding the distribution between variables, including patterns and trends in the data.
- Bivariate analysis: understanding the correlation between variables.
- LSTM Analysis: train a model with an LSTM algorithm.
- Evaluation Model: evaluating includes the use of metrics such as MSE.
- Interpretation of results: predicting future weather using the LSTM model.
- Generalization of results: reviewing the results of sample weather data to the population as a whole.
- Documentation and writing of results: making conclusions and documenting the results that have been obtained.

IV. Results and Discussion

A. Data retrieval

Data was taken via the official BMKG website which can be used legally, along with the website address (https://dataonline.bmkg.go.id/home) by downloading once a month.

B. Data preprocessing

The first step in this process is to load the cleaned data. This dataset contains important weather variables such as average temperature (Tavg) in degrees Celsius (°C), average relative humidity (RH_avg) in percentage (%), daily rainfall (RR) in millimeters (mm), sunshine duration (ss) in hours (hrs), and daily maximum wind speed (ff_x) in meters per second (m/s). A sample excerpt of the data is shown in Table 1, which includes daily records from January 1, 2023, to May 31, 2024.

TABLE I. CLEAN DATA READING CODE

No	Date	Tavg	RH_avg	RR	ss	ff_x
1	01-01-2023	19.6	93	31.40	0.0	3.0
2	01-02-2023	20.2	96	1.60	0.0	4.0
3	01-03-2023	19.5	96	1.00	6.0	4.0
4	01-04-2023	20.3	90	10.80	0.0	4.0
5	01-05-2023	21.3	81	9.30	0.4	4.0
…	…	…	…	…	…	…
513	05-25-2024	23.0	84	9.90	2.0	4.0
514	05-28-2024	23.1	84	6.75	6.1	4.0
515	05-30-2024	23.0	82	0.00	0.5	4.0
516	05-31-2024	23.1	84	0.00	7.4	4.0

In the table, the numeric data are initially stored as floating-point numbers, for example, temperature values like 19.6°C and rainfall amounts such as 31.40 mm. To facilitate easier classification of weather conditions, these floating-point values are converted to integer values by rounding to the nearest whole number. For instance, a temperature of 19.6°C is rounded to 20°C, and rainfall of 31.40 mm becomes 31 mm. This conversion simplifies subsequent analysis and speeds up the classification process.

Following this conversion, weather conditions can be categorized based on a comparison between rainfall (RR) and sunshine duration (ss). If the rainfall amount is greater than the sunshine duration, the day is classified as a rainy day. If both values are equal, the day is considered cloudy. If the rainfall is less than the sunshine duration, the day is classified as sunny.

Moreover, to identify extreme weather events, the average values for each weather variable over the observation period are calculated. These averages serve as a baseline to determine upper and lower threshold limits that define extreme conditions, such as extremely high or low temperatures and unusually heavy rainfall. These thresholds are useful for further analysis and decision-making. Data after data type change:

TABLE II. CODE TO CHANGE FLOAT DATA TO INTEGER

No	Date	Tavg	RH_avg	RR	ss	ff_x
1	01-01-2023	19	93	31	0	3
2	02-02-2023	20	96	1	0	4
3	03-03-2023	19	96	1	0	4
4	04-04-2023	20	90	10	0	4
5	05-05-2023	21	81	9	0	4

Table 2 shows the data after converting floating-point values to integers for easier processing. The table includes daily records of temperature (Tavg in °C), relative humidity (RH_avg in %), rainfall (RR in mm), solar intensity (ss in hours), and wind speed (ff_x in m/s).

Weather conditions are classified into three categories: rainy, cloudy, and clear. A day is considered rainy if the rainfall (RR) exceeds the solar intensity (ss), cloudy if both values are equal, and clear if the rainfall is less than the solar intensity. The table displays sample data after this conversion, which facilitates the classification process. Weather data with conditions:

TABLE III. RESULTS OF WEATHER CONDITIONS

No	Date	Tavg	RH_avg	RR	ss	ff_x	Wx cond.
1	01-01-2023	19	93	31	0	3	Rain
2	01-02-2023	20	96	1	0	4	Rain
3	01-03-2023	19	96	4	0	4	Rain
4	01-04-2023	20	90	10	0	4	Rain
5	01-05-2023	21	81	9	0	4	Rain
…	…	…	…	…	…	…	…
512	05-25-2024	23	84	9	2	4	Rain
513	05-28-2024	23	84	6	6	4	Cloudy
514	05-29-2024	23	83	4	7	3	Light
515	05-30-2024	23	82	0	6	4	Light
516	05-31-2024	23	84	0	7	4	Light

Table 3 presents a detailed overview of daily weather conditions recorded over the study period from January 1, 2023, to May 31, 2024. Each row contains data for a specific date, including the average temperature (Tavg in °C), average relative humidity (RH_avg in %), rainfall (RR in mm), solar radiation or sunshine duration (ss in hours), and maximum wind speed (ff_x in m/s). These meteorological parameters are used to classify daily weather conditions into categorical labels: "Rain," "Cloudy," or "Light".

For example, on January 1, 2023, the recorded values were a temperature of 19°C, humidity of 93%, rainfall of 31 mm, no sunlight (0 hours), and wind speed of 3 m/s, leading to a "Rain" classification. Toward the end of the dataset, we observe decreasing rainfall and increasing sunshine values. For instance, on May 30, 2024, the data shows 23°C temperature, 82% humidity, 0 mm rainfall, 6 hours of sunshine, and 4 m/s wind speed, categorized as "Light" weather.

This categorization is a precursor to the development of predictive models, as it enables the identification of weather patterns and assists in detecting extreme events. The qualitative weather condition in the "Wx cond." column is derived from logical thresholds primarily influenced by the balance between rainfall and sunlight thus forming a foundational step for further quantitative analysis and machine learning classification.

TABLE IV. AVERAGE VALUE RESULTS

No	Date	Tavg	RH_avg	RR	ss	ff_x	Wx cond.	Avg
1	01-01-2023	19	93	31	0	3	Rain	29.40
2	01-02-2023	20	96	1	0	4	Rain	24.36
3	01-03-2023	19	96	4	0	4	Rain	24.84
4	01-04-2023	20	90	10	0	4	Rain	25.04
5	01-05-2023	21	81	9	0	4	Rain	23.20
…	…	…	…	…	…	…	…	
512	05-25-2024	23	84	9	2	4	Rain	24.40
513	05-28-2024	23	84	6	6	4	Cloudy	24.79
514	05-29-2024	23	83	4	7	3	Light	24.26
515	05-30-2024	23	82	0	6	4	Light	23.06
516	05-31-2024	23	84	0	7	4	Light	23.70

Table 4 presents the daily average values of key weather parameters, including average temperature (Tavg, in °C), relative humidity (RH_avg, in %), rainfall (RR, in mm), sunlight duration (ss, in hours), and maximum wind speed (ff_x, in m/s), along with the qualitative weather condition (Wx cond.). The final column, "Avg," represents a composite index calculated from the combined values of the aforementioned weather parameters. This index is used as a quantitative reference to evaluate and classify daily weather conditions.

For example, on January 1st, 2023, the average temperature was 19°C, relative humidity was 93%, rainfall reached 31 mm, sunlight duration was 0 hours, and the maximum wind speed was 3 m/s. The weather condition was classified as "Rain," with a calculated average index value of 29.40. In contrast, on January 2nd, 2023, despite the temperature increasing to 20°C, the rainfall dropped to just 1 mm, resulting in a lower average value of 24.36.

Across the full dataset of 516 days, the highest "Avg" value recorded was 45.0, indicating severe weather conditions, likely driven by intense rainfall or an extreme combination of other factors. The lowest "Avg" value was 18.84, reflecting calm and possibly clear weather. The overall mean of the "Avg" values is approximately 24.69, which is rounded to 25 and used as a standard threshold to differentiate between normal and extreme weather conditions in West Java.

This average weather index plays a crucial role in training and evaluating machine learning models, such as LSTM and RandomForestClassifier, by providing a consistent numerical basis for labeling weather as either "normal" or "extreme" based on real-time parameter combinations.

Following the detailed breakdown of daily weather averages in Table 4, an overall statistical summary is provided in Table 5 to establish a clearer reference for normal and extreme weather conditions based on quantitative thresholds.

TABLE V. AVERAGE VALUE RESULTS

Highest (Average) Score: 45.0
Lowest (Average) Score: 18.84
Value (Average) of average value: 24.688820116054156

Table 5 presents the summary statistics of the average daily weather scores. The highest average score recorded is 45.0 units, the lowest is 18.84 units, and the overall mean average is approximately 24.69 units. Based on this, the typical daily weather value for West Java is around 24.69 units.

This average value serves as a key threshold for classifying daily weather conditions into either "normal" or "extreme." Days with an average score significantly above this threshold (e.g., above 26 or 29 units) are more likely to be associated with extreme weather events, particularly heavy rainfall. Conversely, days with scores below the threshold typically reflect normal or calm weather conditions. This statistical benchmark is used as a reference input for machine learning models such as LSTM and Random Forest to improve accuracy in predicting and labelling weather conditions.

C. Long Short Term Memory Model Result

Before training the Long Short-Term Memory (LSTM) model, it is essential to prepare clean and well-labeled data that accurately reflects daily weather conditions. This involves transforming raw meteorological variables into structured features and categorizing them based on threshold values to distinguish between *normal* and *extreme* weather. These classifications are critical for supervised learning models such as LSTM, which rely on labeled sequences to learn temporal patterns.

TABLE VI. CODE FOR READING CLEAN DATA 26

No	Date	Tavg	RH_avg	RR	ss	ff_x	Wx cond.	Avg	Label
1	01-01-2023	19	93	31	0	3	Rain	29.40	extreme
2	01-02-2023	20	96	1	0	4	Rain	24.36	normal
3	01-03-2023	19	96	4	0	4	Rain	24.84	normal
4	01-04-2023	20	90	10	0	4	Rain	25.04	normal
5	01-05-2023	21	81	9	0	4	Rain	23.20	normal
…	…	…	…	…	…	…	…	…	…
512	05-25-2024	23	84	9	2	4	Rain	24.40	normal
513	05-28-2024	23	84	6	6	4	Cloudy	24.79	normal
514	05-29-2024	23	83	4	7	3	Light	24.26	normal
515	05-30-2024	23	82	0	6	4	Light	23.06	normal
516	05-31-2024	23	84	0	7	4	Light	23.70	normal

For instance, on January 1, 2023, the high rainfall (31 mm), high humidity (93%), and no sunlight led to a composite score of 29.40, which exceeds the threshold and is labeled as *extreme*. In contrast, on January 2, 2023, the rainfall drops to 1 mm and the score is 24.36, thus labelled as *normal*.

This labelling serves as the basis for training and evaluating the LSTM model. Prior to LSTM, a RandomForestClassifier was applied to verify label consistency, achieving 100% accuracy in classifying the normal versus extreme labels based on the input variables. The LSTM model was then trained on this dataset over 50 epochs using a threshold score of ~25 units to forecast weather labels for the next 7 days, learning temporal dependencies and fluctuation patterns in the weather variables.

D. Analysis and prediction

Based on the value from the process, it is 24.68 rounded to 25 as a benchmark for normal daily weather values in the West Java region. Below is a graph of weather conditions. Threshold value 26. Obtained in the graph below, the red dot pattern appears quite a lot. There were 115 conditions above the threshold value of 26

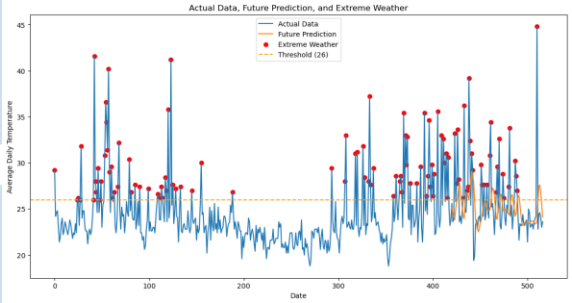


Fig. 4. Threshold Graph 26

Predict weather conditions for the next 7 days using a threshold value of 26, obtained conditions for the next 7 days as shown in the image below.

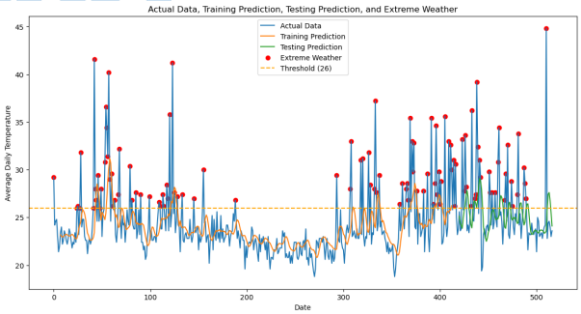


Fig. 5. Threshold 26 Prediction Graph

Next, Table 7 presents the weather prediction results using a threshold value of 26. The time steps show predicted average weather scores along with corresponding weather conditions such as “Bright” (clear) and “Rain” (extreme).

TABLE VII. PREDICTION RESULTS WITH THRESHOLD 26

Time	Step	Prediction	Wx Cond.
520	25.463339	Normal	Bright
521	24.702570	Normal	Bright
522	24.522461	Normal	Bright

Time	Step	Prediction	Wx Cond.
523	24.209877	Normal	Bright
524	27.699635	Extreme	Rain
525	28.909733	Extreme	Rain
526	27.102564	Extreme	Rain

For the next 6 days, the weather forecast is normal, bright conditions and rain on the 7th day with extreme weather. The threshold value can be input as desired with the standard normal weather value in West Java being 24.68 which is rounded to 25. The blue graph is the actual data value, the orange graph is the training data graph, the green graph is the test data graph, and the red graph is the test data graph. at the very end is the actual prediction graph.

From various experiments on the threshold values above, the higher the threshold value entered, the rarer the pattern that appears, then the predicted weather will be normal and clear. It can be seen that the extreme weather patterns that occur throughout 2023 to 2024, the extreme weather that occurs is extremely rainy weather. The following is a table of various experimental threshold values.

To evaluate the sensitivity and precision of the model in detecting extreme weather events, a series of experiments were conducted using various threshold values applied to the composite weather score (referred to as "Avg"). These thresholds serve as classification cut-off points, determining whether a particular day is categorized as experiencing *normal* or *extreme* weather. The goal was to observe how changing the threshold impacts the number of extreme weather events detected and how the model responds in predicting upcoming conditions.

TABLE VIII. EXPERIMENTAL RESULTS OF VARIOUS THRESHOLDS

No	Treshlod value	Emerging Extreme Weather Patterns	Predict conditions that will occur
1	26	115	normal conditions : 6, Rain under extreme conditions: 1
2	29	94	normal conditions : 7, Rain under extreme conditions: 7
3	32	52	normal conditions : 7, Rain under extreme conditions: 7
4	35	33	normal conditions : 7, Rain under extreme conditions: 7
5	38	29	normal conditions : 7, Rain under extreme conditions: 7
6	41	21	normal conditions : 7, Rain under extreme conditions: 7
7	44	15	normal conditions : 7, Rain under extreme conditions: 7
8	47	9	normal conditions : 7, Rain under extreme conditions: 7

Table 8 presents the detailed results of these experiments, where threshold values range from 26 to 47. As the threshold increases, the number of detected extreme weather patterns decreases significantly. At a threshold of 26, a total of 115 extreme weather events are identified. This indicates a highly sensitive setting where even moderately high weather scores are flagged as extreme. However, such sensitivity might increase false positives. As the threshold rises to 29, the number of detected extreme events drops to 94, and at 47, only 9 events are flagged, indicating a sharp decline in identified extremes representing a more conservative approach that reduces false alarms but may miss certain impactful events.

Across all threshold levels, the model consistently forecasts seven days of weather conditions, maintaining a stable predictive routine. The distinction lies in the classification: with lower thresholds, the forecast includes more days labelled as "rain under extreme conditions," whereas at higher thresholds, the number of such labels decreases. For example, at a threshold of 29, the model predicts seven days of rain classified as extreme, while still recognizing some days as normal. This consistency in forecasting behaviour despite varying thresholds indicates that the model is adaptive and responsive to new classification criteria.

These findings highlight the importance of careful threshold selection in weather modelling. Lower thresholds improve sensitivity and are useful for early warning systems, especially in areas vulnerable to frequent rainfall or fluctuating weather. In contrast, higher thresholds favour specificity, minimizing false alarms and better serving scenarios where only the most severe conditions should be flagged. Therefore, tuning the threshold value is essential to align the model with specific operational goals whether to emphasize precautionary warnings or prioritize alert accuracy.

E. Interpretation of Result

The results obtained by this research, from data from January 1 2023 to June 31 2024, by inputting the desired threshold value, can provide output information that matches the input value. The normal weather value obtained from this research is 24.68, rounded up to 25. The higher the threshold value, the more extreme the weather and the less frequently it occurs. The method used is Long Short-Term Memory, to predict and read weather patterns. Determine the occurrence of extreme or normal weather by calculating the average value in each column, because each attribute has a mutually supporting relationship. Like the picture below.

TABLE IX. NORMAL WEATHER

No	Date	Tavg	RH_avg	RR	ss	ff_x	Wx cond.	Avg	Label
6	01-07-2023	22	79	0	6	3	Light	22.0	extreme
7	01-08-2023	21	88	0	5	3	Light	23.4	normal
9	01-10-2023	22	83	1	2	3	Light	22.2	normal
10	01-11-2023	21	83	0	5	4	Light	22.6	normal
11	01-12-2023	22	85	3	4	4	Light	23.6	normal
...	...	...	...	...	...	...	...	...	...

No	Date	Tavg	RH_avg	RR	ss	ff_x	Wx cond.	Avg	Label
511	05-26-2024	22	90	0	0	2	Cloudy	22.8	normal
513	05-28-2024	23	84	6	6	4	Clear	24.6	normal
514	05-29-2024	23	83	4	7	3	Light	24.0	normal
515	05-30-2024	23	82	0	6	4	Light	23.0	normal
516	05-31-2024	23	84	0	7	4	Light	23.6	normal

On 01 – 07 – 2023 the Tavg (average temperature) value is 22°C, RH_avg (humidity) has a value of 79% indicating very humid, RR (rainfall) has a value of 0 mm, SS (sunlight intensity) has a value of 6 hours, and ff_x (maximum wind speed) has a value of 3 m/s. The average obtained is 22.0, so the condition is normal. Rainy weather conditions are caused because the RR (rainfall) value is higher than the ss (sunlight intensity) value according to a journal and vice versa. Also supported by the RH_avg (humidity) value, which is 79% very humid, and other related variables. Then following are the results of extreme weather conditions.

TABLE X. EXTREME WEATHER

No	Date	Tavg	RH_avg	RR	ss	ff_x	Wx cond.	Avg	Label
0	01-01-2023	19	93	31	0	3	Rain	29.2	extreme
24	01-25-2023	19	98	6	4	3	Rain	26.0	extreme
25	01-26-2023	20	90	17	0	4	Rain	26.2	extreme
27	01-28-2023	20	96	14	0	0	Rain	26.0	extreme
28	01-29-2023	20	97	39	0	3	Rain	31.8	extreme
…	…	…	…	…	…	…	…	…	…
481	05-26-2024	22	90	55	0	2	Rain	33.8	extreme
487	05-02-2024	23	91	34	0	3	Rain	30.2	extreme
488	05-03-2024	23	88	25	4	3	Rain	28.6	extreme
489	05-04-2024	22	87	17	6	3	Rain	27.0	extreme
516	05-25-2024	21	93	107	0	3	Rain	44.8	extreme

On 01-01-2023, there were 19 extreme weather conditions recorded, exceeding the monthly average of 29 normal conditions. The average temperature (Tavg) was 25 °C, indicating normal thermal conditions. However, the daily rainfall (RR) reached 31 mm, which qualifies as heavy rainfall and could potentially trigger flooding in affected regions.

To predict such extreme weather events, the process involves data collection, preprocessing, normalization, and evaluation using well-established metrics: Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE). These metrics are widely adopted in meteorological time-series prediction due to their interpretability and sensitivity to error magnitudes.

TABLE XI. ERROR CALCULATION RESULTS

Root Mean Error (RMSE)	4.2423 **mm**
Mean Absolute Error **(MAE)**	2.7247 mm

The results indicate the highest accuracy with an RMSE of 4.24 mm and an MAE of 2.72 mm. These values demonstrate reasonably good predictions, as smaller MAE and RMSE values correspond to higher prediction accuracy.

V. CONCLUSION

The results of this research show that artificial intelligence with the Long Short Term Memory (LSTM) algorithm applied together with RandomForestClassifier on historical weather data can predict extreme weather with an accuracy of up to 100%. Analysis using the Mean Squared Error (MSE) model evaluation technique with 50 epochs and 8 trials at various threshold values (26, 29, 32, 35, 38, 41, 44, 47) shows a different pattern from the normal average of 24.68 in the West Java region. In predicting daily rainfall in Bandung City, data collection, preprocessing, normalization and evaluation were carried out using Root Mean Squared Error (RMSE) and Mean Absolute Error (MAE). This research produces the highest accuracy value with an an RMSE of RMSE of 4.24 mm and an MAE of 2.72 mm. These results show quite good predictions, because the smaller the MAE and RMSE values, the better the prediction accuracy.

The author hopes that this research can be further developed by other researchers in the future. The suggestion given is to add several parameters or other methods that can improve the quality of research and compare which method is superior. Apart from that, increasing the amount of data used is expected to help obtain better prediction values.

ACKNOWLEDGMENT

The author would like to express sincere gratitude to all parties who have supported the completion of this research. Special thanks are extended to the Meteorology, Climatology, and Geophysics Agency (BMKG) of Indonesia for providing access to relevant weather data used in this study. The author also wishes to acknowledge the guidance and encouragement provided by academic supervisors and mentors throughout the research process.

Appreciation is also extended to Universitas Indonesia for facilitating research resources and to fellow researchers and colleagues whose insights and constructive feedback have enriched the quality of this work. Lastly, heartfelt thanks go to the author's family and friends for their unwavering support, motivation, and understanding during the course of this study.

REFERENCES

[1] Russ, M. (2021). Knowledge management for sustainable development in the era of continuously accelerating technological revolutions: A framework and models. Sustainability, 13(6), 3353.

[2] Scharlemann, J. P., Brock, R. C., Balfour, N., Brown, C., Burgess, N. D., Guth, M. K., ... & Kapos, V. (2020). Towards understanding interactions between Sustainable Development Goals: The role of environment–human linkages. Sustainability science, 15, 1573-1584.

[3] Verganti, R., Vendraminelli, L., & Iansiti, M. (2020). Innovation and design in the age of artificial intelligence. Journal of product innovation management, 37(3), 212-227.

[4] Strauss, S., & Orlove, B. (2021). Up in the air: the anthropology of weather and climate. In Weather, climate, culture (pp. 3-14). Routledge.

[5] Walsh, J. E., Ballinger, T. J., Euskirchen, E. S., Hanna, E., Mård, J., Overland, J. E., ... & Vihma, T. (2020). Extreme weather and climate events in northern areas: A review. Earth-Science Reviews, 209, 1033

[6] Ren, X., Li, X., Ren, K., Song, J., Xu, Z., Deng, K., & Wang, X. (2021). Deep learning-based weather prediction: a survey. Big Data Research, 23, 100178.

[7] Chattopadhyay, A., Nabizadeh, E., & Hassanzadeh, P. (2020). Analog forecasting of extreme-causing weather patterns using deep learning. Journal of Advances in Modeling Earth Systems, 12(2), e2019MS001958.

[8] A. M. Siregar, (2020) "Klasifikasi Untuk Prediksi Cuaca Menggunakan Esemble Learning," Petir, vol. 13, no. 2, pp. 138–147 doi: 10.33322/petir.v13i2.998.

[9] A. Prakasa and F. D. Utami (2019) "BMKG The Integrated Weather Radar Information System of BMKG," J. Telecomunication, Electron. Control Eng., vol. 1, no. 2, pp. 86–96

[10] Kurniadi, A., Weller, E., Salmond, J., & Aldrian, E. (2023). Future projections of extreme rainfall events in Indonesia.

[11] Walsh, J. E., Ballinger, T. J., Euskirchen, E. S., Hanna, E., Mård, J., Overland, J. E., ... & Vihma, T. (2020). Extreme weather and climate events in northern areas: A review. Earth-Science Reviews, 209, 103324.

[12] Fitzsimmons, J. P., Lu, R., Hong, Y., & Brilakis, I. (2022). Construction schedule risk analysis-a hybrid machine learning approach.

[13] Xiang, Z., Yan, J., & Demir, I. (2020). A rainfall-runoff model with LSTM-based sequence-to-sequence learning. Water resources research, 56(1), e2019WR025326.

[14] AghaKouchak, A., Chiang, F., Huning, L. S., Love, C. A., Mallakpour, I., Mazdiyasni, O., ... & Sadegh, M. (2020). Climate extremes and compound hazards in a warming world. Annual Review of Earth and Planetary Sciences, 48, 519-548.

[15] Zscheischler, J., Martius, O., Westra, S., Bevacqua, E., Raymond, C., Horton, R. M., ... & Vignotto, E. (2020). A typology of compound weather and climate events. Nature reviews earth & environment, 1(7), 333-347.

[16] Gulzar, A., Islam, T., Gulzar, R., & Hassan, T. (2021). Climate change and impacts of extreme events on human health: An overview. Indonesian Journal of Social and Environmental Issues (IJSEI), 2(1), 68-77.

[17] Hu, J., Wang, X., Zhang, Y., Zhang, D., Zhang, M., & Xue, J. (2020). Time series prediction method based on variant LSTM recurrent neural network. Neural Processing Letters, 52, 1485-1500.

[18] G. van Houdt, C. Mosquera, and G. Nápoles (2020)"A review on the long short-term memory model," Artificial intelligence review: An international survey and tutorial journal, vol. 53, no. 8, pp. 5929– 5955, doi: 10.1007/s10462-020-09838-1.

[19] Sherstinsky, A. (2020). Fundamentals of recurrent neural network (RNN) and long short-term memory (LSTM) network. Physica D: Nonlinear Phenomena, 404, 132306.

[20] Schultz, M. G., Betancourt, C., Gong, B., Kleinert, F., Langguth, M., Leufen, L. H., ... & Stadtler, S. (2021). Can deep learning beat numerical weather prediction?. Philosophical Transactions of the Royal Society A, 379(2194), 20200097.

[21] Zhao, J., Huang, F., Lv, J., Duan, Y., Qin, Z., Li, G., & Tian, G. (2020, November). Do RNN and LSTM have long memory?. In International Conference on Machine Learning (pp. 11365-11375). PMLR.

[22] Huang, X. (2021). Network Intrusion Detection Based on an Improved Long-Short-Term Memory Model in Combination with Multiple Spatiotemporal Structures. Wireless Communications and Mobile Computing, 2021(1), 6623554.

[23] W. Kong, Z. Y. Dong, Y. Jia, D. J. Hill, Y. Xu, and Y. Zhang (2019) "Shortterm residential load forecasting based on LSTM recurrent neural network," IEEE Trans Smart Grid, vol. 10, no. 1, pp. 841–851, doi: 10.1109/TSG.2017.2753802.

[24] Ambaum, M. H. (2020). Thermal physics of the atmosphere (Vol. 1). Elsevier.

[25] Semeniuk, K., & Dastoor, A. (2020). Current state of atmospheric aerosol thermodynamics and mass transfer modeling: A review. Atmosphere, 11(2), 156.

[26] López, G., & Arboleya, P. (2022). Short-term wind speed forecasting over complex terrain using linear regression models and multivariable LSTM and NARX networks in the Andes Mountains, Ecuador. Renewable Energy, 183, 351-368.

[27] Suleman, M. A. R., & Shridevi, S. (2022). Short-term weather forecasting using spatial feature attention based LSTM model. IEEE Access, 10, 82456-82468.

[28] Rohli, R. V., Li, C., Rohli, R. V., & Li, C. (2021). The Seven Basic Equations in Weather Forecasting Models. Meteorology for Coastal Scientists, 171-185.

[29] Bohren, C., & Albrecht, B. (2023). Atmospheric thermodynamics. Oxford University Press.

[30] Sreeraj, M. S., Sruthi, S. L., & Sumod, S. G. (2024). Modeling in Fluid Mechanics: Atmospheric Modeling Using Navier Stokes Equation. In New Advances in Materials Technologies (pp. 59-86). Apple Academic Press.

[31] A. H. Bukhari, M. A. Z. Raja, M. Sulaiman, S. Islam, M. Shoaib, and P. Kumam, "Fractional neuro-sequential ARFIMA-LSTM for financial market forecasting," IEEE Access, vol. 8, pp. 71326–71338, 2020

ISSN 2355-3286

Design of Greenhouse Prototype Controller and Monitor on Green Mustard Plants IoT

Ahmad Nur Rozzaq[1], Alex Harijanto[2], Maryani[3]
[1,2,3]Faculty of Teacher Training and Education, University Jember, Jember, Indonesia
[1]ahmadnurrozzaq@gmail.com

Accepted January 31, 2025
Approved June 16, 2025

***Abstract*— The prototype greenhouse control and monitoring system for green mustard plants is expected to facilitate greenhouse farmers in farming activities. Control and monitoring are carried out on the intensity of incoming sunlight, greenhouse temperature, greenhouse humidity, and soil moisture. The control and monitoring system is carried out by ESP32, BH1750 sensor, DHT22 sensor, and soil moisture sensor. The control process is assisted by an actuator that will turn on and off automatically according to the value limits entered by the greenhouse owner and the readings from the sensor. The control and monitoring device uses Blynk IoT which is connected to an IoT connection. The purpose of this study is to describe the design process of the controller and monitor of the prototype greenhouse for green mustard plants based on the IoT and to find out the results of the controller and monitor of sunlight intensity, air temperature, air humidity, and soil moisture on green mustard plants in the greenhouse. This study uses quantitative descriptive research. The results of this study are the realization of a prototype greenhouse control and monitor for green mustard plants based on the IoT with the main control system being ESP32. The results of the calibration values of the 4 sensor variables are very satisfactory and can be considered as valid tools. The R_square values of the 4 sensor calibrations, namely the BH1750 light sensor, DHT22_1 temperature sensor, DHT22_2 temperature sensor, DHT22_1 humidity sensor, DHT22_2 humidity sensor, and soil moisture sensor are respectively 0.9936; 0.9689; 0.9665; 0.9412; 0.9451; 0.9574.**

***Index Terms*— Controller; Monitor; ESP32; Blynk Iot; Internet of Things.**

I. INTRODUCTION

Green mustard greens are a type of mustard greens or brassicaceae that is quite popular. Also known as caisim, caisin, or bakso mustard greens, this vegetable can be eaten fresh or processed into pickles, lalapan, and various other dishes[1]. Traditional farmers tend to still plant mustard greens in open environments. As a result, during the rainy season, mustard greens are susceptible to rainwater and are susceptible to disease. While during the dry season, the quality of mustard greens can decrease due to exposure to temperatures that are too high. This requires innovation related to the creation of a new environment that is separated from the outside environment and in accordance with the needs of the growth of mustard greens. One solution is to create a new climate environment that can be implemented by using greenhouse farming.

Greenhouse is a structured building that aims to create environmental conditions that suit the needs of existing plants[2]. A greenhouse is a medium that is isolated from the outside environment, which causes the temperature, humidity, and light intensity trapped inside the greenhouse to be different from the outside environment[3]. Greenhouses are made of transparent materials such as transparent plastic or glass so that sunlight can enter the building[4]. Greenhouses are made with the aim of creating an optimal artificial microenvironment with the growth of a plant[5]. Planting in a Greenhouse is one way to increase agricultural productivity, because in a greenhouse environmental condition, such as: humidity, temperature, light and irrigation in the greenhouse can be controlled[6].

Controlling and monitoring in the greenhouse can be done automatically using IoT. The IoT can be described as a connected network consisting of several interconnected components that form a system and enable the network to detect, capture, distribute, and analyze data[7]. The concept of the IoT can be applied to the ESP32 which is connected to several sensors and actuators that act according to the logic system expected by the user. The use of the IoT can be used to control the intensity of sunlight, room temperature, room humidity, and soil moisture in the greenhouse. This control can be used by automating the opening and closing of the roof, plant lights, plant fans, plant foggers, and plant pumps.

ESP32 is a low-power microcontroller that is useful for developing IoT projects. ESP32 has built-in Wi-Fi and Bluetooth, so no additional modules are needed. This tool is capable of accommodating many sensors and devices with a total of 48 GPIO pins[8]. Blynk IoT is one of the interface platforms that can be used to control and monitor microcontroller projects from Android and IOS[9]. The BH1750 sensor is a sensor used to measure light intensity in lux units[10]; [11].

The DHT22 sensor is a sensor that is capable of measuring temperature with a range of values between -40° C to 125° C and air humidity with a range of values between 0% to 100%[12]. The soil moisture sensor has a function to detect the moisture content of water in the soil[13].

II. METHODS

A. System Block Diagram

As can be seen in Fig. 1 there is information about 3 parts that communicate with each other. The first part is the remote control block. There is Blynk IoT which functions as a controller and monitor carried out by the greenhouse owner and there is also Gmail which functions to receive sensor reading reports by the greenhouse owner and also functions to send messages that the system is in error to the greenhouse technician. The second part is the software block. There is Arduino IDE which functions to enter code into ESP32 and there is also Blynk Cloud which functions as a virtual communication intermediary between ESP32 and Blynk IoT and Gmail. The third part is the hardware block. There is ESP32 as the control center and main control system, there is a BH1750 sensor, DHT22 sensor, and soil moisture sensor, there is also a stepper motor actuator, lights, foggers, and water pumps.

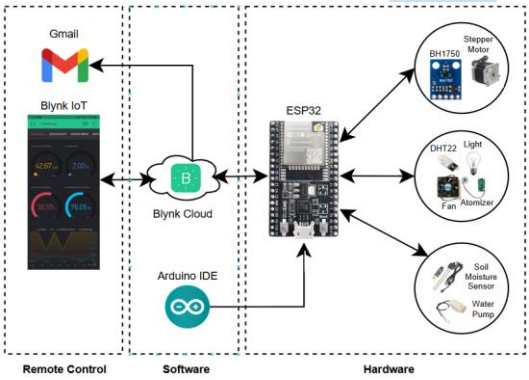


Fig. 1. System Block Diagram

B. Schematic Diagram

In Figure 2 there is an ESP32 that functions as a control center and decision-making center. The ESP32 is connected to several sensors, namely the BH1750 sensor functions to read the intensity of sunlight, the DHT22 sensor functions to read the temperature and humidity of the room, and the soil moisture sensor functions to read the humidity of the soil. There are also several actuators, namely the Nema 17 stepper motor functions to open and close the greenhouse roof made of paranet, the 12V DC lamp functions to replace the role of sunlight, the 5V DC fan functions to circulate the air in the greenhouse, the 5V DC fogger functions to humidify the greenhouse, and the 5V DC pump functions to water the plants.

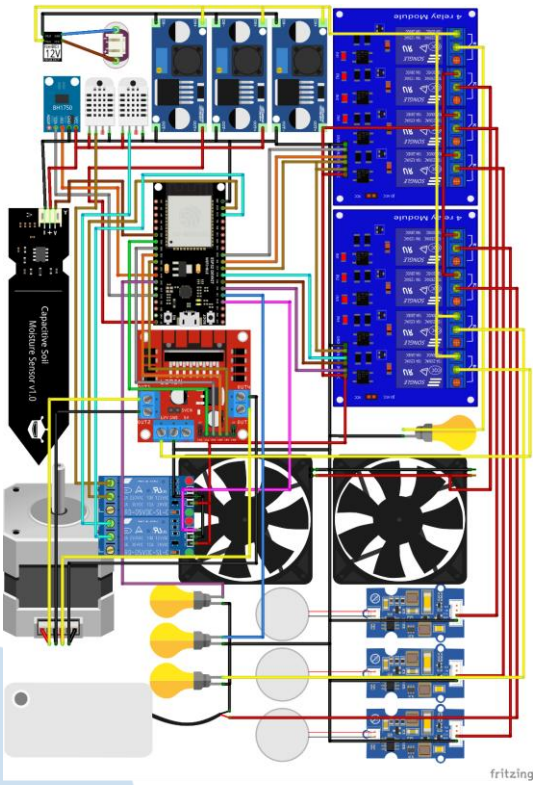


Fig. 2. Schematic Diagram

C. Greenhouse Design, Sensor Placement, and Actuators

In this section, we will explain the greenhouse design from different observation positions. There are also sensor and actuator placement positions in this greenhouse design. The size scale used in drawing this design has been adjusted to the comparison of the original design. The design of the greenhouse will be explained in the Fig. 3 – Fig. 8.

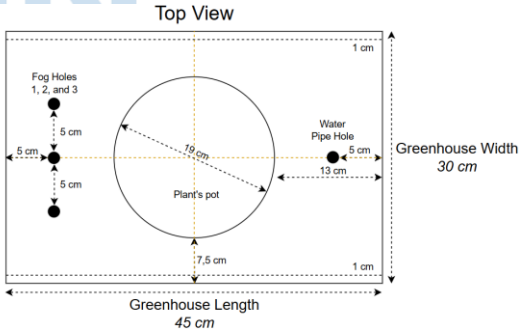


Fig. 3. Greenhouse Design: Top View

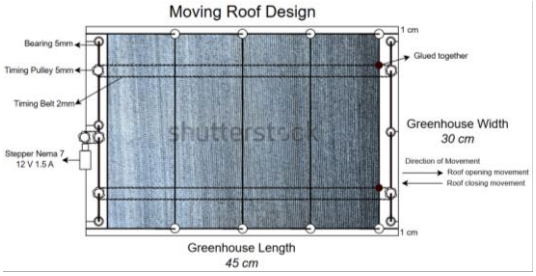


Fig. 4. Greenhouse Design: Moving Roof Design

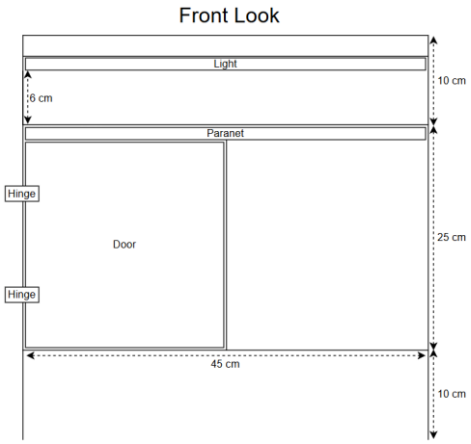


Fig. 5. Greenhouse Design: Front Look

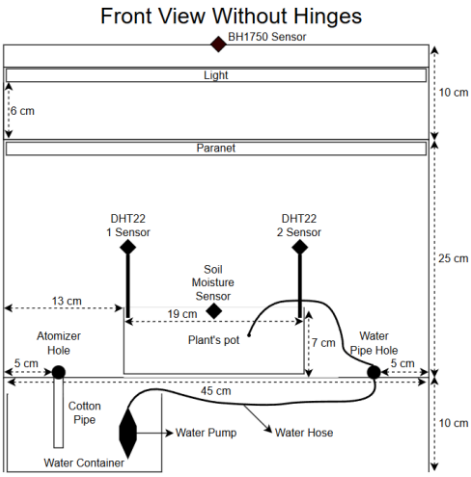


Fig. 6. Greenhouse Design: Front View Without Hinges

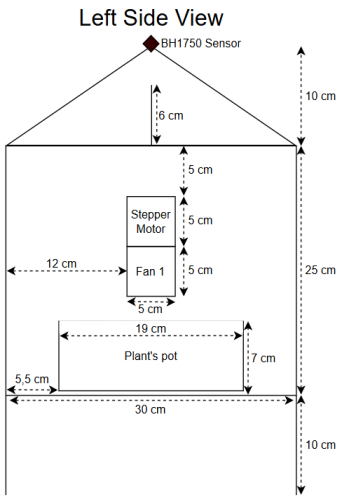


Fig. 7. Greenhouse Design: Left Side View

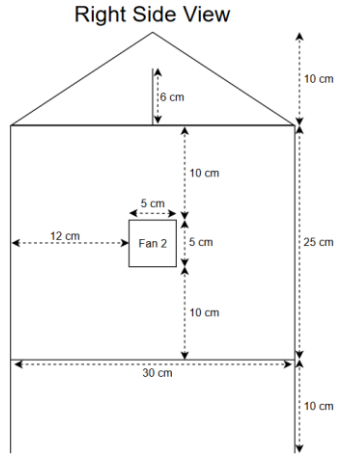


Fig. 8. Greenhouse Design: Right Side View

D. *System Flow*

This section will explain the system algorithm that will be run on this tool. After the greenhouse owner opens the Blynk IoT application. The first step that must be taken is to select the initial condition of the open roof or select the initial condition of the closed roof. The initial condition of the roof is adjusted to the condition of the roof on the greenhouse prototype. The second step is that the greenhouse owner selects the active mode condition that will be run on this system. There are automatic mode and manual mode active modes that can be selected. While the emergency active mode will be run automatically if the sensor experiences an error. The results of the system flow design can be seen in Fig. 9.

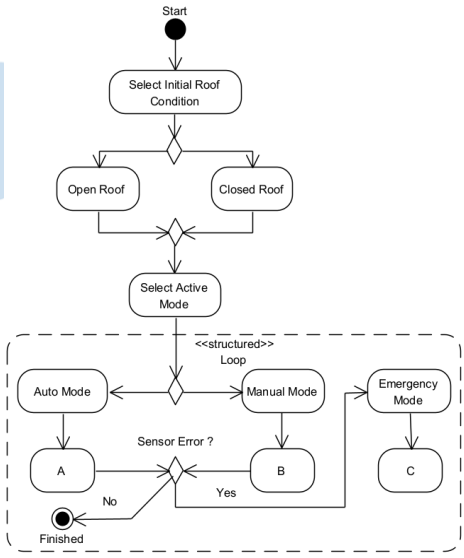


Fig. 9. System Flow

When the automatic mode is run by the greenhouse owner. Then the greenhouse owner can set the limits of the sunlight sensor, the limits of the greenhouse temperature sensor, the limits of the greenhouse humidity sensor, and the limits of the soil humidity sensor. These sensor limits are adjusted to the needs of the plants in the greenhouse building. In this study, the

plants in it are green mustard greens. When the system is running, when the limits of the sunlight sensor are greater than the reading results of the sunlight sensor, the greenhouse roof will be closed, but when the limits of the sunlight sensor are not greater than the reading results of the sunlight sensor, the greenhouse roof will open. In parallel to other system processes, when the limits of the greenhouse temperature sensor are greater than the reading results of the greenhouse temperature sensor, the greenhouse lights will turn off, but when the limits of the greenhouse temperature sensor are not greater than the reading results of the greenhouse temperature sensor, the greenhouse lights will turn on. In parallel to other system processes, when the limits of the greenhouse humidity sensor are greater than the reading results of the greenhouse humidity sensor, the greenhouse fogger and fan will turn off, but when the limits of the greenhouse humidity sensor are not greater than the reading results of the greenhouse humidity sensor, the greenhouse fogger and fan will turn on. In parallel with other system processes, when the soil moisture sensor limit is greater than the soil moisture sensor reading, the greenhouse pump will turn off, but when the soil moisture sensor limit is not greater than the soil moisture sensor reading, the greenhouse pump will turn on. The results of the automatic mode system flow design can be seen in Fig. 10.

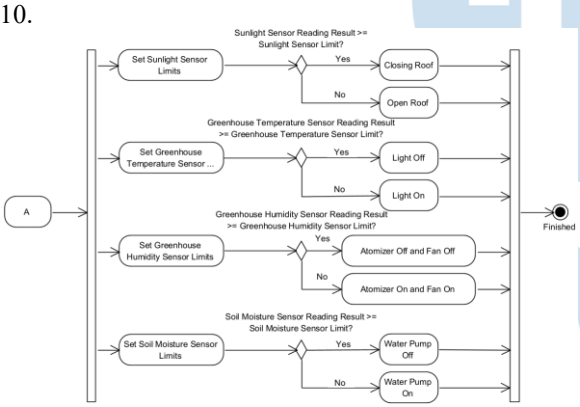


Fig. 10. Automatic Mode System Flow

When manual mode is run by the greenhouse owner. So, the greenhouse owner can manually open and close the greenhouse roof, turn on and off the greenhouse lights, turn on and off the greenhouse fogger, turn on and off the greenhouse fan, and turn on and off the greenhouse pump. The results of the manual mode system flow design can be seen on Fig. 11.

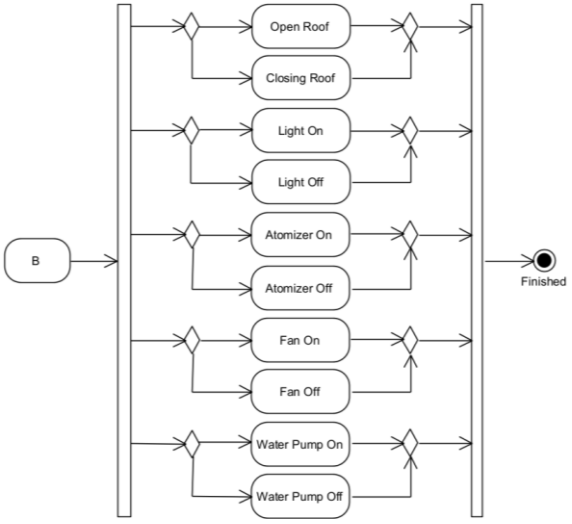


Fig. 11. Manual Mode System Flow

Emergency mode will be automatically activated by the system if there is a sensor that experiences an error. When emergency mode is active, the system will run a sensor error check. When the sensor error check process is running, the system will run the BH1750 error check function, DHT22 error check, and soil moisture sensor error check in parallel. After the system finds the error sensor. The system will automatically select the type of actuator that will be run in automatic mode and manual mode. For more details, see Table I. While the process that occurs in emergency mode can be seen in Fig. 12.

TABLE I. EMERGENCY MODE ACTUATOR

Emergency Mode				
Sensor Error			Actuator is running in Mode -	
BH1750	DHT22	Soil moisture	Automatic	Manual
Error	Normal	Normal	Lights, Fans, Atomizer, Water Pumps	Roof
Normal	Error	Normal	Roof, Water Pump	
Normal	Normal	Error	Roof, Lights, Fans, Atomizer	Water pump
Error	Error	Normal	Water pump	Roof, Lights, Fans, Atomizer
Error	Normal	Error	Lights, Fans, Atomizer	Roof, Water Pump
Error	Error	Error	-	All Manuals

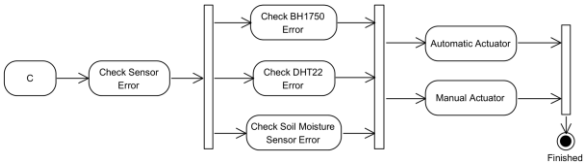


Fig. 12. Manual Mode System Flow

For example, when the BH1750 sensor experiences an error in reading sensor data, while the DHT22 sensor and soil moisture sensor are both still normal. Then the system will automatically run in emergency mode, the system will run the Lights, Fans, Atomizers, Water Pumps actuators in automatic mode, specifically for the roof actuator the system will run it in manual mode. So that the greenhouse owner can open and close the roof manually. Keep in mind at this stage, the Lights, Fans, Atomizers, Water Pumps actuators on and off are still affected by the applicable sensor limits and sensor readings.

E. Blynk IoT App View

On the Blynk IoT application display, it will explain the existing menus. Each menu has a specific task. Keep in mind, the main function of the Blynk IoT application is as a controller and monitor of the greenhouse prototype. So that greenhouse owners can control and limit the amount of sunlight entering the greenhouse, the temperature and humidity that are appropriate for the growth of green mustard plants, and the soil moisture that is appropriate for green mustard plants. Greenhouse owners can also monitor the value of sunlight, greenhouse temperature and humidity, and soil moisture in real time. The appearance of the menus in the Blynk IoT application can be seen in the Fig. 13 – Fig. 17.

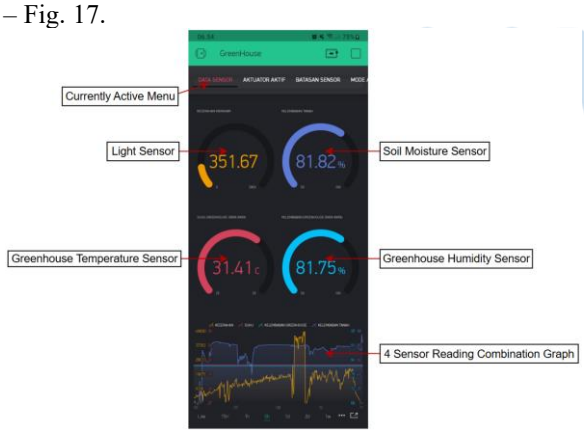


Fig. 13. Blynk IoT: Data Sensor

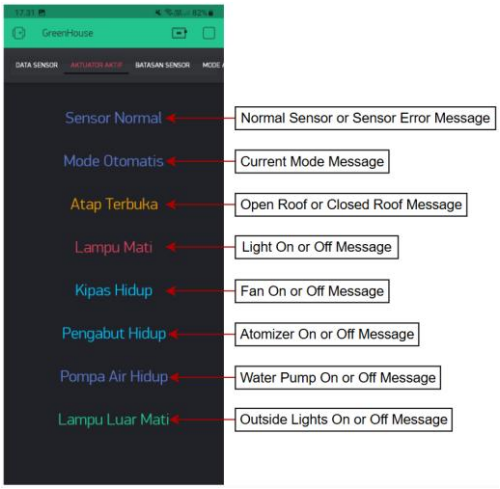


Fig. 14. Blynk IoT: Active Actuator

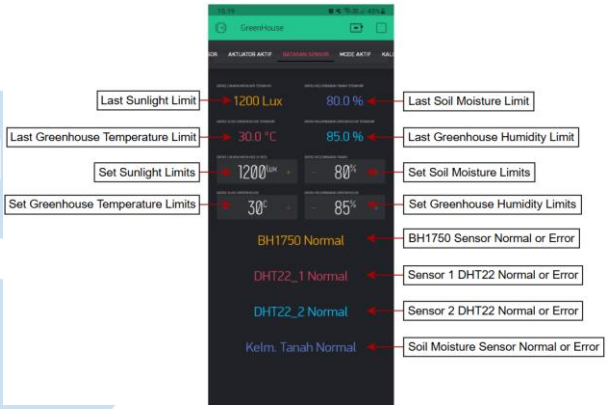


Fig. 15. Blynk IoT: Sensor Limitations

Fig. 16. Blynk IoT: Active Mode

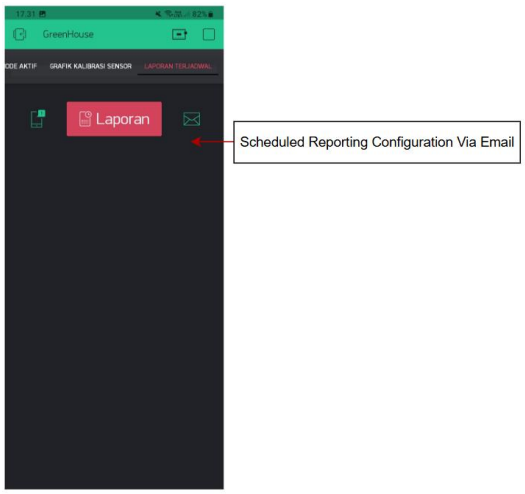


Fig. 17. Blynk IoT: Scheduled Reports

F. *Characteristics of Green Mustard Plants*

Green mustard greens are classified as vegetables that are widely consumed by the Indonesian population, because they taste sweet and have high nutritional content. Green mustard greens contain protein, fat, carbohydrates, fiber, and vitamins[14]. Traditional farmers still plant mustard greens in open environments. During the rainy season, many mustard greens are damaged due to rainwater and disease. Meanwhile, during the dry season, mustard greens are susceptible to being eaten by insects[15]. Green mustard greens require sufficient sunlight intensity, because during their growth, mustard greens require low to warm temperatures (22 - 33 ° C), soil temperatures in the range of 7 - 28 ℃, environmental humidity ± 75% and soil moisture in the range of 60 - 88%. The quality of sunlight exposure is a major factor in the optimal growth of mustard greens.

III. RESULTS AND DISCUSSION

A. *Sensor Calibration Testing*

Calibration of the BH1750 sensor or sunlight intensity sensor was carried out with the AS803 lux meter. The calibration process was carried out 14 times with a data collection interval of every 30 minutes. Data collection was carried out from 05.30 WIB to 12.00 WIB. The BH1750 sensor calibration data with the AS803 lux meter can be seen on Fig. 18. From the calibration results, the value $y = 0{,}8571x + 2436{,}6$, was also obtained, this value will be entered into the calibration function contained in the code that will be uploaded into the ESP32 microcontroller or this system.

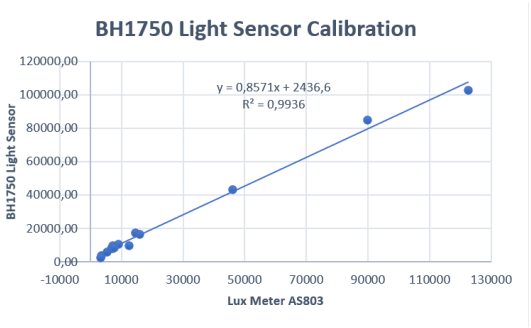


Fig. 18. BHI750 Sensor Calibration Chart

The calibration of the DHT22_1 temperature sensor was carried out with HTC 1 temperature. The calibration process was carried out 14 times with a data collection interval of every 30 minutes. Data collection was carried out from 05.30 WIB to 12.00 WIB. The calibration data of the DHT22_1 temperature sensor with HTC 1 temperature can be seen in Fig. 19. From the calibration results, the value $y = 0{,}8898x + 2{,}4332$, was also obtained, this value will be entered into the calibration function contained in the code that will be uploaded into the ESP32 microcontroller or this system.

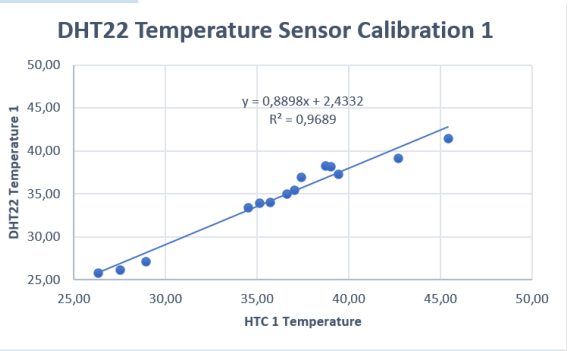


Fig. 19. DHT22 Sensor Calibration Chart 1 Greenhouse Temperature

The calibration of the DHT22_2 temperature sensor was carried out with HTC 1 temperature. The calibration process was carried out 14 times with a data collection interval of every 30 minutes. Data collection was carried out from 05.30 WIB to 12.00 WIB. The calibration data of the DHT22_2 temperature sensor with HTC 1 temperature can be seen on Fig. 20. From the calibration results, the value $y = 0{,}9129x + 1{,}9394$, was also obtained, this value will be entered into the calibration function contained in the code that will be uploaded into the ESP32 microcontroller or this system.

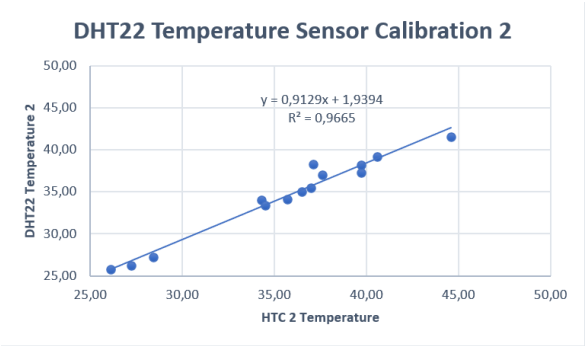


Fig. 20. DHT22 Greenhouse Temperature Sensor Calibration Chart

The DHT22_1 humidity sensor calibration was performed with HTC 1 humidity. The calibration process was performed 14 times with a data collection interval of every 30 minutes. Data collection was carried out from 05.30 WIB to 12.00 WIB. The DHT22_1 humidity sensor calibration data with HTC 1 humidity can be seen in Fig. 21. From the calibration results, the value $y = 0{,}9952x - 4{,}3267$, was also obtained, this value will be entered into the calibration function contained in the code that will be uploaded into the ESP32 microcontroller or this system.

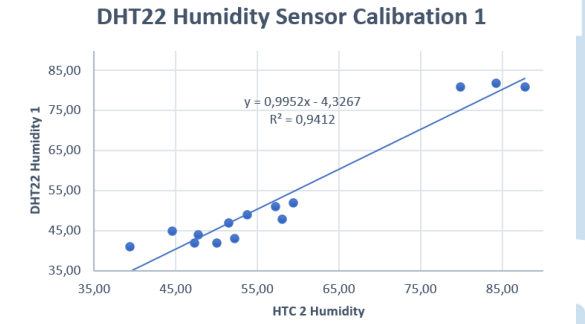


Fig. 21. DHT22 Sensor Graph 1 Greenhouse Humidity

The DHT22_2 humidity sensor calibration was performed with HTC 1 humidity. The calibration process was performed 14 times with a data collection interval of every 30 minutes. Data collection was carried out from 05.30 WIB to 12.00 WIB. The DHT22_2 humidity sensor calibration data with HTC 1 humidity can be seen in Fig. 22. From the calibration results, the value $y = 1{,}0058x - 7{,}9798$, was also obtained, this value will be entered into the calibration function contained in the code that will be uploaded into the ESP32 microcontroller or this system.

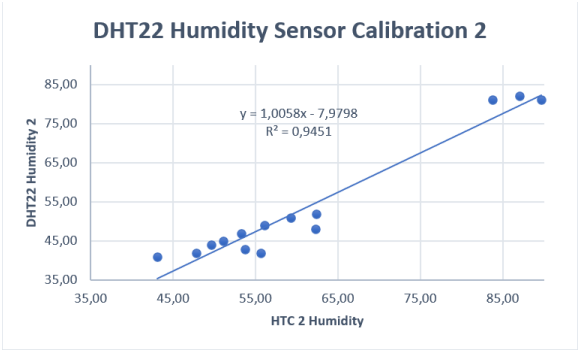


Fig. 22. DHT22 Sensor Graph 2 Greenhouse Humidity

Calibration of soil moisture sensors was done with Grain Moisture Meter AR991. The calibration process was done 8 times. Soil moisture sensor calibration data was done with Grain Moisture Meter AR991 can be seen on Fig. 23. From the calibration results, the value $y = 1{,}0894x - 7{,}4549$, was also obtained, this value will be entered into the calibration function contained in the code that will be uploaded into the ESP32 microcontroller or this system.

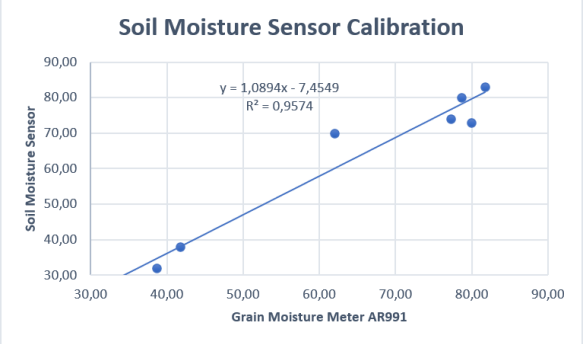


Fig. 23. Soil Moisture Sensor Graph

B. Testing of Greenhouse Prototype Controller and Monitoring Design Tools for Green Mustard Plants Based on IoT

The test was carried out in the Gumuk Kerang Regency housing complex on November 24 2024 to November 26 2024. The test time was carried out 24 hours. This test focuses on collecting data from sensor readings and focuses on the automatic on and off conditions of the actuator according to the sensor limits that have been entered into the Blynk IoT application. Based on research by Gallagher in 1990, mustard plants require low to warm temperatures (22 - 33 ° C), soil temperatures in the range of 7 - 28 °C, environmental humidity ± 75% and soil moisture in the range of 60 - 88% (wb); so the researcher took the sunlight sensor limit of 15000 lux, the greenhouse temperature sensor limit of 30 ° C, the greenhouse humidity sensor limit of 85%, and the soil moisture limit of 80%. Sensor reading data can be seen from Table II to Table V. Meanwhile, the actuator on and off time data can be seen in Table VI to Table VIII.

TABLE II. SUNLIGHT READING

Time	Sunlight Reading		
	24/11/2024	25/11/2024	26/11/2024
00.00.00	0,00	-1,00	6,70
00.30.00	0,00	-1,00	6,55
01.00.00	0,00	-2,00	6,61
01.30.00	0,00	-2,00	6,60
02.00.00	0,00	-2,00	6,54
02.30.00	0,00	-2,00	10,00
03.00.00	0,00	-2,00	10,00
03.30.00	0,00	-2,00	10,00
04.00.00	0,00	-2,00	5,83
04.30.00	0,00	-2,00	5,00
05.00.00	-2,00	-2,00	17,41
05.30.00	1749,17	-2,00	109,76
06.00.00	3652,80	-2,00	295,67
06.30.00	7500,33	-2,00	398,21
07.00.00	10455,36	0,78	775,92
07.30.00	11957,81	1141,67	989,76
08.00.00	15959,53	8698,69	1254,90
08.30.00	16841,51	17289,76	1604,70
09.00.00	18543,85	19848,81	1555,89
09.30.00	10677,61	18630,02	1247,81
10.00.00	9808,87	19342,31	1554,00
10.30.00	17593,33	19820,15	1228,06
11.00.00	20072,60	19150,88	1273,18
11.30.00	22518,44	17750,68	900,06
12.00.00	223,07	11206,35	1296,24
12.30.00	310,68	16969,78	698,36
13.00.00	187,76	15536,63	521,56
13.30.00	293,33	6770,65	407,69
14.00.00	294,84	191,94	172,13
14.30.00	305,94	30,62	266,64
15.00.00	167,55	49,56	385,49
15.30.00	180,10	78,95	200,03
16.00.00	155,57	110,61	194,58
16.30.00	67,38	66,27	116,64
17.00.00	28,33	45,49	56,56
17.30.00	10,36	45,49	15,83
18.00.00	8,33	10,00	13,28
18.30.00	8,33	10,00	13,33
19.00.00	8,39	9,93	13,33
19.30.00	8,23	9,73	13,33
20.00.00	8,24	9,73	13,33
20.30.00	6,16	9,73	13,33
21.00.00	-2,00	9,87	8,36
21.30.00	-2,00	9,81	8,33
22.00.00	-2,00	10,00	12,50
22.30.00	-2,00	9,54	11,67
23.00.00	-2,00	10,00	7,52
23.30.00	-2,00	6,61	7,50

TABLE III. GREENHOUSE TEMPERATURE READING

Time	Greenhouse Temperature Reading		
	24/11/2024	25/11/2024	26/11/2024
00.00.00	0,00	27,63	27,68
00.30.00	0,00	27,46	27,59
01.00.00	0,00	27,33	27,42
01.30.00	0,00	27,20	27,31
02.00.00	0,00	27,09	27,08
02.30.00	0,00	26,98	27,09
03.00.00	0,00	26,88	26,98
03.30.00	0,00	26,83	26,97
04.00.00	0,00	26,76	26,97
04.30.00	0,00	26,68	26,91
05.00.00	26,62	26,59	26,86
05.30.00	27,09	26,71	26,87
06.00.00	27,99	26,99	27,08
06.30.00	29,94	26,99	27,45
07.00.00	36,80	28,04	27,98
07.30.00	37,56	28,94	28,55
08.00.00	36,43	33,48	29,35
08.30.00	37,67	35,76	30,20
09.00.00	39,26	38,11	30,85
09.30.00	37,78	38,95	31,58
10.00.00	41,11	39,02	30,61
10.30.00	41,07	40,40	32,23
11.00.00	40,96	40,49	32,51
11.30.00	41,35	40,28	33,18
12.00.00	33,88	39,62	33,73
12.30.00	32,12	40,74	33,43

Time	Greenhouse Temperature Reading		
	24/11/2024	25/11/2024	26/11/2024
13.00.00	31,80	40,73	33,02
13.30.00	31,43	37,42	32,17
14.00.00	31,05	33,44	31,77
14.30.00	30,69	31,39	31,61
15.00.00	30,27	28,14	31,85
15.30.00	29,85	27,94	31,33
16.00.00	29,50	27,81	30,91
16.30.00	29,25	27,73	30,23
17.00.00	29,03	27,86	30,23
17.30.00	28,70	27,88	30,28
18.00.00	28,58	27,87	30,15
18.30.00	28,57	27,87	30,04
19.00.00	28,51	27,86	29,98
19.30.00	28,47	27,77	29,86
20.00.00	28,47	27,64	29,83
20.30.00	28,43	27,53	29,62
21.00.00	28,59	27,84	29,60
21.30.00	28,14	27,64	29,33
22.00.00	28,03	27,52	29,31
22.30.00	27,98	27,46	29,17
23.00.00	27,80	27,53	27,95
23.30.00	27,77	27,59	28,41

TABLE IV. GREENHOUSE HUMIDITY READING

Time	Greenhouse Humidity Reading		
	24/11/2024	25/11/2024	26/11/2024
00.00.00	0,00	92,50	93,29
00.30.00	0,00	92,65	93,22
01.00.00	0,00	93,08	93,48
01.30.00	0,00	93,07	93,74
02.00.00	0,00	92,31	93,94
02.30.00	0,00	92,49	94,01
03.00.00	0,00	93,10	94,22
03.30.00	0,00	93,32	94,36
04.00.00	0,00	93,47	94,39
04.30.00	0,00	93,03	94,46
05.00.00	89,34	93,56	94,70
05.30.00	88,70	93,63	94,66
06.00.00	86,77	92,70	94,98

Time	Greenhouse Humidity Reading		
	24/11/2024	25/11/2024	26/11/2024
06.30.00	83,89	90,79	94,54
07.00.00	85,53	89,49	93,20
07.30.00	78,90	80,98	91,79
08.00.00	83,22	79,73	88,55
08.30.00	80,23	79,61	87,52
09.00.00	80,38	83,45	86,16
09.30.00	81,83	80,74	90,16
10.00.00	82,76	80,85	90,66
10.30.00	81,72	81,69	84,65
11.00.00	80,01	81,51	84,78
11.30.00	79,34	81,65	75,48
12.00.00	84,86	81,81	74,44
12.30.00	89,58	79,64	75,00
13.00.00	85,19	80,22	76,43
13.30.00	85,98	84,88	81,14
14.00.00	85,67	82,19	81,96
14.30.00	85,76	88,59	81,40
15.00.00	87,43	90,87	81,12
15.30.00	86,14	92,60	83,52
16.00.00	88,45	93,07	84,71
16.30.00	89,20	92,78	84,17
17.00.00	88,73	93,17	83,87
17.30.00	88,53	93,37	85,17
18.00.00	88,33	93,47	86,37
18.30.00	88,37	93,17	87,69
19.00.00	88,48	93,30	88,30
19.30.00	88,73	93,45	89,17
20.00.00	88,61	93,77	89,79
20.30.00	88,95	94,03	90,54
21.00.00	88,27	93,34	90,63
21.30.00	89,66	93,70	91,60
22.00.00	90,00	94,09	91,71
22.30.00	89,91	94,20	92,00
23.00.00	90,31	94,00	92,00
23.30.00	91,15	93,56	97,38

TABLE V. SOIL MOISTURE READING

Time	Soil Moisture Reading		
	24/11/2024	25/11/2024	26/11/2024
00.00.00	0,00	81,57	80,17
00.30.00	0,00	81,53	80,22
01.00.00	0,00	81,81	80,55
01.30.00	0,00	81,31	80,53
02.00.00	0,00	81,45	79,97
02.30.00	0,00	81,65	81,00
03.00.00	0,00	81,61	80,56
03.30.00	0,00	81,48	80,09
04.00.00	0,00	81,15	80,25
04.30.00	0,00	80,35	80,75
05.00.00	78,67	79,71	80,61
05.30.00	80,69	80,09	80,53
06.00.00	84,00	78,14	80,41
06.30.00	84,55	80,36	80,39
07.00.00	83,55	80,30	80,56
07.30.00	80,73	80,93	80,56
08.00.00	75,55	81,81	79,28
08.30.00	81,10	80,28	80,76
09.00.00	81,13	75,42	81,73
09.30.00	79,17	80,17	81,06
10.00.00	81,14	81,47	81,01
10.30.00	80,43	75,52	80,13
11.00.00	78,94	79,40	80,41
11.30.00	82,82	80,16	79,67
12.00.00	89,72	80,80	80,47
12.30.00	82,90	75,03	81,17
13.00.00	86,43	76,46	81,34
13.30.00	83,82	83,56	81,16
14.00.00	79,14	78,34	80,82
14.30.00	79,70	84,18	81,09
15.00.00	81,47	80,24	80,97
15.30.00	78,70	81,81	80,97
16.00.00	74,66	83,84	80,97
16.30.00	80,48	80,77	80,99
17.00.00	81,07	81,17	80,91
17.30.00	81,30	80,84	80,78
18.00.00	81,46	80,66	80,78
18.30.00	80,31	80,92	80,72
19.00.00	80,15	82,16	80,77
19.30.00	70,65	81,56	80,80
20.00.00	75,96	81,88	80,76
20.30.00	80,89	82,39	80,78
21.00.00	78,39	82,08	80,85
21.30.00	80,58	81,76	81,01
22.00.00	80,23	79,20	80,96
22.30.00	80,62	82,14	80,95
23.00.00	80,87	81,69	76,41
23.30.00	81,08	81,25	63,91

TABLE VI. ACTUATOR ON AND OFF TIME DATA DATE 24 OCTOBER 2024

Time	Active Actuator				
	Roof	Lamp	Fan	Sprayer	Water Pump
05.00.00	Manual	On	Off	Off	On
05.30.00	Open				Off
06.00.00					
06.30.00			On	On	
07.00.00		Off	Off	Off	
07.30.00			On	On	
08.00.00	Close				On
08.30.00					Off
09.00.00					
09.30.00	Open				On
10.00.00					Off
10.30.00	Close				
11.00.00					On
11.30.00					Off
12.00.00	Open				
12.30.00			Off	Off	
13.00.00					
13.30.00					
14.00.00					On
14.30.00					
15.00.00					Off
15.30.00		On			On
16.00.00					
16.30.00					Off
17.00.00					
17.30.00					

Time	Active Actuator				
	Roof	Lamp	Fan	Sprayer	Water Pump
18.00.00					
18.30.00					
19.00.00					
19.30.00					On
20.00.00					
20.30.00					Off
21.00.00	Manual				On
21.30.00					
22.00.00					
22.30.00					Off
23.00.00					
23.30.00					

TABLE VII. ACTUATOR ON AND OFF TIME DATA DATE 25 OCTOBER 2024

Time	Active Actuator				
	Roof	Lamp	Fan	Sprayer	Water Pump
00.00.00	Manual	On	Off	Off	
00.30.00					
01.00.00					
01.30.00					
02.00.00					Off
02.30.00					
03.00.00					
03.30.00					
04.00.00					
04.30.00					
05.00.00					On
05.30.00					Off
06.00.00					On
06.30.00					Off
07.00.00	Open		On	On	
07.30.00					
08.00.00		Off			
08.30.00	Close				
09.00.00					On
09.30.00					Off
10.00.00					
10.30.00					On

Time	Active Actuator				
	Roof	Lamp	Fan	Sprayer	Water Pump
11.00.00					
11.30.00					Off
12.00.00	Open				
12.30.00	Close				On
13.00.00					
13.30.00	Open	On	Off	Off	Off
14.00.00					On
14.30.00					Off
15.00.00					
15.30.00					
16.00.00					
16.30.00					
17.00.00					
17.30.00					
18.00.00					
18.30.00					
19.00.00					
19.30.00					
20.00.00					
20.30.00					
21.00.00					
21.30.00					
22.00.00					On
22.30.00					Off
23.00.00					
23.30.00					

TABLE VIII. ACTUATOR ON AND OFF TIME DATA DATE 26 OCTOBER 2024

Time	Active Actuator				
	Roof	Lampu	Fan	Sprayer	Water Pump
00.00.00	Open	On	Off	Off	Off
00.30.00					
01.00.00					
01.30.00					
02.00.00					On
02.30.00					Off
03.00.00					
03.30.00					

Time	Active Actuator				
	Roof	Lampu	Fan	Sprayer	Water Pump
04.00.00					
04.30.00					
05.00.00					
05.30.00					
06.00.00					
06.30.00					
07.00.00					
07.30.00					
08.00.00					On
08.30.00		Off			
09.00.00					
09.30.00					Off
10.00.00					
10.30.00			On	On	
11.00.00					
11.30.00					On
12.00.00					
12.30.00					
13.00.00					
13.30.00					
14.00.00					
14.30.00					
15.00.00					
15.30.00					
16.00.00					
16.30.00					
17.00.00					Off
17.30.00			Off	Off	
18.00.00					
18.30.00					
19.00.00		On			
19.30.00					
20.00.00					
20.30.00					
21.00.00					
21.30.00					
22.00.00					
22.30.00					
23.00.00					On
23.30.00					

The data collection process is carried out 24 hours, at night the greenhouse building will be moved to a shady and safe place. This aims to secure the tool from potential theft by others. Another goal is to avoid potential rain at night that escapes the supervision of researchers. Researchers realize that this tool is still not resistant to heavy rain. The researcher understands that the design of the tool in this study is still in the prototype stage, so it still needs to be refined by further researchers or in further research. Especially in the control and monitoring variables at the greenhouse temperature, in the greenhouse temperature sensor reading data contained in Table III the greenhouse temperature touches a value of 41 C, this has the potential to make the green mustard plants wilt and can even cause the green mustard plants to die. The researcher's suggestion for further researchers is the use of a greenhouse fogger actuator that has more qualified specifications so that it is hoped that the use of the fogger actuator can increase greenhouse humidity and will reduce the value of the greenhouse temperature.

IV. Conclusion

This study designs a prototype greenhouse controller and monitors for green mustard plants based on IoT with the main control system being ESP32. This controller and monitor system is integrated with the Blynk IoT application so that greenhouse owners can control and monitor anywhere and anytime. This system can be run in automatic mode so that it is very helpful in smart agricultural automation efforts. The results of the calibration values of the 4 sensor variables are very satisfactory and can be considered as valid tools. The R_square values of the 4 sensor calibrations, namely the BH1750 light sensor, the DHT22_1 temperature sensor, the DHT22_2 temperature sensor, the DHT22_1 humidity sensor, the DHT22_2 humidity sensor, and the soil moisture sensor are respectively 0.9936; 0.9689; 0.9665; 0.9412; 0.9451; 0.9574. Based on the interpretation of the simple linear regression coefficient according to Sugiyono (2020), this tool can be categorized as a valid tool.

Based on the research results, suggestions for further research are: a. To overcome the problem of noise or ripple from DC voltage or voltage drop on the sensor that can cause the sensor to experience an error, the next researcher needs to add a voltage filter circuit to the signal input pin of each sensor. b. During the data collection process, the placement of the tool box needs to be placed in a position that is free from water or high humidity potential that can damage the hardware components in the tool box.

References

[1] Alifah, S., Nurfida, & Hermawan, A. (2019). Pengolahan Sawi Hijau Menjadi Mie Hijau Yang Memiliki Nilai Ekonomis Tinggi Di Desa Sukamanis Kecamatan Kadudampit Kabupaten Sukabumi. Journal of Empowerment Community (JEC), 1(2), 52-58. https://doi.org/10.36423/jec.v1i2.364

[2] Widodo, Y. B., Sibuea, S., Sutabri, T., Aziz, I. (2022).

Rancang Bangun Smart Greenhouse Berbasis Raspberry Pi dengan Web Framework Flask untuk Pertanian Perkotaan. Jurnal Teknologi Informatika & Komputer, 8(2), 237-250. https://doi.org/10.37012/jtik.v8i2.1247

[3] Nusantara, E. V., Ardiansah, I., & Bafdal, N. (2021). Desain Sistem Otomatisasi Pengendalian Suhu Rumah Kaca Berbasis Web Pada Budidaya Tanaman Tomat. Jurnal Keteknikan Pertanian Tropis Dan Biosistem, 9(1), 34-42. https://doi.org/10.21776/ub.jkptb.2021.009.01.05

[4] Ristian, U., Ruslianto, I., Hasfani, H., & Sari, K. (2023). Perancangan Arsitektur Node Nirkabel dalam Efisiensi Bandwidth Smart Greenhouse Berbasis Protokol MQTT. Jurnal Edukasi Dan Penelitian Informatika (JEPIN), 9(2), 218. https://doi.org/10.26418/jp.v9i2.63885

[5] Mukaromah, H., Ikhsanudin, A., Wang, L., Ningsiah, & Lestari, S. (2023). PENERAPAN SMART FARMING UNTUK BUDIDAYA CABAI DALAM GREENHOUSE. Aisyah Journal of Informatics and Electrical Engineering, 5(2), 207-217. https://doi.org/10.30604/jti.v5i2.227

[6] Setiawan, D. M., Zaman, S., & Kartika, J. G. (2023). Manajemen Pemanenan Terong (Solanum melongena L.) di Rumah Kaca Kebun Steenbergen, Belanda. Buletin Agrohorti, 11(2), 240-248. https://doi.org/10.29244/agrob.v11i2.46844

[7] Sarker, I. H., Khan, A. I., Abushark, Y. B., Alsolami, F. (2022). IoT Security Intelligence: A Comprehensive Overview, Machine Learning Solutions and Research Directions. Mobile Networks and Applications, 28(1), 296-312. https://doi.org/10.1007/s11036-022-01937-3

[8] Pereira, G. P., Chaari, M. Z., & Daroge, F. (2023). IoT-Enabled Smart Drip Irrigation System Using ESP32. IoT, 9(1), 221-243. https://doi.org/10.3390/iot4030012

[9] Inayah, I., Hayati, N., Nurcholis, A., Dimyati, A., & Prasetia, M. G. (2023). Realtime Monitoring System of Solar Panel Performance Based on Internet of Things Using Blynk Application. Elinvo (Electronics Informatics and Vocational Education), 7(2), 135-143. https://doi.org/10.21831/elinvo.v7i2.53365

[10] Kurniawan, G. W., Agung, I. G. A. P. R., & Rahardjo, P. (2023). Rancang Bangun Sistem Pemantauan Panel Surya Berbasis Internet of Things. Majalah Ilmiah Teknologi Elektro, 22(1). https://doi.org/10.24843/mite.2023.v22i01.p17

[11] Syahrindra, R., Soelistianto, F. A., Soelistianto, F. A., Mas'udia, P. E., & Mas'udia, P. E. (2020). Telemonitoring dan Pengendalian Suhu Kadar Kelembapan serta Intensitas Cahaya Matahari di dalam Ruangan Semi Indoor Menggunakan Sistem Wireless Sensor. Jurnal Jartel: Jurnal Jaringan Telekomunikasi, 10(4), 168-172. https://doi.org/10.33795/jartel.v10i4.11

[12] Rosmiati, R., Nirsal, N., & Renaldi, A. (2021). PROTOTYPE KIPAS ANGIN OTOMATIS MENGGUNAKAN SENSOR SUHU DHT22, ULTRASONIK HC-SR04, DAN BLUETOOTH HC-05 BERBASIS MIKROKONTROLER. D'computare: Jurnal Ilmiah Teknologi Informasi Dan Ilmu Komputer, 11(2), 50-56. https://doi.org/10.30605/dcomputare.v11i2.20

[13] Pratama, Y. A., Ardita, M., & Widodo, K. A. (2022). Perancangan Sistem Komunikasi Lora untuk Deteksi Dini Tanah Longsor. Prosiding SENIATI, 6(3), 699-705. https://doi.org/10.36040/seniati.v6i3.5004

[14] Hermansyah, D., Patiung, M., & Wisnujati, N. S. (2021). Analisis Trend dan Prediksi Produksi dan Konsumsi Komoditas Sayuran Sawi (Brassica Juncea L) di Indonesia Tahun 2020 s/d 2029. Jurnal Ilmiah Sosio Agribis, 21(2). https://doi.org/10.30742/jisa21220211383

[15] Telaumbanua, M., Purwantana, B., & Sutiarso, L. (2014). Rancang bangun Aktuator Pengendali Iklim Mikro di dalam Greenhouse untuk Pertumbuhan Tanaman Sawi (Brassica Rapa Var.parachinensis L.). Agritech, 34. https://doi.org/10.22146/agritech.9512.

ISSN 2355-3286

Design of a Nutrient and Environment Monitoring IoT Device in Vertical Hydroponic System

Ricardo Linelson[1], Fahmy Rinanda Saputri[2]
[1,2] Department of Engineering Physics, Universitas Multimedia Nusantara, Tangerang, Indonesia
[1]ricardo.linelson@student.umn.ac.id
[2]fahmy.rinanda@umn.ac.id

Accepted April 19, 2025
Approved June 16, 2025

***Abstract*—This study presents the design and performance evaluation of an Internet of Things (IoT)-based nutrient and environmental monitoring device for vertical hydroponic farming. The system employs multiple sensors to measure pH, Total Dissolved Solids (TDS), nutrient temperature, air temperature, and air humidity. Data is transmitted via ESP32 and integrated with the Arduino IoT Cloud, enabling real-time monitoring through a web dashboard and IoT Remote mobile application. A 10-day testing period was conducted to compare sensor outputs against standard calibrator references. The device demonstrated minimal bias (e.g., 0.20 for pH, 0.51 °C for nutrient temperature) and high precision (100.00%) across all parameters. Accuracy ranged from 92.33% (TDS) to 98.24% (nutrient temperature), while error rates were relatively low (e.g., 1.76% for nutrient temperature and 7.67% for TDS). These findings validate the system's reliability and consistency, supporting its potential for scalable implementation in precision-controlled, real-time monitoring applications within urban agriculture.**

***Index Terms*—IoT Device; Vertical Hydroponic; Nutrient and Environment Monitoring; Sensors; Real-time Data; Arduino IoT Cloud.**

I. INTRODUCTION

The global agricultural sector is currently transforming due to increasing pressures from urbanisation, climate change, and the rising demand for sustainable food systems. Conventional soil-based agriculture has become less viable in urban environments, prompting the adoption of alternative cultivation methods such as hydroponics. This soil-less system offers efficient utilisation of water and nutrients while enabling year-round crop production, making it highly suitable for urban farming initiatives [1].

However, the success of hydroponic systems largely depends on the accurate and continuous monitoring of nutrient concentrations and environmental parameters. Manual monitoring techniques or the use of standalone sensors are still commonly applied, despite their limitations in providing real-time integration, automated feedback, and system responsiveness [2]. This is particularly problematic in vertically integrated and large-scale hydroponic setups, where system precision monitoring farming is essential to maintain optimal plant growth.

According to the Food and Agriculture Organization (FAO), conventional agriculture achieves less than 50% water use efficiency, whereas vertical hydroponic systems can exceed 95%, but only under precise microclimatic and nutrient control [3], [4]. Furthermore, reports from Indonesia's National Research and Innovation Agency (BRIN) highlight the rapid growth of urban farming initiatives, although technical challenges in monitoring and system responsiveness continue to hinder their sustainability [5], [6]. These trends emphasise the urgent need for integrated, cost-effective, and adaptive systems for smart nutrient and environmental management in hydroponic agriculture [5].

Several recent studies have attempted to address this need. Sulaiman et al. (2025) conducted a review on pH and EC control systems in hydroponics, stressing the necessity for real-time nutrient monitoring using cloud platforms [1]. Rofiansyah et al. (2025) developed an image-based IoT hydroponic system, but the absence of sensor validation through statistical analysis limited its accuracy [2]. Simanungkalit et al. (2023) proposed an IoT-enabled vertical hydroponic system focused on hardware deployment yet lacked integration with cloud analytics and benchmarking tools [6]. Meanwhile, Oton and Iqbal (2021) implemented a low-cost SCADA solution using ESP32 and Arduino IoT Cloud, although not in the context of agriculture [7]. Noviardi (2022) introduced an Arduino IoT Cloud-based aquaponic model; however, it lacked synchronization across multiple sensor inputs and did not evaluate data accuracy against reference standards [8].

Sneineh and Shabaneh introduced an ESP32-based hydroponic IoT system, but it lacked quantitative evaluation of sensor error effects on nutrient balance and plant growth [9]. Moreover, Yuan et al. pointed out that the disconnection between sensor accuracy

metrics (e.g., RMSE, precision) and actionable agronomic decisions remains a significant research gap in smart agriculture systems [3]. This gap underscores that mere data collection is insufficient without rigorous validation and its translation into effective nutrient management strategies. In vertical hydroponics, even minor inaccuracies in nutrient monitoring can disrupt root absorption efficiency, impact photosynthetic activity and biomass accumulation [10], [11], [12], [13].

Advancements in Internet of Things (IoT) technologies have enabled smart farming solutions that combine microcontrollers, digital sensors, and cloud-based platforms for remote and real-time monitoring [7], [14], [15]. Among available platforms, Arduino IoT Cloud stands out for its secure ESP32 compatibility, built-in sketch programming, and support for real-time mobile access via IoT Remote [7]. It offers advantages over other platforms, such as Blynk [16], ThingSpeak [17], Antares [18] or MongoDB [19], which often require complex backend configuration or lack native mobile applications.

While previous studies have demonstrated the efficacy of IoT frameworks in hydroponic and soil-based cultivation, few have addressed the real-time synchronisation between nutrient dynamics (pH and TDS) and environmental parameters (air temperature and humidity) in mist-based vertical hydroponic setups [8], [20], [21]. Although prior research shows promise, most systems are limited in scope, lacking integration between sensors, real-time synchronisation, user-accessible dashboards, and performance validation. These gaps present critical barriers to deploying scalable and responsive hydroponic monitoring systems. A recent review of smart nutrient management technologies further underscores the need for IoT systems that combine multi-sensor input, cloud automation, and adaptive control frameworks specific to crop environments [22].

This study proposes the design and implementation of an IoT-based monitoring device specifically for vertical hydroponic systems, fully integrated with the Arduino IoT Cloud platform. The proposed system incorporates TDS, pH, water temperature, air temperature, and humidity sensors, all interfaced with an ESP32 microcontroller. A real-time dashboard enables remote visualisation and control of system parameters. To ensure reliability and measurement accuracy, the sensor readings will be compared with a calibrated reference instrument to calculate bias, precision, accuracy, and error, thereby enabling comprehensive performance validation of the monitoring system [16], [22].

By filling a methodological and technological research gap, this study offers a comprehensive framework for smart vertical hydroponic monitoring that is responsive, accurate, and cost-effective. It also aligns with the objectives of the United Nations Sustainable Development Goals (SDGs), particularly in enhancing sustainable agriculture and promoting resilient infrastructure for urban food production systems [23], [24].

The paper is organised as follows: Section I introduces the research background and objectives. Section II discusses system architecture, component configuration, and calibration procedures. Section III presents experimental results and system performance evaluation. Finally, Section IV concludes with key findings and provides recommendations for future system development.

II. METHODS

This study employed a design and implementation approach that integrates hardware and software components in a real-time Internet of Things (IoT)-based monitoring system for vertical hydroponic agriculture. The methodology was structured into four main stages, as outlined below:

A. System Architecture & Design

The proposed IoT system is designed to monitor nutrients and environmental conditions in an vertical hydroponic setup. It integrates both hardware and software components to support real-time monitoring and automation. The hardware includes one microcontroller, five sensors, two actuators, and a display. The software includes a web dashboard and a mobile interface. The system interconnection is illustrated in Fig. 1.

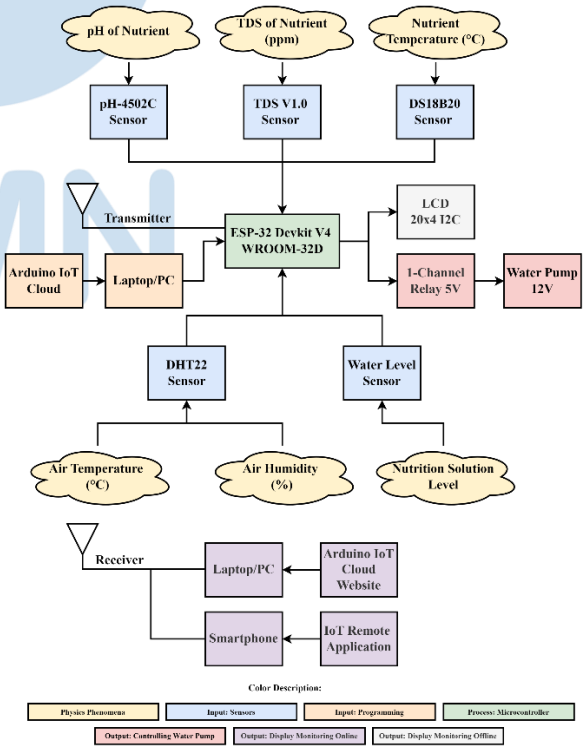


Fig. 1. Block Diagram of The IoT Device Interconnection

The system uses the ESP32 WROOM-32D as the main controller. This microcontroller receives, processes, and transmits data to the Arduino IoT Cloud

via Wi-Fi. It supports wireless data transmission, enabling cloud-based monitoring. Each sensor has a specific function. The pH-4502C sensor measures the pH of the nutrient solution, which affects nutrient uptake. The TDS V1.0 CHN sensor detects total dissolved solids, showing the strength of the nutrient concentration. These values help in adjusting the fertilizer according to plant needs.

The DS18B20 sensor measures the nutrient solution temperature. It is waterproof and has high accuracy (±0.5°C), making it ideal for liquid monitoring. Temperature affects root metabolism and oxygen solubility. The DHT22 sensor records air temperature and humidity. These factors are important for controlling transpiration and preventing plant stress. A water level sensor checks the height of the solution in the reservoir. If the level is low, the system sends alerts or activates the pump. This prevents dry-run damage and ensures continuous nutrient flow.

A 5-volt single-channel relay acts as a switch to control the 12V water pump. The pump transfers the solution from a 20-liter tank to the vertical hydroponic pipes. An LCD 20x4 display with I2C shows real-time data from the sensors in numeric and text format. On the software side, the Arduino IoT Cloud is used to upload the program and build the web dashboard. The IoT Remote app is used to build the mobile dashboard interface. All hardware components are wired into a single IoT device unit, as illustrated in Fig. 2.

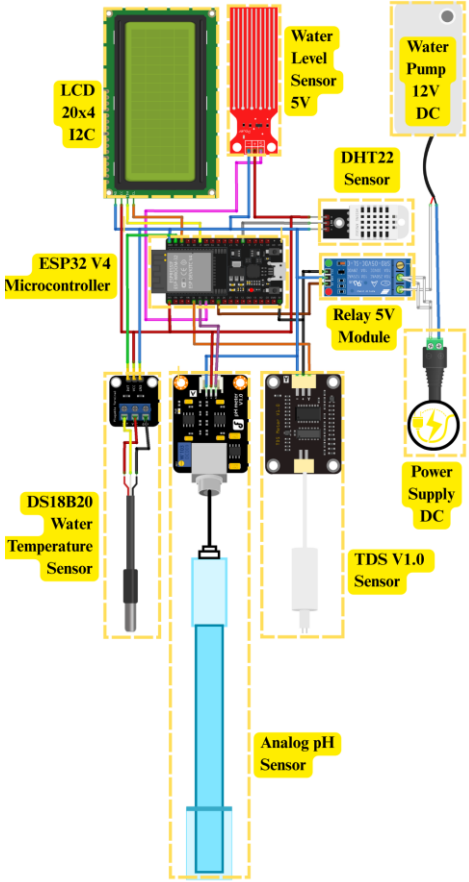


Fig. 2. Schematic Diagram of The IoT Device

Both hardware and software components form an integrated environmental and nutrient monitoring system that enables real-time, cloud-based data acquisition via Arduino IoT Cloud.

B. System Programming & Algorithm

The code for the IoT device was developed and compiled using the Sketch menu on the Arduino IoT Cloud web platform. The program initializes each sensor, acquires data, converts analogue signals into digital values, and transmits the processed results to the Arduino IoT Cloud via Wi-Fi using compatible libraries. Figure 3 illustrates the code compilation process executed through the Arduino IoT Cloud interface.

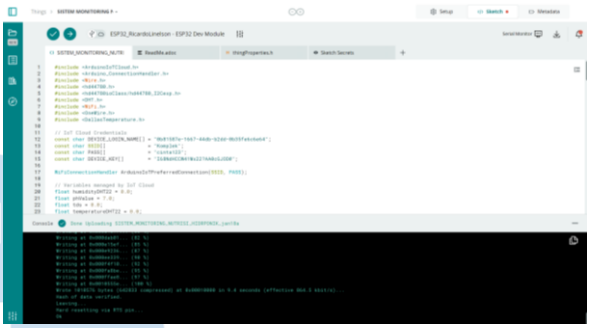


Fig. 3. Code Compilation Process

As shown in Fig. 4, the device was successfully flashed with the compiled program. This was verified through a confirmation message displayed on the serial monitor, indicating that the code upload and cloud connectivity were successful. The serial output included the following key messages as described in Fig. 4. These messages confirm that the device established a Wi-Fi connection, synchronised with the Arduino IoT Cloud, and began updating sensors data in real-time.

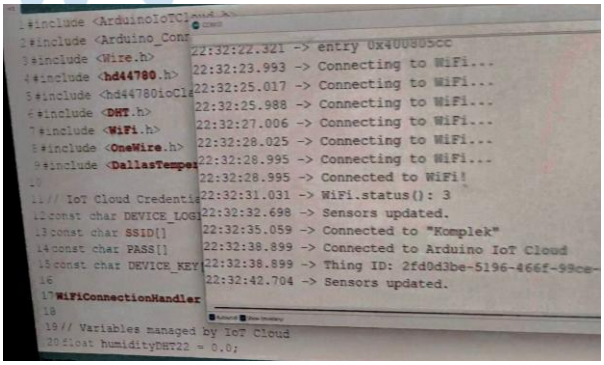


Fig. 4. Result on Serial Monitor

To illustrate the logical workflow of the system, Figure 5 presents a flowchart detailing the core algorithm implemented in the microcontroller. The system begins by initializing all sensors and proceeds to sequentially read environmental and nutrient-related parameters. Each sensor value undergoes basic

validation and calibration routines before being transmitted to the Arduino IoT Cloud for real-time monitoring and data logging.

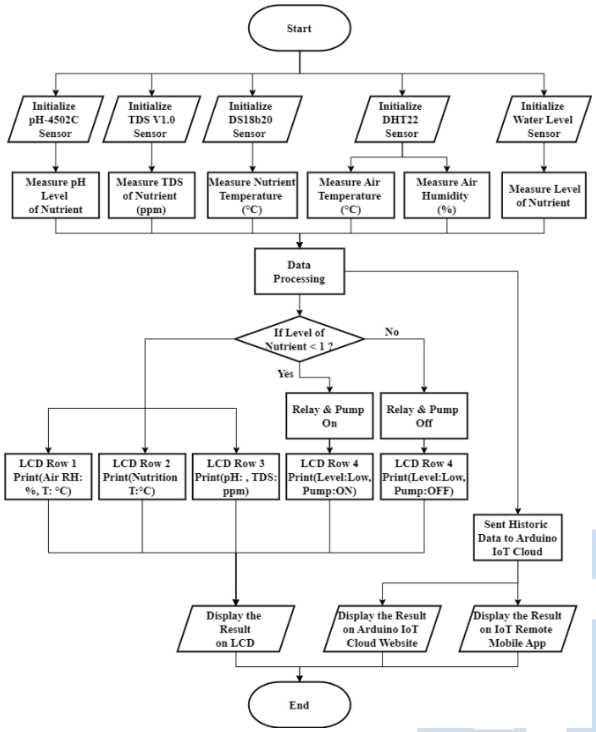


Fig. 5. Flowchart of the Code Algorithm

Conditional logic is embedded within the system to control the actuator, specifically a relay module that regulates the nutrient pump based on sensor thresholds. A binary trigger mechanism (digital 0 or 1) governs the relay's activation state. The water level sensor is strategically installed at a height of 20 cm from the base of the nutrient reservoir, corresponding to approximately 5 litres of solution. When the sensor detects the presence of nutrient solution (binary signal: 1), the system deactivates the pump to prevent overflow or unnecessary circulation. Conversely, when the sensor detects the absence of liquid (binary signal: 0), the relay is triggered to activate the pump and restore the appropriate fluid level in the system. This control logic ensures automated nutrient regulation within the vertical hydroponic environment, enhancing operational efficiency and reducing the risk of human error or pump dry-run damage.

C. *System Cloud Connectivity*

In Fig. 7, the device status history on the cloud dashboard indicates consistent uptime and communication. Cloud connectivity ensures that sensor readings are regularly updated and stored on a centralised server for visualisation and logging.

Fig. 6. Device Status History

D. *Setting Dashboard*

To visualize sensor data remotely, a dashboard was built using the Arduino IoT Cloud interface. The configured dashboard allows users to monitor environmental and nutrient parameters and also control actuators remotely. The dashboard design is mobile responsive, ensuring accessibility across different devices and user interfaces. Each monitored variable was assigned to a corresponding widget on the platform, such as real-time gauges and line charts as shown in Fig. 9.

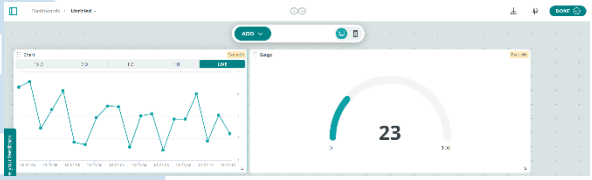


Fig. 7. Dashboard Setting

E. *Sensor Data Validation Test*

The readings from the five main sensors on the IoT device were compared against three calibrator instruments to assess measurement accuracy. The pH-4502C sensor was validated using a digital pH meter. The TDS V1.0 sensor was compared with a TDS-3 meter. The DS18B20 sensor was validated using an external thermometer probe on the HTC-2. The DHT22 sensor was used to measure air temperature and humidity, which were compared against the internal thermometer and hygrometer on the HTC-2, respectively.

The validation test was conducted over a period of 10 days. Due to the calibrator instruments lacking automatic logging features, measurements were taken manually three times per day at 8 AM, 1 PM, and 6 PM. The daily average was calculated from these three samples for both the IoT device and the calibrator tools. Consequently, 10 average data points were obtained for each sensor.

The collected data was processed and analysed using descriptive statistical methods. Sensor data were compared to reference values using standard equations to calculate precision, accuracy, bias, and error [22], [25]. as shown below:

$$Precision = 100\% \left(1 - \frac{\sigma}{\bar{x}}\right) \quad (1)$$

$$Accuracy = 100\% \left(1 - \frac{|Bias|+3\sigma}{x_{real}}\right) \quad (2)$$

$$|Bias| = |x_{real} - \bar{x}| \quad (3)$$

$$Error = 100\% \left(\frac{|Bias|+3\sigma}{x_{real}}\right) \quad (4)$$

In these equations, x̄ denotes the average value obtained from the sensor measurements. σ refers to the standard deviation indicating data dispersion. x_{real} represents the reference value obtained from the calibrator device. $|Bias|$ (3) determines the absolute value of bias between the measured value ($\bar{x}$) and actual value (x_{real}). Precision (1) reflects the consistency of repeated measurements. Accuracy (2) measures the accuracy by accounting for both bias and associated uncertainty. Error (4) calculates the relative error in percentage, comparing the deviation to the actual reference value.

III. Result and Discussion

The results of this research encompass four main areas of analysis: (1) the design outcome of the IoT-based monitoring device, (2) the visualization output of the real-time monitoring dashboard, (3) the validation results of sensor data accuracy and performance, and (4) the identified potentials for system improvements, including future integration with predictive analytics and automation frameworks.

A. IoT Device Design

The designed IoT device is positioned at the front side of the reservoir to allow users easy access to view real-time monitoring data on the LCD screen. A vertical pipe is installed above the reservoir, featuring several holes that serve as planting spots for hydroponic crops. This vertical pipe is directly connected to the reservoir, functioning to collect and return the nutrient solution that flows down from the top, driven by the water pump. The final design of the IoT device is shown in Fig. 8.

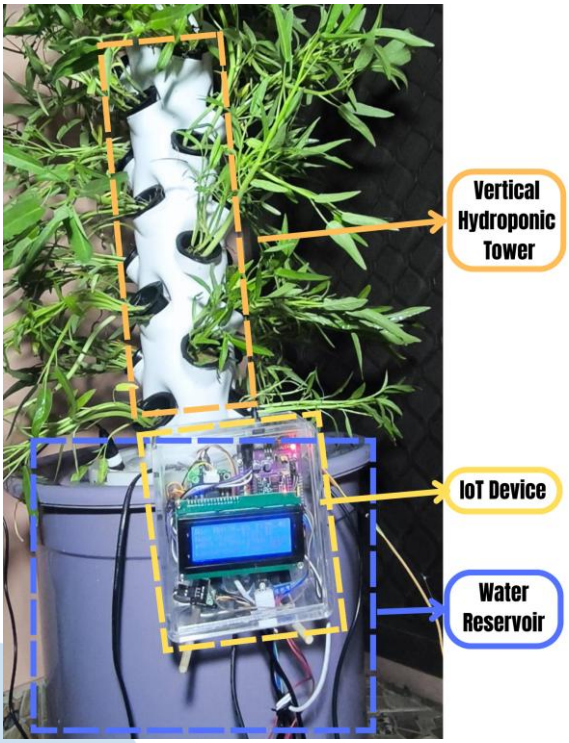


Fig. 8. IoT Device Result

B. Dashboard Monitoring Display

From a functionality standpoint, Arduino IoT Cloud offers up to 10 widgets in the free version, enabling more comprehensive monitoring of variables such as pH, TDS, nutrient temperature, air temperature, and humidity. In contrast, Blynk limits free users to only 4 widgets, without built-in data export, and historical data access is constrained by cloud storage capacity. ThingSpeak, Antares, and MongoDB, while flexible, lack native mobile applications and require separate backend integration or manual query configuration, thus increasing development complexity. Furthermore, data in Arduino IoT Cloud is stored for 1 day, but can be exported manually prior to expiration, offering a balance between free-tier limitations and usability.

The monitoring data history is presented using five gauges and five-line charts arranged on a single dashboard interface. The five monitored variables include Potential of Hydrogen (pH), Total Dissolved Solids (ppm), Nutrient Temperature (°C), Air Temperature (°C), and Air Humidity (%). The dashboard display is accessible via laptop, PC, or mobile phone through the Arduino IoT Cloud website, as shown in Fig. 9.

Fig. 9. Dashboard on Website Arduino IoT Cloud

Additionally, Arduino IoT Cloud provides a mobile application called IoT Remote, which can be downloaded from app stores. The IoT Remote app facilitates easier access to the monitoring dashboard by allowing users to open it directly from the application. The mobile dashboard display via IoT Remote is illustrated in Fig. 10.

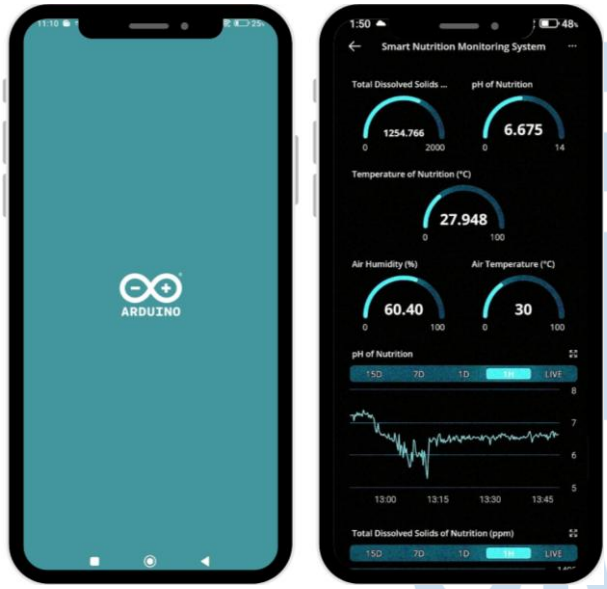


Fig. 10. Dashboard on IoT Remote Mobile App

C. *Device Sensor Reading Performance*

The detailed measurement results from each monitoring sensor were analysed to determine the percentage of precision, accuracy, and error of the IoT device readings compared to the reference values obtained from calibrator instruments. The comparative performance outcomes over a 10-day testing period are presented in Table I.

TABLE I. DEVICE SENSOR READING PERFORMANCE

Variable	Mean Actual Value	Mean Device Reading	|Bias| ()	Precision (%)	Accuracy (%)	Error (%)
Potential of Hydrogen (No unit)	6.43	6.23	0.20	100.00	96.84	3.16
Total Dissolved Solids (ppm)	1244.60	1149.43	95.17	100.00	92.33	7.67
Nutrient Temperature (°C)	29.03	28.52	0.51	100.00	98.24	1.76
Air Temperature (°C)	29.61	28.22	1.39	100.00	95.33	4.67
Air Humidity (%)	73.10	68.78	4.32	100.00	94.08	5.92

The mean device readings for each variable closely approximate the actual values, demonstrating relatively low bias, indicating that the sensors in the system provide accurate readings within acceptable margins. For instance, the bias in pH readings is only 0.20, and for nutrient temperature, it is 0.51, both of which are minimal discrepancies, ensuring the reliability of the device.

In terms of precision, all measurements show values close to 100%, with the highest being 100.60% for pH, TDS, and nutrient temperature. This indicates that the device is highly consistent in producing readings that align closely with the actual values, which is crucial for ensuring stable performance in an IoT monitoring system. However, while precision is consistently high, the accuracy and error percentages demonstrate slight deviations, especially in TDS, air temperature, and air humidity. For example, TDS has an accuracy of 92.33% and an error of 7.67%, suggesting a moderate discrepancy between the device's readings and the actual values.

The error percentage for all variables varies, with the lowest error observed in nutrient temperature (1.76%) and the highest in air humidity (5.92%). These discrepancies in error could be due to sensor calibration issues, environmental factors, or limitations in the sensor technology. While the system performs well within acceptable limits, the higher errors in variables like air temperature and humidity indicate that further calibration and fine-tuning of the sensors may be needed for more precise monitoring, especially in variable environmental conditions. Therefore, continuous calibration and periodic validation against reference instruments are recommended to improve overall device performance and reliability.

To further illustrate the measurement results of each monitoring sensor in comparison to the calibrator readings, scatter chart visualizations are provided in Fig. 11 through Fig. 15.

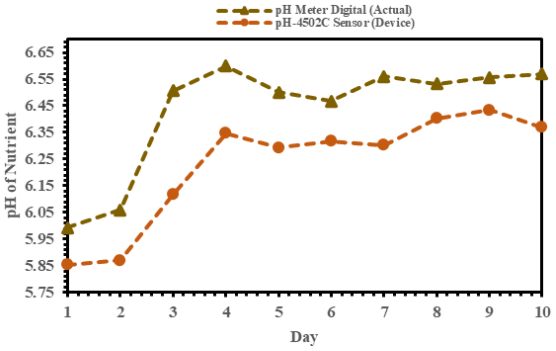


Fig. 11. Chart of pH Actual Values and Device Reading Value

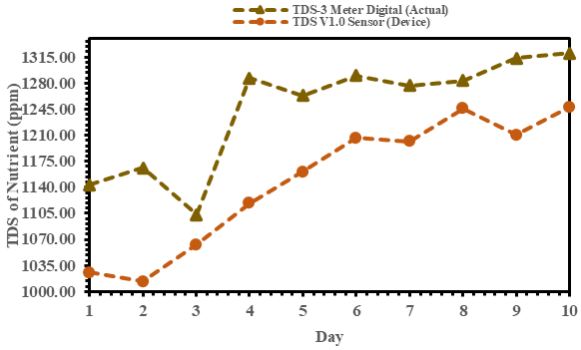


Fig. 12. Chart of TDS Actual Values and Device Reading Value

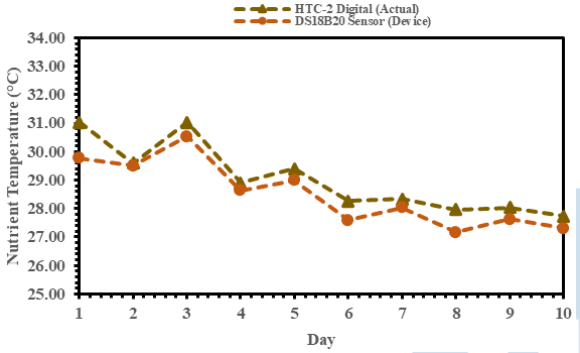


Fig. 13. Chart of Nutrient Temperature Actual Values and Device Reading Value

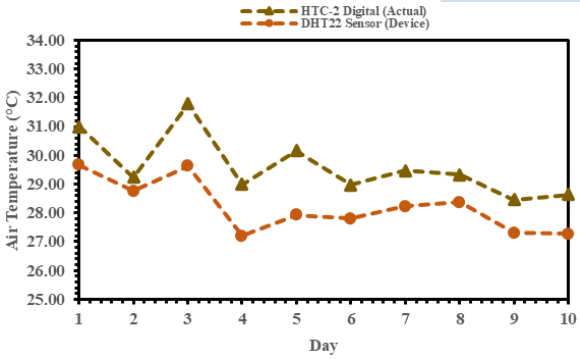


Fig. 14. Chart of Air Temperature Actual Values and Device Reading Value

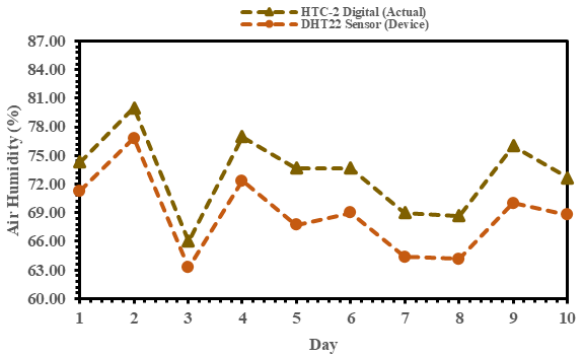


Fig. 15. Chart of Air Humidity Actual Values and Device Reading Value

D. Potentials of System Improvements

In future development stages, the integration of machine learning algorithms could enable predictive analytics for nutrient dosing and environmental adjustments. By collecting and analysing historical sensor data such as pH, TDS, nutrient temperature, air temperature, and humidity, supervised models such as random forest, support vector machines, or LSTM (Long Short-Term Memory) neural networks could be trained to forecast nutrient requirements or detect anomalies in the microclimate environment. Similar techniques have proven effective in predictive irrigation scheduling and fertigation management in precision agriculture contexts [26], [27], [28], [29], [30]. This would enable a closed-loop system with automated actuation based on real-time prediction rather than threshold-based rules.

IV. Conclusion

This research presents the design and performance evaluation of an IoT-based nutrient monitoring system for vertical hydroponic planting. The system demonstrated high reliability, with mean device readings closely aligning with actual values, indicating low bias (e.g., 0.20 for pH and 0.51 for nutrient temperature). Precision values for all variables were found to exceed 100%, signifying consistency in the system's readings. However, the accuracy results revealed slight discrepancies, particularly in Total Dissolved Solids (TDS) (92.33%) and air humidity (94.08%), with error percentages ranging from 1.76% (nutrient temperature) to 7.67% (TDS).

The real-time monitoring dashboard, accessible via PC, laptop, or mobile application, provides users with an intuitive interface for efficient data visualization and analysis. Despite some minor deviations in accuracy, the device demonstrated sufficient performance for reliable nutrient and environmental monitoring in hydroponic systems.

Future studies should focus on further refining sensor calibration and reducing error margins, particularly in variables such as air temperature and humidity. Incorporating advanced sensor technologies or machine learning algorithms could enhance the system's accuracy and adaptability, thereby improving the precision and robustness of IoT-based agricultural monitoring systems.

Acknowledgment

The authors would like to express their sincere gratitude to Universitas Multimedia Nusantara for the financial support and research facilities provided. These contributions were instrumental in the successful completion of this research.

References

[1] H. Sulaiman, A. A. Yusof, and M. K. Mohamed Nor, "Automated Hydroponic Nutrient Dosing System: A Scoping Review of pH and Electrical Conductivity Dosing

Frameworks," AgriEngineering 2025, Vol. 7, Page 43, vol. 7, no. 2, p. 43, Feb. 2025, doi: 10.3390/AGRIENGINEERING7020043.

[2] W. Rofiansyah, F. R. Zalianty, F. A. La Ito, I. Wijayanto, H. H. Ryanu, and I. D. Irawati, "IoT-based control and monitoring system for hydroponic plant growth using image processing and mobile applications," PeerJ Comput Sci, vol. 11, p. e2763, Mar. 2025, doi: 10.7717/PEERJ-CS.2763/SUPP-6.

[3] G. N. Yuan et al., "A review on urban agriculture: technology, socio-economy, and policy," Heliyon, vol. 8, no. 11, p. e11583, Nov. 2022, doi: 10.1016/J.HELIYON.2022.E11583.

[4] U. Lele and S. Goswami, "The Food and Agriculture Organization of the United Nations," in Food for All: International Organizations and the Transformation of Agriculture, 2021. doi: 10.1093/oso/9780198755173.003.0010.

[5] Rais Nurwahyudin and Bagus Setya Rintyarna, "Optimasi Waktu Pemaparan Cahaya Monokromatik terhadap Produktivitas Mikrogreens Pakcoy melalui Sistem Internet of Things," Prosiding Seminar Nasional Pembangunan dan Pendidikan Vokasi Pertanian, vol. 4, no. 1, 2023, doi: 10.47687/snppvp.v4i1.684.

[6] E. Simanungkalit, M. Husna, and J. S. Tarigan, "Smart Farming On IoT-Based Aeroponik Systems," Sinkron, vol. 8, no. 1, 2023, doi: 10.33395/sinkron.v8i1.11988.

[7] C. N. Oton and M. T. Iqbal, "Low-Cost Open Source IoT-Based SCADA System for a BTS Site Using ESP32 and Arduino IoT Cloud," in 2021 IEEE 12th Annual Ubiquitous Computing, Electronics and Mobile Communication Conference, UEMCON 2021, 2021. doi: 10.1109/UEMCON53757.2021.9666691.

[8] Noviardi, "DESAIN ARSITEKTUR MODEL PADA SMART AQUAPONIC BERBASIS ARDUINO IoT CLOUD," Jurnal SIMTIKA, vol. 5, no. 2, 2022.

[9] A. Abu Sneineh and A. A. A. Shabaneh, "Design of a Smart Hydroponics Monitoring System Using an ESP32 Microcontroller and the Internet of Things," MethodsX, vol. 11, 2023, doi: 10.1016/j.mex.2023.102401.

[10] A. A. Alexopoulos, E. Marandos, A. Assimakopoulou, N. Vidalis, S. A. Petropoulos, and I. C. Karapanos, "Effect of nutrient solution ph on the growth, yield and quality of taraxacum officinale and reichardia picroides in a floating hydroponic system," Agronomy, vol. 11, no. 6, 2021, doi: 10.3390/agronomy11061118.

[11] K. Singh, V. Guleria, S. Kaushal, and Shubham, "Utilization of Biofertilizers and Plant Growth Promoters in Hydroponic Production System," Current Journal of Applied Science and Technology, vol. 42, no. 37, 2023, doi: 10.9734/cjast/2023/v42i374243.

[12] Y. Park and K. A. Williams, "Organic hydroponics: A review," 2024. doi: 10.1016/j.scienta.2023.112604.

[13] M. Krastanova, I. Sirakov, S. Ivanova-Kirilova, D. Yarkov, and P. Orozova, "Aquaponic systems: biological and technological parameters," 2022. doi: 10.1080/13102818.2022.2074892.

[14] Prof. N. A. Dawande, R. Morye, D. Sarode, and N. Siddiqui, "IOT Based Home Automation System over Cloud," Int J Res Appl Sci Eng Technol, vol. 11, no. 5, 2023, doi: 10.22214/ijraset.2023.53357.

[15] Z. Akbar, S. Wetenriajeng Sidehabi, and T. Aulani, "Temperature and Humidity Monitoring System on Bread Poofer Using Arduino IoT Cloud," JEAT : Journal of Electrical and Automation Technology, vol. 2, no. 1, 2023, doi: 10.61844/jeat.v2i1.510.

[16] F. R. Saputri, R. Linelson, M. Salehuddin, D. M. Nor, and M. I. Ahmad, "Design and development of an irrigation monitoring and control system based on blynk internet of things and thingspeak," PLoS One, vol. 20, no. 4, p. e0321250, Apr. 2025, doi: 10.1371/JOURNAL.PONE.0321250.

[17] K. E. Lakshmiprabha and C. Govindaraju, "Hydroponic-based smart irrigation system using Internet of Things," International Journal of Communication Systems, vol. 36, no. 12, 2023, doi: 10.1002/dac.4071.

[18] D. Perdana, K. Ramadhani, and I. Alinursafa, "Analysis Of The MQTT Protocol On Hydroponic System Based On Internet Of Things And Antares Platform," Webology, vol. 19, no. 2, 2022.

[19] Y. M. Wu et al., "IoT-interfaced solid-contact ion-selective electrodes for cyber-monitoring of element- specific nutrient information in hydroponics," Comput Electron Agric, vol. 214, 2023, doi: 10.1016/j.compag.2023.108266.

[20] R. Yohan Husnira and R. Rivaldi, "Pendeteksi Kadar Air Pada Tanah Dalam Pot Untuk Mengeluarkan Peringatan Menggunakan Arduino IoT Cloud," Journal of Computer Science and Informatics Engineering (CoSIE), 2023, doi: 10.55537/cosie.v2i2.589.

[21] L. Hartawan et al., "Penyiraman Tanaman Otomatis Berbasis Arduino IoT Cloud di Lahan Pertanian," Jurnal Pengabdian Kepada Masyarakat, vol. 2, no. 1, 2023.

[22] R. Linelson, F. R. Saputri, S. D. Wijaya, V. R. Lee, and M. D. Al-Haidar, "A Security Radar System on a Semi-Autonomous Car based on the Ultrasonic Sensor, Servo Motor and Arduino Uno," in Proceedings of the 3rd 2023 International Conference on Smart Cities, Automation and Intelligent Computing Systems, ICON-SONICS 2023, 2023. doi: 10.1109/ICON-SONICS59898.2023.10435222.

[23] R. Sampedro, "The Sustainable Development Goals (SDG)," Carreteras, vol. 4, no. 232, 2021, doi: 10.1201/9781003080220-8.

[24] N. Zhu, Q. Ma, J. Ai, Z. Zeng, and C. Zhou, "Potential land-use function conflicts of cultivated land for urban sustainable development: a case study in Yancheng City, China," Front Sustain Food Syst, vol. 7, 2023, doi: 10.3389/fsufs.2023.1274980.

[25] V. Lee and F. R. Saputri, "Website-Based Lighting Monitoring System Design in a Laboratory of Universitas Multimedia Nusantara," in Proceedings of 2nd 2021 International Conference on Smart Cities, Automation and Intelligent Computing Systems, ICON-SONICS 2021, 2021. doi: 10.1109/ICON-SONICS53103.2021.9617167.

[26] P. Catota-Ocapana, C. Minaya-Andino, P. Astudillo, and D. Pichoasamin, "Smart control models used for nutrient management in hydroponic crops: A systematic review," IEEE Access, 2025, doi: 10.1109/ACCESS.2025.3526171.

[27] H. Sulaiman, A. A. Yusof, and M. K. Mohamed Nor, "Automated Hydroponic Nutrient Dosing System: A Scoping Review of pH and Electrical Conductivity Dosing Frameworks," AgriEngineering 2025, Vol. 7, Page 43, vol. 7, no. 2, p. 43, Feb. 2025, doi: 10.3390/AGRIENGINEERING7020043.

[28] K. H. Mohd Azmi, N. A. Mohamed Radzi, and A. Ahmad, "The Future Of Sustainable Agriculture: A Review Of Iot And Autonomous Control In Vertical Hydroponic Farming," Advances in Electrical and Electronic Engineering, vol. 22, no. 2, pp. 146–162, Jun. 2024, doi: 10.15598/AEEE.V22I2.5321.

[29] H. Chowdhury, D. B. P. Argha, and M. A. Ahmed, "Artificial Intelligence in Sustainable Vertical Farming," 2023.

[30] S. V. S. Ramakrishnam Raju, B. Dappuri, P. Ravi Kiran Varma, M. Yachamaneni, D. M. G. Verghese, and M. K. Mishra, "Design and Implementation of Smart Hydroponics Farming Using IoT-Based AI Controller with Mobile Application System," J Nanomater, vol. 2022, 2022, doi: 10.1155/2022/4435591.

ISSN 2355-3286

An Adaptive Stacking Approach for Monthly Rainfall Prediction with Hybrid Feature Selection

Ahmad Zulfa[1,2], Ahmad Saikhu [1], Muhamad Hilmil Muchtar Aditya Pradana[1], Irvan Budiawan[3,4]
[1] Department of Informatics, Institut Teknologi Sepuluh November, Surabaya, Indonesia
[2] The Agency for Meteorology, Climatology and Geophysics (Badan Meteorologi, Klimatologi, dan Geofisika, BMKG), Indonesia
[3] Department of Electrical Engineering, Faculty of Engineering, Universitas Jenderal Achmad Yani, Cimahi, Indonesia
[4]Center for Instrumentation Technology and Automation, Institut Teknologi Bandung, Bandung, Indonesia
[1,2]ahmad.zulfa@bmkg.go.id, [1]saikhu@its.ac.id, [1]hilmi@its.ac.id, [3,4]budiawan.irvan@gmail.com

Accepted April 30, 2025
Approved June 24, 2025

***Abstract*— Accurate rainfall prediction is critical for effective water resource management, agriculture, and climate risk mitigation. However, the inherent non-linearity and variability of rainfall patterns present significant modeling challenges. This study proposes an Adaptive Stacking Ensemble framework for monthly rainfall prediction, enhanced by a Hybrid Feature Selection strategy. The feature selection integrates three techniques—Correlation Analysis, Feature Importance (Random Forest), and Recursive Feature Elimination (RFE)—using a voting mechanism to ensure robust and consistent feature selection. The ensemble framework employs a diverse set of machine learning algorithms, including Random Forest, K-Nearest Neighbors, XGBoost, AdaBoost, Decision Tree, and Linear Regression, as base learners. Meta-learners are selected adaptively based on empirical performance, with the three top-performing models—Linear Regression, AdaBoost, and XGBoost—evaluated individually and collectively through a voting-based stacking approach. This flexible strategy ensures the model captures both linear and nonlinear dependencies in the data. Experimental results show that while standalone Linear Regression achieved the highest individual accuracy (R^2 = 0.931), the best ensemble performance was attained using the voting-based stacking model, which combined the top meta-learners and achieved an R^2 of 0.917, SMAPE of 13.33%, MAE of 0.287, and RMSE of 0.339. These findings confirm the effectiveness of adaptively integrating multiple strong learners in enhancing model generalization and prediction reliability for climatological applications.**

Index Terms—Elimination; Feature; Importance; Learning; Rainfall; Stacking.

I. INTRODUCTION

The Riau Islands Province is a maritime region with a coastline stretching 2,367.6 km and a total area of 251,810 km², of which only 4% consists of land, while the remaining 96% is made up of water [1]. The climate type of Tanjungpinang City is classified as an equatorial tropical climate with relatively high rainfall throughout the year. During the period from 1991 to 2024, the maximum recorded rainfall occurred in January 2021, reaching 926.9 mm over an average of 22 rainy days in that month. Despite the significant rainfall, there is no distinct separation between the rainy and dry seasons. Tanjungpinang City is categorized as a Non-Seasonal Zone (Non-ZOM), meaning it does not have a clear seasonal boundary between wet and dry periods, and its climatic analysis does not follow the strict monsoon patterns typically observed in other regions of Indonesia.

Changes in the global climate system have a significant influence on local weather patterns and climate variability. These impacts can be observed through shifts in rainfall distribution, temperature extremes, and the increasing frequency of extreme weather events in various regions [2].

Weather and climate have fundamental differences. Weather refers to various processes occurring in the atmosphere at a specific time and location, reflecting the immediate state of the atmosphere and its short-term changes within a particular area [3].

The characteristics of rainfall in a region are important to understand in order to determine water availability and to identify potential issues and disasters related to water resources. Information about rainfall characteristics, including the identification of wet months, moist months, and dry months, is highly valuable for regional management. Through this understanding, the utilization of rainfall can be optimized while minimizing any potential negative impacts that may arise [4].

The unit used for measuring rainfall parameters is millimeters. According to the Meteorology, Climatology, and Geophysics Agency of Indonesia (Badan Meteorologi, Klimatologi, dan Geofisika — BMKG), rainfall can be understood as measuring the height of accumulated rainwater. If the rainfall is recorded as 1 millimeter, it means that on a surface area of one square meter, the water would reach a height roughly equivalent to the thickness of a fingernail, or about the same as a medium-sized bottled water (approximately one liter). In other words, if rainfall is measured at a location with a water depth of 1 millimeter, and the water is evenly distributed over a flat surface without evaporation or runoff, it would represent that amount. Runoff itself refers to the water that flows over the ground surface as a result of rainfall or other water sources that do not infiltrate into the soil.

Rainfall classification in this study refers to the standards set by the Meteorology, Climatology, and Geophysics Agency (BMKG), which categorizes rainfall intensity as follows: (1) cloudy if rainfall is 0 mm/day; (2) light rain between 0.5–20 mm/day; (3) moderate rain between 20–50 mm/day; (4) heavy rain between 50–100 mm/day; (5) very heavy rain between 100–150 mm/day; and (6) extreme rain if rainfall exceeds 150 mm/day. However, for the purposes of this study, the classification is simplified into two main categories: rain and no rain. The "rain" category includes any rainfall greater than 0 mm/day, while the "no rain" category applies when the rainfall is exactly 0 mm/day [5].

This study aims to develop a rainfall prediction model for Tanjungpinang City using an adaptive stacking ensemble learning approach combined with feature selection through the Voting Feature Selector method, where features selected by at least 2 out of 3 methods will be included in the base model. The model will then be evaluated using R^2, MAE, and RMSE. This model is expected to be a reliable solution for providing weather prediction information, particularly the rainy season forecast, and supporting better decision-making for various sectors dependent on weather conditions in the Tanjungpinang area.

Typically, stacking ensemble learning uses a combination of static models. With the adaptive stacking ensemble learning approach, the meta-learner is automatically selected based on the best performance of the base learners or initial results. With the adaptive stacking ensemble learning approach, the model is expected to deliver the best results as it adjusts to the previous training data

II. METHODOLOGY

The dataset used in this study is secondary data obtained from the III Class Meteorological Station Raja Haji Fisabilillah Tanjungpinang with official permission from the relevant authorities. In addition, global climate index data such as the Southern Oscillation Index (SOI) and the Indian Ocean Dipole (IOD) were sourced from trusted institutions, namely the National Oceanic and Atmospheric Administration (NOAA).

The dataset consists of 408 samples (34 years × 12 months) containing various climatological variables relevant to the prediction of monthly rainy seasons. The variables in this dataset are categorized as follows: Target (Dependent Variable): Rainfall (CH, in mm) → Represents the amount of monthly rainfall (curah hujan) to be predicted.

Predictor (Independent Variables): Air Pressure (hPa): Consists of two variations: P Mean Sea level pressure correction (MSL), P0 Land surface pressure correction (Stasiun). Air Temperature (°C): Consists of six variations: T07 (Temperature at 07:00), T13 (Temperature at 13:00), T18 (Temperature at 18:00), T (Temperature average), Tx (Temperature maximum), Tn (Temperature minimum). Relative Humidity (%): Consists of four variations: RH07 (Humidity at 07:00), RH13 (Humidity at 13:00), RH18 (Humidity at 18:00), RH (Humidity average). SSS (Solar Radiation): Monthly solar radiation intensity influences weather patterns. Global Climate Indices:

SOI (Southern Oscillation Index) → Measures the air pressure difference between Tahiti and Darwin, influencing the El Niño and La Niña phenomena.

IOD (Indian Ocean Dipole) → An index describing the sea surface temperature difference between the western and eastern parts of the Indian Ocean, which can affect rainfall patterns in Indonesia.

A. Exploratory Data Analysis

This process helps us get a closer look at the dataset's contents, such as the characteristics of each variable, their distribution, and whether there are potential issues such as missing data, outliers, or other anomalies. During this exploration phase, several tasks are performed, including:

1. Viewing the data structure
2. Checking data types
3. Reviewing value distribution
4. Observing time trends
5. Identifying missing or invalid values
6. Analyzing relationships between variables

TABLE I. METADATA

No	Fitur (Unit)	[Min, Max]	[Mean, Stdev]
Tekanan Udara (millibar)			
1	P (MSL)	1008.50, 1013.70	1010.93, 0.86
2	P0 (Stasiun)	1006.21, 1011.40	1008.60, 0.85
Temperature Udara (°C)			
3	T07	21.20, 30.80	24.59, 0.83
4	T13	26.10, 38.80	29.55, 0,99
5	T18	24.70, 29.40	27.73, 0.62
6	T	25.00, 28.10	26.60, 0.62
7	Tx	30.50, 34.80	32.79, 0.69
8	Tn	18.20, 24.30	22.29, 0.93
Relative Humidity (%)			
9	RH07	88.00, 98.00	94.81, 1.70
10	RH13	57.00, 97.00	73.26 ,4.95

11	RH18	71.00, 96.00	81.66, 3.46
12	RH	77.00, 96.00	86.15, 2.68
Other			
13	SSS	2.30, 84.40	42.25, 18.11
14	CH	0.00, 926.90	275.21, 151.95
15	SOI	-28.60, 27.10	-0.44, 10.62
16	IOD	-1.18, 1.81	-0.01, 0.47

Table I presents the metadata of all features used in this study, including measurement units, minimum and maximum values, as well as the mean and standard deviation (stdev). These features cover air pressure, air temperature at different observation times (07:00, 13:00, and 18:00), relative humidity, and additional parameters such as Sea Surface Salinity (SSS), rainfall (CH), the Southern Oscillation Index (SOI), and the Indian Ocean Dipole (IOD). This statistical summary provides an initial overview of the distribution and variability of the input data used for monthly rainfall prediction modeling.

B. Feature Selection

In building an accurate and efficient prediction model, selecting the right features is a crucial step. Too many features can make the model complex and slow, while too few can reduce the quality of predictions. Therefore, in this study, the Hybrid Feature Selection Framework approach is used, which is a combined method of three feature selection techniques.

With the combination of these three approaches, the selected features are believed to have a significant contribution to the accuracy of monthly rainfall predictions. The Voting Feature Selector means that features will be selected if they are included in at least two of the three methods mentioned above. This approach strikes a balance between the robustness of statistical analysis and the power of machine learning models, making the selected features more validated from various.

C. Base Learner

Three algorithms were selected as base learners in the stacking scheme due to their ability to handle various types of data and their complementary characteristics:

Random Forest (RF) – An ensemble method based on trees that is resistant to overfitting and effective in identifying feature interactions. One effective way to enhance prediction accuracy is by using a random forest ensemble model, which combines the results of multiple decision trees. This approach allows the model to learn from different perspectives, leading to more stable and reliable predictions [6], [7].

K-Nearest Neighbors (KNN) – A non-parametric model that works based on the proximity of feature values, suitable for detecting recurring local patterns. The KNN algorithm is a reliable approach that makes predictions by averaging the values of nearby data points. It determines what counts as “nearby” based on the distance between each observation and the input being evaluated [8].

Decision trees (DT) – are commonly used in operations research, especially in decision analysis, to help find the best way to reach a goal. This model looks like a tree, where each branch represents a question that helps classify the data, and the leaves show the final result or category of the data [8].

XGBoost (XGB)– A boosting algorithm that is very popular due to its high accuracy and computational efficiency, excelling in handling tabular data. The XGBoost algorithm is a highly effective machine learning method that's gained popularity for time series forecasting. It works by combining multiple regression trees to make predictions and is especially good at handling seasonal and nonlinear patterns in data [9].

Linear Regression (LR) – is one of the simplest types of algorithms. Its main goal is to reduce the gap between predicted values and actual data. This algorithm is designed to produce numerical (quantitative) results, and it's often used to make predictions about future outcomes based on existing data [10].

Adaptive Boosting (ADB) – also known as AdaBoost, is an iterative algorithm first introduced by Freund and Schapire in 1997. It is an ensemble learning method that aims to build a strong predictive model by combining several simple models, known as "weak classifiers."

The core idea of AdaBoost is to run a simple learning algorithm repeatedly, while gradually shifting the focus toward training data that is harder to predict. This is done by adjusting the weights (or probabilities) of the data in each iteration [11].

D. Adaptive Stacking Ensemble Learner

After all base learners are trained and evaluated using validation metrics, the next step involves constructing the final predictive model through a stacking ensemble framework. Stacking ensemble learning is a powerful technique that integrates multiple models in a layered architecture to enhance predictive performance. It leverages the strengths of diverse algorithms and mitigates individual weaknesses, thereby reducing error in both classification and regression tasks [12].

Unlike conventional stacking methods that predetermine the meta-learner, this study initially proposed an adaptive stacking strategy. The main idea was to promote the best-performing base learner—based on key metrics such as R^2, RMSE, and MAE—as the meta-learner. However, during empirical evaluation, it was discovered that the model with the highest individual performance—Linear Regression ($R^2 = 0.931$)—did not necessarily yield the best results when applied as the final estimator in the ensemble framework.

To refine the strategy, several high-performing base learners were evaluated as potential meta-learners. In this extended approach, the top three models based on individual performance—Linear Regression, AdaBoost, and XGBoost—were each tested as meta-

learners. This allowed the stacking mechanism to be adaptive not only in selecting a single best learner but in exploring how different strong learners perform in combining outputs from the base layer (Random Forest, KNN, Decision Tree, and others).

As an additional enhancement, a voting-based ensemble was constructed using the outputs from the three stacking models, each with one of the top-ranked meta-learners. By combining their predictions through a voting mechanism, the ensemble aimed to further reduce variance and capitalize on the strengths of each configuration. The results from this voting-based stacking demonstrated improved generalization and accuracy, outperforming individual stacking variants.

This strategy reaffirms the foundational principle of adaptive stacking—not to select the meta-learner rigidly based on isolated validation scores, but to assess ensemble effectiveness empirically. By incorporating multiple high-performing candidates as meta-learners and integrating their outputs through voting, the proposed approach remains flexible, data-driven, and capable of capturing complex inter-model interactions. Ultimately, this leads to a more resilient and accurate predictive system, particularly valuable for tasks such as monthly rainfall prediction where data variability and non-linearity are prevalent.

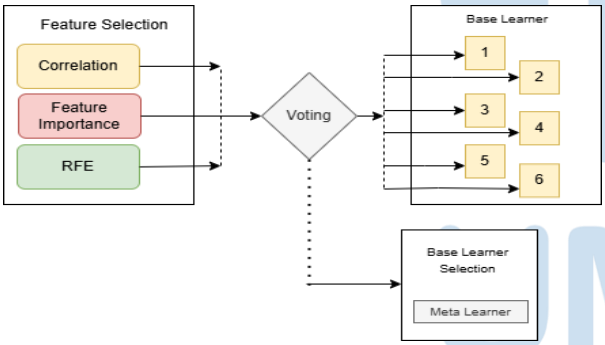


Fig. 1. Voting Feature Selector

"Fig 1" illustrates the architecture of the proposed adaptive stacking model incorporating hybrid feature selection and ensemble learning. Initially, three feature selection techniques—Correlation, Feature Importance, and Recursive Feature Elimination (RFE)—are applied independently. A voting mechanism is then employed to determine the optimal feature subset. The selected features are used to train multiple base learners, including Random Forest (RF), K-Nearest Neighbors (KNN), and XGBoost (XGB). Subsequently, a meta-learner is constructed by aggregating the outputs of the best-performing base learners to improve predictive accuracy and generalization.

Base models, or Level-0 models, are the first set of algorithms trained on the original data. They each make their own predictions, which are then used as input for the next step. The Level-1 model, known as the meta-model, learns how to blend those predictions from the base models in the most effective way to improve overall accuracy [13].

E. Tuning Hyperparameter

To make the tuning process more efficient and comprehensive, this study uses the Grid Search technique combined with Cross-Validation. With this approach, the system will try various combinations of hyperparameter values and evaluate the performance of each combination using training data that is randomly split into several folds. The final result of this process is the best combination of settings for each model that provides the most accurate and stable predictions.

Tuning is performed not only on the base models—XGBoost, AdaBoost, Random Forest (RF), K-Nearest Neighbors (KNN), Decision Tree (DT), and Linear Regression (LR)—but also on the meta-learner used in the adaptive stacking scheme. In this way, every layer of the modeling process is thoroughly optimized to operate harmoniously and effectively.

Cross-Validation Strategy – Given the relatively small size of the dataset—consisting of 408 monthly samples over a 34-year period—model validation becomes a crucial step to ensure the results are unbiased and generalize well to unseen data. To address this, cross-validation techniques were employed. During the hyperparameter tuning phase for the base models (XGBoost, AdaBoost, RF, KNN, DT, and LR), a 3-fold cross-validation scheme was applied. For the final training of the stacking ensemble, a 5-fold cross-validation was used to improve the model's robustness and evaluation stability. Although the folds were generated randomly, attention was given to maintaining balanced data distribution due to the temporal and seasonal nature of climatological data. This validation strategy helps reduce the risk of overfitting and provides a more reliable estimate of model performance under real-world conditions.

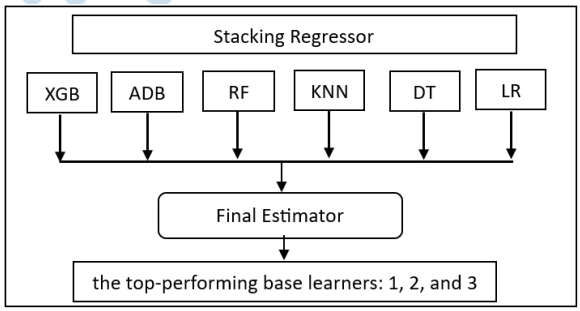


Fig. 2. Adaptive Stacking Ensemble

"Fig 2" The figure illustrates a brief overview of the stacking ensemble learning process. After completing the feature selection phase, the next step involves constructing a prediction model using the Adaptive Stacking Ensemble approach. In this setup, multiple base models—namely XGBoost, AdaBoost, Random Forest, K-Nearest Neighbors, Decision Tree, and

Linear Regression—are simultaneously trained to generate initial predictions.

Following the adaptive stacking principle, instead of selecting a single meta-learner, the final estimator is constructed using the top-performing base learners, as determined by their predictive performance across evaluation metrics. Specifically, the three best-performing models are promoted to serve as meta-learners, and their outputs are integrated in the final stage to improve robustness and accuracy. This strategy allows the ensemble to leverage both linear and nonlinear strengths across models, resulting in a more reliable and generalized prediction system.

III. Results and Discussion

This section presents the outcomes of the proposed monthly rainfall prediction model along with a comprehensive analysis of its performance. The results are discussed in relation to the accuracy of predictions, the effectiveness of the hybrid feature selection framework, and the comparative performance of different machine learning algorithms used. Furthermore, the impact of each selected feature on the model's predictive capability is evaluated, followed by a discussion on how the adaptive stacking approach enhances generalization across seasonal data. The findings are interpreted based on statistical metrics and are compared against baseline and conventional ensemble models to highlight the advantages of the proposed method.

TABLE II. FEATURE SELECTION

No.	Feature Selection	Selected Features
1	Correlation	RH13, RH, RH18, T13, RH07, T18, T, SSS, Tx
2	Feature Importance	RH13, T13, SSS, RH, Tx
3	RFE	P0, T07, T, Tn, RH13
4	Voting Feature Selector	RH13, RH, T13, T, SSS, Tx

Table II presents the results of feature selection obtained from three different methods: Correlation, Feature Importance, and Recursive Feature Elimination (RFE). Correlation Analysis selects features such as RH13, RH, RH18, T13, RH07, T18, T, SSS, and Tx, which show a strong relationship with the target. Meanwhile, the Feature Importance method (using decision tree models) identifies RH13, T13, SSS, RH, and Tx as important features. RFE, which gradually eliminates less important features, selects P0, T07, T, Tn, and RH13.

To achieve more stable and objective results, we apply the Voting Feature Selector by combining the results of these three methods. Finally, the features selected through voting are RH13, RH, T13, T, SSS, and Tx. This process ensures that the features used in the model are the most consistent and relevant according to various feature selection approaches.

After building the model with the Adaptive Stacking Ensemble approach, performance evaluations are conducted on each of the base learners, namely Random Forest (RF), K-Nearest Neighbors (KNN), and XGBoost (XGB), as well as on the stacking model itself. This evaluation aims to assess how well each model predicts the target. The three metrics used to evaluate model performance are the coefficient of determination (R^2), Mean Absolute Error (MAE), and Root Mean Square Error (RMSE) [14], [15]. The evaluation results from each model are summarized in the following Table III.

TABLE III. EVALUATION OF BASE LEARNERS AND ADAPTIVE STACKING

No	Base Learner		Evaluation		
		R^2	*SMAPE*	*MAE*	*RMSE*
1	XGB	0.907	14.34	0.305	0.360
2	ADB	0.909	13.68	0.293	0.356
3	RF	0.899	14.56	0.314	0.375
4	KNN	0.749	20.93	0.478	0.591
5	DT	0.811	18.10	0.419	0.513
6	LR	**0.931**	**11.81**	**0.259**	**0.309**

Table III presents the performance comparison of six base learner models—XGBoost (XGB), AdaBoost (ADB), Random Forest (RF), K-Nearest Neighbors (KNN), Decision Tree (DT), and Linear Regression (LR) evaluated using four metrics: coefficient of determination (R^2), symmetric mean absolute percentage error (SMAPE), mean absolute error (MAE), and root mean square error (RMSE).

Among all models, Linear Regression (LR) demonstrates the most outstanding performance, achieving the highest R^2 value of 0.931, which reflects its strong ability to explain the variance in monthly rainfall. Furthermore, LR recorded the lowest error across all other metrics—SMAPE of 11.81, MAE of 0.259, and RMSE of 0.309—highlighting not only its high predictive accuracy but also its effectiveness in modeling the predominantly linear relationships found in the dataset.

Following LR, AdaBoost and XGBoost also achieved strong results with R^2 values of 0.909 and 0.907, respectively. Both models outperformed RF in terms of generalization accuracy and yielded relatively low error rates. Their ensemble structure allows them to capture more complex patterns, particularly where non-linearity is present. Random Forest, although slightly behind (R^2 = 0.899), still demonstrated stable and consistent performance, reinforcing its reliability as a base ensemble model.

In contrast, K-Nearest Neighbors (KNN) and Decision Tree (DT) performed less favorably. KNN showed the weakest overall performance with an R^2 of

0.749 and the highest errors (SMAPE = 20.93, MAE = 0.478, RMSE = 0.591), indicating a tendency toward overfitting and limited generalization. Similarly, DT reported suboptimal results (R^2 = 0.811, RMSE = 0.513), suggesting difficulty in capturing the variability of the rainfall data when used independently.

In summary, while ensemble methods such as AdaBoost, XGBoost, and RF offer robust performance and better flexibility in capturing nonlinear patterns, Linear Regression remains the best-performing base learner in this study. Its consistently high accuracy across all evaluation metrics reinforces the presence of strong linear components in the rainfall dataset. These results also underscore the importance of carefully selecting base learners in ensemble strategies, depending on the complexity and structure of the target variable.

TABLE IV. EVALUATION OF STACKING AND VOTING STACKING

No	Adaptive Stacking Model		Evaluation		
		R²	*SMAPE*	*MAE*	*RMSE*
1	XGB+ADB+RF+ KNN+DT (Meta Learner LR)	0.916	13.53	0.288	0.341
2	XGB+RF+KNN+ DT +LR (Meta Learner ADB)	0.914	13.24	0.288	0.345
3	ADB+RF+KNN+ DT +LR (Meta Learner XGB)	0.913	13.43	0.293	0.347
4	Voting Stacking (Meta Learner LR+ADB+XGB)	0.917	13.33	0.287	0.339

Table IV presents a comparative analysis of four adaptive stacking ensemble configurations, each constructed from various combinations of base learners and meta-learners. The models are evaluated using four key performance metrics: the coefficient of determination (R^2), symmetric mean absolute percentage error (SMAPE), mean absolute error (MAE), and root mean square error (RMSE).

Among all configurations, Model 4, which employs a hybrid meta-learner composed of Linear Regression (LR), AdaBoost (ADB), and XGBoost (XGB) through a soft voting mechanism, delivers the best overall performance. It achieves the highest R^2 score (0.917), the lowest RMSE (0.339), and the lowest MAE (0.287), indicating strong predictive accuracy and generalization capability. The combination of diverse meta-learners enables this approach to effectively capture both linear and non-linear patterns in the predictions generated by the base learners.

Model 1, which uses Linear Regression as a single meta-learner stacked over five base learners (XGB, ADB, RF, KNN, and DT), also demonstrates competitive performance with an R^2 of 0.916 and RMSE of 0.341. Although slightly less accurate than the voting-based ensemble, it outperforms both Model 2 and Model 3, which utilize AdaBoost and XGBoost respectively as meta-learners.

Interestingly, when Linear Regression is applied independently (i.e., as a standalone model without stacking), it achieves even better results—R^2 of 0.931, SMAPE of 11.81, MAE of 0.259, and RMSE of 0.309—outperforming all stacking models. This finding suggests that Linear Regression is highly effective when applied directly to the original feature space of the dataset.

However, when used as a meta-learner in the stacking ensemble, its performance slightly degrades. This discrepancy may be attributed to the nature of the input it receives at the meta-level: rather than raw features, it processes predictions from the base learners, which may contain correlated errors and non-linear interactions. As a purely linear model, Linear Regression may struggle to effectively integrate such complex prediction spaces. In contrast, when applied directly to the raw data, it can fully leverage linear dependencies and underlying statistical distributions.

Therefore, the following conclusions can be drawn:

- If the objective is to achieve the highest standalone accuracy, then Linear Regression as an individual model is the most effective choice.
- If the goal is to obtain balanced and robust performance within an ensemble framework, the voting stacking approach (Model 4) is the most advantageous, due to its ability to combine the strengths of multiple meta-learners.
- Stacking ensembles with a single meta-learner may offer good results but tend to be less flexible and adaptive compared to hybrid strategies that employ voting across multiple meta-models.

These results reinforce the importance of adaptivity in ensemble learning strategies, particularly in data-driven domains like rainfall prediction, where capturing diverse patterns is essential for generalization.

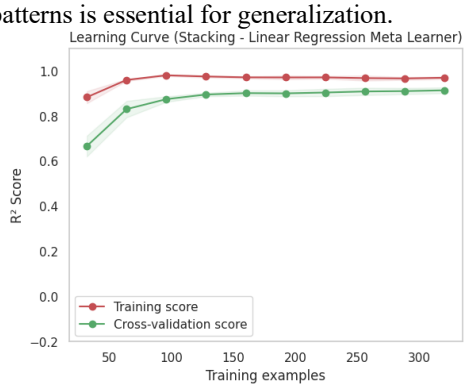


Fig. 3. Learning Curve Stacking (Linear Regression - R^2)

The "Fig. 3," illustrates the learning curve for the stacking ensemble model using Linear Regression as the meta-learner. The training and cross-validation R^2 scores demonstrate a stable and converging pattern as

the number of training examples increases. The training score remains consistently high (close to 0.97), indicating low bias, while the cross-validation score gradually improves and stabilizes around 0.91, suggesting good generalization. The narrow gap between both curves confirms that the model does not suffer from significant overfitting or underfitting, reflecting a well-balanced bias-variance tradeoff.

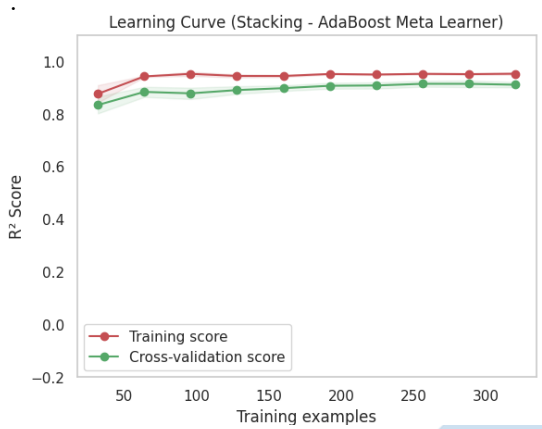


Fig. 4. Learning Curve Stacking (AdaBoost - R²)

The "Fig. 4," shows the learning curve of the stacking ensemble model with AdaBoost as the meta-learner. The training R² score remains consistently high (above 0.93), while the cross-validation score steadily improves and plateaus around 0.91 as the number of training examples increases. The narrow and stable gap between training and validation curves indicates low variance and good generalization ability. This pattern suggests that the model is well-fitted, with minimal risk of overfitting or underfitting, making AdaBoost an effective choice for meta-learning in this stacking configuration.

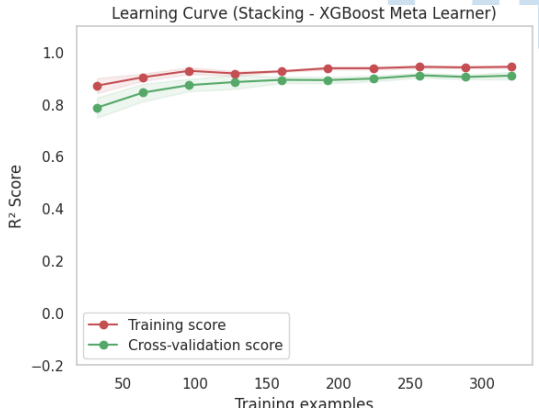


Fig. 5. Learning Curve Stacking (XGBoost - R²)

The "Fig. 5," presents the learning curve of the stacking ensemble model utilizing XGBoost as the meta-learner. The training R² score remains consistently high (above 0.92), while the cross-validation score steadily improves and converges toward 0.90 as more training examples are added. Although a slight gap persists between the training and validation curves, it narrows progressively, indicating improved generalization with increased data. This trend suggests a stable learning process with moderate variance and reflects XGBoost's strong capability in capturing complex nonlinear relationships when used as a meta-learner within the stacking framework.

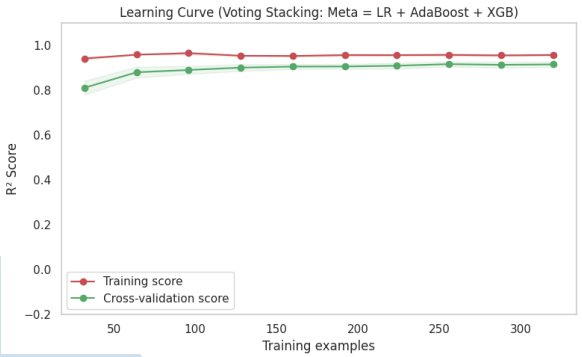


Fig. 6. Learning Curve Stacking (Voting LR+ADB+XGB - R²)

The "Fig 6" illustrates the learning curve for the voting-based stacking ensemble, which integrates three meta-learners—Linear Regression (LR), AdaBoost (ADB), and XGBoost (XGB)—through a soft voting strategy. The training R² score remains consistently high (≈0.94), indicating the model's strong fit on the training data. Meanwhile, the cross-validation R² score steadily increases with more training examples, stabilizing around 0.917, which is the highest among all evaluated configurations.

The narrow and stable gap between the training and validation curves suggests a well-balanced bias-variance tradeoff, with no signs of overfitting or underfitting. This behavior highlights the robust generalization capability of the voting ensemble, benefiting from the complementary strengths of linear and nonlinear learners. The curve confirms that the model continues to learn effectively as more data is introduced and that it maintains high predictive stability across various training set sizes.

Overall, the learning curve validates the effectiveness of combining multiple strong meta-learners via soft voting, making this configuration the most reliable and accurate among the tested stacking strategies.

Stacking ensemble learning works in two layers: a group of base models and a meta-learner. First, the base models are trained using the original training data. Then, their predictions are gathered and used to train the meta-learner, which learns how to best combine those outputs for more accurate final results [16].

IV. Conclusion

This study has successfully demonstrated that the integration of a Hybrid Feature Selection Framework with an Adaptive Stacking Ensemble significantly enhances the accuracy and robustness of monthly rainfall prediction models. By combining Correlation Analysis, Feature Importance, and Recursive Feature Elimination (RFE) through a voting mechanism, the proposed feature selection approach effectively identifies the most relevant meteorological predictors while excluding less informative variables—such as the global climate indices SOI and IOD—which were not selected by any method. This led to a more parsimonious and computationally efficient model without sacrificing predictive performance.

Experimental results confirm that the proposed Adaptive Stacking approach outperforms individual learners and conventional ensemble methods. While standalone Linear Regression recorded the highest individual performance ($R^2 = 0.931$), it did not retain this advantage when used as a meta-learner in the stacking framework. Instead, the most effective configuration was achieved through voting-based meta-learning, combining Linear Regression, AdaBoost, and XGBoost, which produced the best overall ensemble performance with an R^2 of 0.917, MAE of 0.287, and RMSE of 0.339.

The learning curves of each stacking configuration further validated the model's generalization capability. The voting ensemble showed the most stable bias-variance tradeoff, benefiting from the diversity of its meta-learners. These findings emphasize that in adaptive ensemble learning, meta-learner selection should not be rigidly based on individual model scores but evaluated empirically within the ensemble context.

Overall, this research presents a robust, flexible, and data-driven predictive framework that can adapt to the nonlinear and dynamic nature of rainfall patterns. Its practical applicability holds strong potential for climate-sensitive sectors such as agriculture, hydrology, water resource management, and early warning systems for hydrometeorological hazards.

V. Suggestions

Based on the results and insights gained from this study, several directions are proposed for future research to further enhance the adaptability and predictive strength of the proposed framework. One potential improvement involves expanding the variety of meta-learners used in the stacking ensemble. While this study focused on top-performing learners such as Linear Regression, AdaBoost, and XGBoost, incorporating other advanced algorithms—such as support vector machines, deep learning models, or neural-based regressors—may improve performance under more complex or highly non-linear climate conditions.

Additionally, considering the spatial and temporal variability of rainfall, future research could explore region-specific adaptations or spatio-temporal extensions of the model to improve its generalization across different climatic zones. This would be particularly relevant for scaling the model to a national or regional level, where rainfall dynamics may vary significantly.

Enhancing the hyperparameter optimization process is also a promising avenue. The use of more sophisticated methods—such as Bayesian optimization or evolutionary algorithms—could yield better parameter configurations than traditional grid search, thus improving overall model efficiency and accuracy.

Furthermore, integrating additional climate-related indicators, particularly those linked to ocean-atmosphere interactions, may help refine the model's ability to capture long-term and seasonal rainfall anomalies. Lastly, due to its robust and flexible nature, the proposed adaptive stacking framework holds significant promise for broader applications beyond rainfall prediction. It could be extended to areas such as drought monitoring, precision agriculture, flood risk management, and climate-related decision support systems, offering valuable tools for anticipating and mitigating the impacts of environmental variability.

References

[1] E. Oprasmani, T. Amelia, and E. Muhartati, "Membangun Masyarakat Peduli Lingkungan Pesisir Melalui Edukasi Kepada Masyarakat Kota Tanjungpinang Terkait Pelestarian Daerah Pesisir," To Maega : Jurnal Pengabdian Masyarakat, vol. 3, no. 2, p. 66, Aug. 2020, doi: 10.35914/tomaega.v3i2.372.

[2] P. D. Riajaya, "Rainy season period and climate classification in sugarcane plantation regions in indonesia," in IOP Conference Series: Earth and Environmental Science, Institute of Physics Publishing, Jan. 2020. doi: 10.1088/1755-1315/418/1/012058.

[3] R. Ruqoyah, Y. Ruhiat, and A. Saefullah, "Analisis Klasifikasi Tipe Iklim Dari Data Curah Hujan Menggunakan Metode Schmidt-Ferguson (Studi Kasus: Kabupaten Tangerang)," Serang, Jan. 2023. doi: https://doi.org/10.23960/jtaf.v11i1.327.

[4] Juliati, "ANALISIS KARAKTERISTIK CURAH HUJAN DENGAN MENGGUNAKAN KLASIFIKASI SCHMIDT-FERGUSSON DI KOTA MAKASSAR," Jurnal Sains dan Pendidikan Fisika (JSPF), vol. 19, no. 2, pp. 229–235, Aug. 2023, doi: prefix10.35580.

[5] S. R. Aisy, M. K. Ramadhan, A. S. Salsabila, and R. Kurniawan, "Perbandingan Algoritma Klasifikasi Data Mining dalam Memprediksi Curah Hujan di Jawa Barat," Seminar Nasional Sains Data, pp. 180–192, 2024, doi: https://doi.org/10.33005/senada.v4i1.179.

[6] S. D. Latif et al., "Assessing rainfall prediction models: Exploring the advantages of machine learning and remote sensing approaches," Nov. 01, 2023, Elsevier B.V. doi: 10.1016/j.aej.2023.09.060.

[7] A. Raut, D. Theng, and S. Khandelwal, "Random Forest Regressor Model for Rainfall Prediction," in 2023 International Conference on New Frontiers in Communication, Automation, Management and Security, ICCAMS 2023, Institute of Electrical and Electronics Engineers Inc., 2023. doi: 10.1109/ICCAMS60113.2023.10526085.

[8] S. Biruntha, B. S. Sowmiya, R. Subashri, and M. Vasanth, "Rainfall Prediction using kNN and Decision Tree," in Proceedings of the International Conference on Electronics and Renewable Systems, ICEARS 2022, Institute of Electrical and

Electronics Engineers Inc., 2022, pp. 1757–1763. doi: 10.1109/ICEARS53579.2022.9752220.

[9] P. Mishra et al., "Modeling and forecasting rainfall patterns in India: a time series analysis with XGBoost algorithm," Environ Earth Sci, vol. 83, no. 6, Mar. 2024, doi: 10.1007/s12665-024-11481-w.

[10] C. Vijayalakshmi, K. Sangeeth, R. Josphineleela, R. Shalini, K. Sangeetha, and D. Jenifer, "Rainfall Prediction using ARIMA and Linear Regression," in 2022 1st International Conference on Computer, Power and Communications, ICCPC 2022 - Proceedings, Institute of Electrical and Electronics Engineers Inc., 2022, pp. 366–370. doi: 10.1109/ICCPC55978.2022.10072125.

[11] A. Belghit, M. Lazri, F. Ouallouche, K. Labadi, and S. Ameur, "Optimization of One versus All-SVM using AdaBoost algorithm for rainfall classification and estimation from multispectral MSG data," Advances in Space Research, vol. 71, no. 1, pp. 946–963, Jan. 2023, doi: 10.1016/j.asr.2022.08.075.

[12] M. H. D. M. Ribeiro, R. G. da Silva, S. R. Moreno, V. C. Mariani, and L. dos S. Coelho, "Efficient bootstrap stacking ensemble learning model applied to wind power generation forecasting," International Journal of Electrical Power and Energy Systems, vol. 136, no. September 2021, 2022, doi: 10.1016/j.ijepes.2021.107712.

[13] P. G. Jaiswal et al., "A Stacking Ensemble Learning Model for Rainfall Prediction based on Indian Climate," 2023 6th International Conference on Information Systems and Computer Networks, ISCON 2023, pp. 1–6, 2023, doi: 10.1109/ISCON57294.2023.10112077.

[14] D. Chicco, M. J. Warrens, and G. Jurman, "The coefficient of determination R-squared is more informative than SMAPE, MAE, MAPE, MSE and RMSE in regression analysis evaluation," PeerJ Comput Sci, vol. 7, pp. 1–24, 2021, doi: 10.7717/PEERJ-CS.623.

[15] Roshani et al., "Analyzing trend and forecast of rainfall and temperature in Valmiki Tiger Reserve, India, using non-parametric test and random forest machine learning algorithm," Acta Geophysica, vol. 71, no. 1, pp. 531–552, Feb. 2023, doi: 10.1007/s11600-022-00978-2.

[16] D. Chicco, M. J. Warrens, and G. Jurman, "The coefficient of determination R-squared is more informative than SMAPE, MAE, MAPE, MSE and RMSE in regression analysis evaluation," PeerJ Comput Sci, vol. 7, pp. 1–24, 2021, doi: 10.7717/PEERJ-CS.623.

ISSN 2355-3286

Micro-Scale CPV Performance Enhancement through V-Trough Concentration and Passive Cooling

Nicholas Robert[1], Johanes Dimas Paramasatya[1], and Niki Prastomo[1,*]
[1] Physics Engineering Department,
Universitas Multimedia Nusantara, Scientia Garden, Jl. Boulevard Gading Serpong, Curug Sangereng, Kelapa Dua, Tangerang, Banten 15810, Indonesia
[1,*] niki.prastomo@umn.ac.id

Accepted June 01, 2025
Approved June 24, 2025

***Abstract*—The global reliance on fossil fuels has driven the need for clean, renewable alternatives. Concentrated Photovoltaic (CPV) systems offer a promising solution by increasing energy yield per unit area, particularly in regions with high solar irradiance. This study investigates the performance enhancement of a micro-scale CPV system through the integration of a V-trough optical concentrator and passive thermal regulation mechanisms. Five system variants were developed and tested: a baseline with no enhancement, a standard CPV, and three CPV systems incorporating heat sinks, heat pipes, and a hybrid of both. Optical simulations were performed to achieve a 2× concentration ratio using planar mirrors angled at 60°, while all cooling systems relied on passive methods to maintain simplicity and low cost. Field tests conducted in a tropical environment revealed that all CPV systems outperformed the baseline, with the hybrid-cooled system delivering the highest average power output—138.76 mW, a 32.37% improvement over the baseline. Surface temperatures were also significantly reduced, with the hybrid system lowering temperatures by up to 6.8°C. These results highlight the synergistic potential of combining optical and passive thermal enhancements in compact CPV designs, providing a scalable, cost-effective solar energy solution suitable for rural and off-grid applications in high-irradiance regions.**

***Index Terms*—CPV; heat pipe; heat sink; passive cooling; photovoltaic efficiency; V-trough concentrator**

I. Introduction

Global energy demand has risen by 4.5% between 2024 and 2025 alone and is projected to continue to increase rapidly [1]. Although fossil fuels have historically been used globally for most energy applications, they have significant drawbacks [2]. They are not renewable within human timescales [3], and contribute significantly to climate change, air pollution, and environmental destruction through their extraction and usage [2].

Solar energy is a promising and abundant renewable energy source [4]. Solar energy has immense potential to meet global energy needs and mitigate climate change [5-7]. A common method of gathering solar energy into usable electricity is via photovoltaic (PV) technology. Unfortunately, solar PV has inherent efficiency constraints. Due to its working principle, PV cells can only use a limited range of wavelengths of light efficiently; excess energy then manifests as heat. Excessive heat on solar PV cells further decreases energy collection efficiency.

A potential method of maximizing power generation from PV cells involves concentrating light rays on them, generally known as concentrated photovoltaics (CPV, also called concentrating photovoltaics). Unlike conventional PV systems, CPV aims to reduce the cost per generation capacity by using optical components called concentrators to focus sunlight onto cells. This increases the density of solar irradiance on the cells and boosts overall power production and production per unit area. Furthermore, in the case of some multi- junction cells, efficiency also rises within CPV [8].

A simple and low-cost implementation of CPV uses a V-trough design, where flat mirrors are placed next to a line of solar cells to increase the concentration of incident sunlight. Unfortunately, this also increases the amount of heat received by the cells. This issue of heat concentration would lead to a phenomenon called thermal derating and would further decrease the efficiency if left unsolved. Previous research suggests that conventional PV cells would lose efficiency at a rate of approximately 0.45–0.5% per 1 °C increase in operating temperature above the optimum [9], [10].

Cooling systems can mitigate the negative effects of this increased heat load. These systems may be passive, such as heat pipes and heat sink arrangements, which can be implemented without requiring additional power, and rely purely on natural convection and conduction. However, active cooling systems require active components such as pumps, and fans may be better suited for more intensive heat removal [11], [12].

While previous studies have separately evaluated optical concentration or passive cooling implementations in CPV systems, few have explored their combined implementation in a compact, micro-scale format tailored to tropical environments [2], [4], [12]. This study aims to address that gap by evaluating

the combined effect of optical and thermal enhancements on CPV performance under real tropical conditions. We used variants of a simple V-trough CPV for this purpose, with variants that test which cooling system is the most effective and produces the greatest efficiency gains.

II. Related Works

A. History of CPV

CPV emerged in the 1970s as a method for enhancing solar efficiency in high-irradiance regions. Despite its early promise, its adoption was limited because of high system costs and poor competitiveness with conventional energy. The development slowed, but small-scale research continued. Recently, interest has grown because concentrator materials remain significantly cheaper than high-efficiency PV cells [13].

B. Low-concentration Photovoltaics

CPV systems are often classified by their concentration ratio (CR), which is expressed as multiples of the solar intensity. Low-concentration photovoltaics (LCPV), defined by concentration ratios (CR) below 10× (or 10 "suns"), offer a balance between performance gains and system simplicity. Various optical concentrators have been proposed, including Fresnel lenses, linear reflectors, and compound parabolic concentrators (CPC). However, many require active tracking systems, increasing cost and complexity [14].

C. V-Through Concentrators

V-trough concentrators are simple concentrator systems that use only two planar mirrors to focus sunlight onto PV cells without any necessary tracking [13]. Thus, they are suitable for deployment in microscale CPV systems. Prior studies reported CR values between 1.4× and 3×, with power output increases of up to 2.6× over baseline systems in the highest-CR systems [15]–[17]. Their passive geometry and low material requirements make them a practical solution for small-scale low-budget deployments in high-irradiance regions.

D. Cooling Systems

A PV system can convert solar energy into electrical energy; however, this conversion is not perfect, and some of the energy within the PV system is converted into heat [18]. The heat generated within a PV system may lead to a decrease in electrical energy production [19]. For every 1 °C increase in temperature above 25 °C, a 0.45% reduction in the PV module efficiency occurs [9]. As a result, the implementation of a cooling system for PV would mitigate the reduction or prevention of heating effects.

E. Heat Sink Cooling System

Heat-sink cooling systems have been successfully implemented in CPV systems. One study tested 36 heat sink variants to determine the most effective. It was found that a heat sink with a wide contact area and large gap between fins was the most effective variant, with an average temperature reduction of 7.5˚C [20].

Another research tested the impact of heat sink cooling on a triple junction CPV system and observed a temperature reduction of 2.8˚C to 33.3˚C based on the CPV's concentration ratio [11].

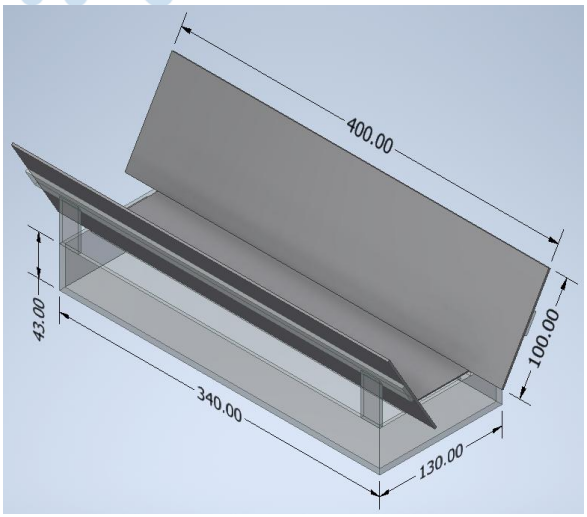
F. Heat Sink Cooling System

Considerable research has been conducted to observe the effects of heat pipe implementation on the performance of PV systems. One study utilized a curving heat pipe to increase the contact area for the cooling system on a PV system; the implementation of the heat pipe resulted in a 19.45% power production increase and a temperature reduction of 10.5 °C [19]. Another study investigated the effects of different fluid ratios on a heat pipe used for PV cooling and found that a filling ratio of 45% fluid was the most effective and resulted in a 3.2% increase in electrical energy production.

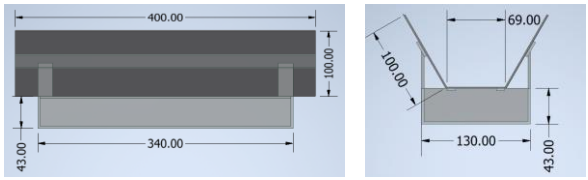
III. Methodology

A. System Architecture

This study employs a micro-scale CPV system with conventional silicon solar cells. One unit is used for each system variant. Four variants of a microscale V-trough CPV system were simulated, prototyped, and tested in real-life conditions, with another variant serving as a baseline with neither concentrators nor cooling, for a total of five systems. One variant uses no additional cooling, whereas the remaining three variants each employ a passive cooling system to manage the thermal load. Of the three variants that use cooling, one uses heat sinks, one uses heat pipes, and one uses both heat sinks and pipes simultaneously. Figure 1 shows the design renders of the CPV system with a concentrator.



(a) 3D perspective view



(b) Top view (c) Cross-sectional view

Fig. 1. Design of the V-trough CPV System (dimensions in mm)

B. System Architecture

A passive V-trough reflector was selected because of its simplicity, cost efficiency, and suitability for LCPV applications. The design consists of two planar mirrors angled to focus incoming sunlight onto a fixed PV receiver.

Simulations were performed using Phydemo Ray Optics to obtain the geometry of the appropriate CR. The target CR was 2, with a mirror angle of 60 °and receiver width of 69 mm. The 60° angle was selected as a practical compromise between concentration ratio and manufacturability. Steeper angles would increase the concentration ratio but reduce the acceptance angle and ease of assembly, which are critical factors in low-cost, rural-deployable systems [17]. Owing to material acquisition constraints, the mirrors slightly exceeded the ideal dimensions, resulting in minor optical spillovers. This configuration was chosen for its ease of assembly and repeatability under field conditions. Figure 2 shows the simulation conditions with concentrators.

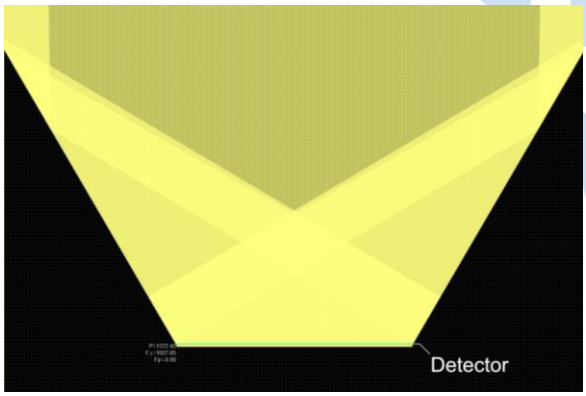


Fig. 2. Ray Simulation with Concentrator

The simulation confirms the intended focusing effect of the V-trough mirrors, with the majority of rays successfully directed toward the detector area. Although slight optical spillovers are observed due to dimensional compromises during construction, the concentration pattern remains consistent and well-aligned with the receiving surface. This validates the system's optical geometry resulting in a CR of 2.

C. Cooling System

For the purposes of this study, heat sink and heat pipe cooling systems were chosen because of their simplicity, low cost, and the relatively minor additional heat load of an LCPV system compared to non-concentrating PV systems. Both cooling systems are passive in nature, making their implementation much simpler than in an active cooling system. A third variant using a hybrid cooling system, which is a combination of a heat sink and a heat pipe cooling system used in tandem, was also designed and tested.

1) Heat sink cooling system:

Within the system, aluminium heat sinks were attached to the back of each solar cell using thermal adhesives, ensuring proper thermal conductivity. The heat sinks were placed in the geometric middle of each cell.

2) Heat pipe cooling system

A custom rounded rectangular copper pipe loop was used as the heat pipe. Half of the inner volume of the copper pipe was filled with deionized water to act as the working fluid. Each solar cell in the system was attached to a heat pipe with the ends of the heat pipe exposed to open air.

3) Hybrid cooling system

Both heat sink and heat pipe cooling systems were applied to this variant, using the same installation methods described above.

D. Experimental Setup

To evaluate the performance of the CPV variants, a series of controlled outdoor measurements were conducted. The setup involved real-time monitoring of electrical and thermal parameters under natural sunlight and subject to weather conditions. This phase uses consistent data acquisition protocols across all system configurations. Key components, instrumentation, and procedures are detailed below.

1) Location

Universitas Multimedia Nusantara (UMN) Rooftop, Tangerang Regency (6.2568˚S, 106.6185˚E)

2) Sensors and tools

- Current-voltage sensor: INA219
- Thermometer: Krisbow #10206574 IR
- Microcontroller: Arduino Uno

3) Data collection protocol

The data collection steps, as shown in Figure 3, were followed to acquire the real-world performance data of each system. Activities are done according to local solar noon at the corresponding time of year (≈ 11:40 across test days) with a total measurement time of 2 hours per day. Measurements were conducted in the timespan of one hour before solar noon and one hour after solar noon. To minimize environmental variability, all systems were tested concurrently in the same location.

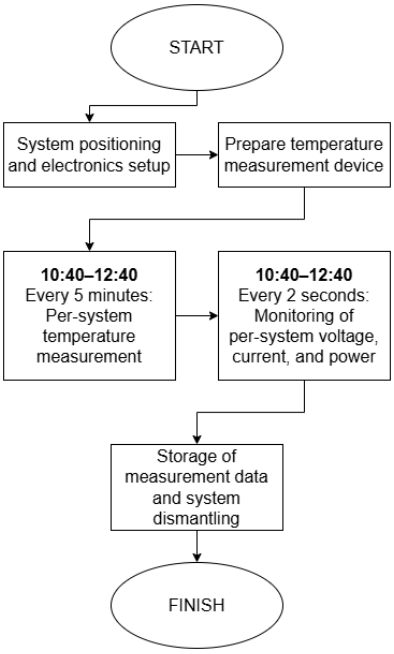


Fig. 3. Ray Simulation with Concentrator

Figure 4 shows a photograph showing all variations of the tested PV systems. The study's baseline is defined as a variant system with the same electronics and wiring as the rest, but with neither any concentrators nor cooling, measured in exactly the same manner and tested concurrently in the same location, ensuring direct comparability with the tested CPV variants.

From left to right, the order of the system variations are as follows: the baseline system without concentration, the CPV system without a cooler (CPV), the CPV system with a heat sink cooling system (CPV-HP), the CPV system with a heat pipe cooling system (CPV-HS), and the CPV system with a combination of both passive cooling systems (CPV-Hybrid).

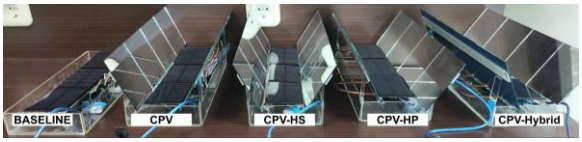


Fig. 4. Variants of the CPV Systems

Figure 5 shows the cooling system on the back of each cell in the case of the CPV-Hybrid variant.

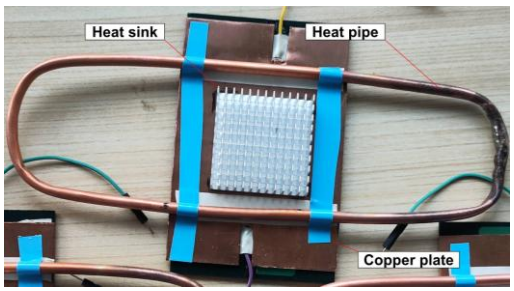


Fig. 5. Variants of the CPV Systems

The cooling configuration integrates a copper base plate as a thermal spreader in direct contact with the CPV cell, promoting efficient conduction in all three cooled variants. In the heat sink variants (CPV-HS and CPV-Hybrid), an aluminum heat sink placed in the center enhances surface area and convective heat dissipation. In the heat pipe variants (CPV-HP and CPV-Hybrid) variant, a loop of copper pipe filled halfway with deionized water to act as heat pipes is affixed along the sides of the plate to facilitate lateral heat distribution.

All components are mechanically fastened using thermally conductive adhesive and electrically insulating thermal tape, ensuring secure contact while maintaining modularity. This arrangement was designed to minimize complexity and cost while achieving significant thermal regulation gains without active cooling.

IV. Results and Discussion

A. V-Trough CPV vs Baseline

Performance analyses revealed that CPV systems show a much higher power output than non-concentrated PV systems across every test period, with maximum gains observed during peak irradiance. At irradiance levels near the peak (≈ 1000 W/m^2), the CPV system without cooling delivered up to 49.5% more power than the baseline, and the systems with cooling showed even better gains. The average performance of the CPV variant without cooling was approximately 22.2% better than that of the baseline, with the cooled systems exhibiting better performance than the non-cooled variant.

In contrast, under lower irradiance conditions, the relative performance gain decreased significantly for all the systems tested. This confirms a strong dependency relationship between CPV effectiveness and solar irradiance, where higher irradiance directly correlates with increased power output due to more effective light concentration, both in relative and absolute terms. Table I lists the relative power generation and electrical performance of each tested variant over test days.

TABLE I. AVERAGE CPV SYSTEM PERFORMANCE

Variant	Average V (V)	Average I (mA)	Average P (mW)	Improvement (vs. Baseline)
Baseline	1.03	101.84	104.95	-
CPV	0.95	136.88	129.11	22.23%
CPV-HP	0.97	136.03	132.56	25.33%
CPV-HS	1.02	133.24	136.04	29.45%
CPV-Hybrid	1.02	136.02	138.76	32.37%

These performance metrics strongly suggest that CPV systems are best suited for locations with consistently high solar irradiance. At lower irradiance, such as during cloudy periods, losses from reflection and

optical inefficiencies would outweigh the concentration benefits, reducing performance gains.

B. Cooling Mechanism Impact

As mentioned, the CPV system performance further increased with the addition of cooling. Systems implementing heat sink cooling experienced an operating temperature reduction of 0.5˚C– 1.6˚C; heat pipe results in a reduction of 0.8˚C– 3.3˚C; and lastly, hybrid cooling results in a temperature reduction of 2.2˚C– 6.5˚C compared to systems without any cooling.

For energy production, the best performer, hybrid cooling, resulted in a 32.37% increase compared to the baseline and 7.47% increase compared to CPV. These observed effects show a relationship between the reduction in the system cell temperature and increase in power production. Table II lists the effects of the cooling systems on the operating temperatures.

TABLE II. CPV SYSTEM OPERATING TEMPERATURES

Variant	Operating temperature range
CPV	41.6˚C – 44.8˚C
CPV-HP	41.4˚C – 44.1˚C
CPV-HS	41.0˚C – 43.7˚C
CPV-Hybrid	38.0˚C – 42.0˚C

C. Synergistic Effect

The integration of passive cooling notably improves the thermal regulation and power output under high-irradiance conditions. As shown in Table II, the non-cooled CPV variant reached 44.8˚C, while the hybrid-cooled system maintained the lowest temperatures (38.0 °C to 42.0 °C). Figure 6 illustrates the comparative power output and operating temperatures of all the tested configurations.

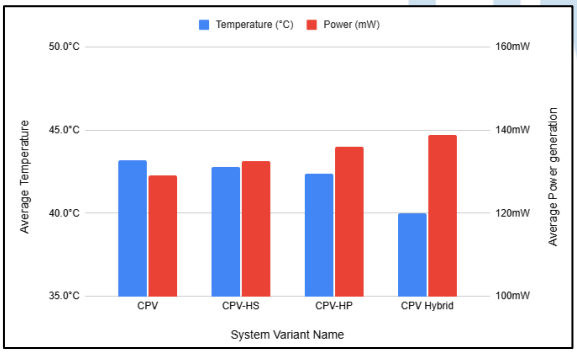


Fig. 6. Power Generation and Thermal Performance Per Variant

Table I shows that the CPV hybrid system achieved the highest average power output of 138.76 mW, a 32.37% improvement over the baseline, outperforming both CPV-HS (29.45%) and CPV-HP (25.33%), showing increased power generation aligned with reduced temperatures. This demonstrates a clear synergy between optical concentration and thermal management.

Although the hybrid system has greater assembly complexity, its superior electrical performance and thermal stability highlight the potential value of hybrid passive cooling in this implementation. The enhanced light concentration via the V-trough design effectively increased the photon flux per unit area, improving the charge carrier generation. However, it also imposes a higher thermal burden, highlighting the necessity of pairing the concentration with adequate thermal management to sustain efficiency gains.

D. Engineering Implications

The V-trough CPV system designed in this study emphasizes a low-cost and relatively scalable implementation, making it suitable as a basis for slightly larger-scale deployments in both rural and urban Indonesian contexts. And due to the exclusive use of only cheap and basic optical elements (flat mirrors) and widely available electrical components (mini conventional PV cells, Arduino, passive cooling), this implementation allows for simple replication and localized manufacturing without high upfront cost, high precision requirements, or any tracking systems.

While it can be argued that V-trough concentrators are more commonly used in testing scenarios rather than for deployment, their potential in rural settings should not be dismissed. In areas where cost, component availability, and ease of maintenance are more critical than peak efficiency, the use of inexpensive, easy-to-acquire flat mirrors and passive cooling offers a compelling trade-off for rural solar energy deployment. The absence of complex optics or solar tracking makes this system particularly attractive for localized, off-grid implementations where technical support is limited. In such contexts, the enhanced energy output enabled by basic concentration can already provide tangible benefits without significantly raising deployment cost or complexity.

In rural areas, this system offers an accessible solution for off-grid energy generation, particularly in high-irradiance regions with limited infrastructure. For urban applications, compact and passive designs enable integration on rooftops or unused land, with potentially minimal maintenance requirements. The passive cooling systems tested here also reduce the need for active thermal management, further simplifying deployment and lowering long-term costs while also increasing power generation, especially by preventing thermal derating.

Scalability is primarily limited by the need for constant irradiance and the linear arrangement required by V-trough geometry. However, the modularity of the system allows for flexible array expansion, particularly in distributed energy scenarios. This suggests the potential for broader adoption of Indonesia's growing decentralized energy initiatives. Further, although the absolute power output is low due to the micro-scale nature of the prototype, the relative improvement between configurations is representative and scalable to higher capacity systems.

Tropical regions are characterized by high average solar irradiance and frequent fluctuations due to intermittent cloud cover, atmospheric humidity, and aerosol content. This variability presents a unique challenge for CPV systems that rely heavily on direct

beam sunlight. In this study, although maximum gains were observed during peak irradiance, the relative improvement from the CPV systems was notably reduced under low-light conditions.

E. Limitations

Despite its promising results, the system remains a prototype and requires further refinement prior to large-scale deployment. The current design lacks waterproofing and must be manually disassembled and reassembled for each testing session, which limits its long-term durability and practical use.

Additionally, the absence of automated tracking may restrict the performance in non-optimal sun positions if deployed over the course of a year. Sun positions in the tropics thankfully mitigate this loss; however, further research and prototyping may reveal further increased performance with a solar tracking system.

Future studies should address these limitations through enclosure development, improved mounting systems, and passive or semi-active tracking integration. These improvements should enhance the robustness and scalability in real-world conditions over the long term.

V. Conclusion

The experimental results confirm that the V-trough CPV system significantly outperforms conventional PV setups under high irradiance. The integration of passive cooling, particularly the hybrid configuration, effectively reduced module temperatures and led to the highest power gains of 32.37% above the baseline. These findings underline the critical role of thermal management in enhancing CPV efficiency, particularly in compact, low-cost designs. The demonstrated synergy between the optical concentration and passive cooling offers a practical pathway for scalable solar solutions suitable for tropical resource-constrained environments. Future work may explore improving system durability, incorporating tracking systems, and extending deployment in real-world scenarios to validate the long-term performance.

Acknowledgment

The authors thank the Institute for Research and Community Service Universitas Multimedia Nusantara (UMN) for supporting this research through its internal funding scheme.

References

[1] [1] International Energy Agency, Electricity, Mar. 2025. [Online]. Available: https://www.iea.org/reports/global-energy-review-2025 /electricity.

[2] S. Kiyaee, Y. Saboohi, and A. Z. Moshfegh, “A new designed linear fresnel lens solar con- centrator based on spectral splitting for passive cooling of solar cells,” Energy Conversion and Management, vol. 230, p. 113 782, 2021. DOI: 10.1016/j.enconman.2020.113782.

[3] A. Ziemin´ska-Stolarska, M. Pietrzak, and I. Zbicin´ski, “Application of lca to determine environmental impact of concentrated photo- voltaic solar panels—state-of-the-art,” Energies, vol. 14, no. 11, p. 3143, 2021. DOI: 10.3390/en14113143.

[4] G. Wang, F. Wang, F. Shen, Z. Chen, and P. Hu, “Novel design and thermodynamic analysis of a solar concentration pv and thermal combined system based on compact linear fresnel reflec- tor,” Energy, vol. 180, pp. 133–148, 2019. DOI: 10.1016/j.energy.2019.05.082.

[5] H. H. Pourasl, R. V. Barenji, V. M. Khojastehnezhad, "Solar energy status in the world: A comprehensive review", Energy Reports, Vol. 10, pp. 3474-3493, 2023. DOI: 10.1016/j.egyr.2023.10.022.

[6] G. Molnár, D. Ürge-Vorsatz, S. Chatterjee, "Estimating the global technical potential of building-integrated solar energy production using a high-resolution geospatial model", Journal of Cleaner Production, Vol. 375, 2022, ISSN: 0959-6526. [Online]. DOI: 10.1016/j.jclepro.2022.134133.

[7] E. H. Adeh, S. P. Good, M. Calaf, C. W. Higgins, "Solar PV Power Potential is Greatest Over Croplands", Scientific Reports, Vol. 9, 2019, 11442. DOI:10.1038/s41598-019-47803-3.

[8] M. A. Green, E. D. Dunlop, J. Hohl-Ebinger, M. Yoshita, N. Kopidakis, and A. W. Ho- Baillie, “Solar cell efficiency tables (version 55),” Progress in Photovoltaics: Research and Applications, vol. 28, no. 1, pp. 3–15, Dec. 2019. DOI: 10.1002/pip.3228.

[9] E. Kozak-Jagieła, P. Cisek, and P. Ocłon´, Cool- ing techniques for pv panels: A review, Mar. 2023. DOI: 10.58332/scirad2023v2i1a03.

[10] F. Karimi, H. Xu, Z. Wang, J. Chen, and M. Yang, “Experimental study of a concentrated pv/t system using linear fresnel lens,” Energy, vol. 123, pp. 402–412, 2017. DOI: 10.1016/j.energy.2017.02.028.

[11] H. Refaey, M. Abdelrahman, M. A. Alharthi, S. Bendoukha, S. G. Khan, and M. Emam, Passive cooling of highly-concentrator triple- junction solar cell using a straight-finned heat sink: An experimental investigation, Dec. 2022. [Online]. DOI: 10.1016/j.csite.2022.102521.

[12] M. Sabry and A. Lashin, “Performance of a heat-pipe cooled concentrated photovoltaic/thermoelectric hybrid system,” Energies, vol. 16, no. 3, p. 1438, 2023. DOI: 10.3390/en16031438.

[13] A. L. Luque and A. Viacheslav, Concentrator photovoltaics. Springer-Verlag, 2007. [Online]. DOI: 10.1007/978-3-540-68798-6.

[14] A. Ejaz, H. Babar, H. M. Ali, et al., “Concen- trated photovoltaics as light harvesters: Outlook, recent progress, and challenges,” Sustainable Energy Technologies and Assessments, vol. 46, p. 101 199, 2021. DOI: 10.1016/j.seta.2021.101199.

[15] A. Ustaoglu, C. Kandilli, M. Cakmak, and H. Torlaklı, “Experimental and economical per- formance investigation of v-trough concentra- tor with different reflectance characteristic in photovoltaic applications,” Journal of Cleaner Production, vol. 272, p. 123 072, Nov. 2020. DOI: 10.1016/j.jclepro.2020.123072.

[16] R. V. Parupudi, H. Singh, and M. Kolokotroni, “Low concentrating photovoltaics (lcpv) for buildings and their performance analyses,” Applied Energy, vol. 279, 2020, ISSN: 0306-2619. [Online]. DOI: 10.1016/j.apenergy.2020.115839.

[17] M. Alnajideen and M. Gao, “A new configuration of v-trough concentrator for achieving improved concentration ratio of 3.0x,” Solar Energy Materials and Solar Cells, vol. 245, p. 111 877, Sep. 2022. DOI: 10.1016/j.solmat.2022.111877.

[18] M. G. Noxpanco, J. Wilkins, S. Riffat, "A Review of the Recent Development of Photovoltaic/Thermal (PV/T) Systems and Their Applications". Future Cities and Environment, Vol 6(1): 9, pp. 1–16, 2020. DOI: 10.5334/fce.97.

[19] U. Arachchige and S. Weliwaththage, Solar energy technology, Jul. 2020. ISSN: 2714-1837. [Online]. Available:

https://www.researchgate.net/publication/353306311_Solar_Energy_Technology.

[20] M. Krstic, L. Pantic, S. Djordjevic, et al., Pas- sive cooling of photovoltaic panel by aluminum heat sinks and numerical simulation, Jan. 2024. DOI: 10.1016/j.asej.2023.102330.

[21] E. Roslan and I. Hassim, Solar pv system with pulsating heat pipe cooling, 2019, pp. 311-318. DOI: 10 . 11591/ijeecs.v14.i1.

AUTHOR GUIDELINES

1. Manuscript criteria

- The article has never been published or in the submission process on other publications.
- Submitted articles could be original research articles or technical notes.
- The similarity score from plagiarism checker software such as Turnitin is 20% maximum.
- For December 2021 publication onwards, Ultima Computing : Jurnal Sistem Komputer will be receiving and publishing manuscripts written in English only.

2. Manuscript format

- Article been type in Microsoft Word version 2007 or later.
- Article been typed with 1 line spacing on an A4 paper size (21 cm x 29,7 cm), top-left margin are 3 cm and bottom-right margin are 2 cm, and Times New Roman's font type.
- Article should be prepared according to the following author guidelines in this template. Article contain of minimum 3500 words.
- References contain of minimum 15 references (primary references) from reputable journals/conferences

3. Organization of submitted article

The organization of the submitted article consists of Title, Abstract, Index Terms, Introduction, Method, Result and Discussion, Conclusion, Appendix (if any), Acknowledgment (if any), and References.

- Title
  The maximum words count on the title is 12 words (including the subtitle if available)
- Abstract
  Abstract consists of 150-250 words. The abstract should contain logical argumentation of the research taken, problem-solving methodology, research results, and a brief conclusion.
- Index terms
  A list in alphabetical order in between 4 to 6 words or short phrases separated by a semicolon (;), excluding words used in the title and chosen carefully to reflect the precise content of the paper.
- Introduction
  Introduction commonly contains the background, purpose of the research, problem identification, research methodology, and state of the art conducted by the authors which describe implicitly.
- Method
  Include sufficient details for the work to be repeated. Where specific equipment and materials are named, the manufacturer's details (name, city and country) should be given so that readers can trace specifications by contacting the manufacturer. Where commercially available software has been used, details of the supplier should be given in brackets or the reference given in full in the reference list.
- Results and Discussion
  State the results of experimental or modeling work, drawing attention to important details in tables and figures, and discuss them intensively by comparing and/or citing other references.
- Conclusion
  Explicitly describes the research's results been taken. Future works or suggestion could be explained after it
- Appendix and acknowledgment, if available, could be placed after Conclusion.
- All citations in the article should be written on References consecutively based on its' appearance order in the article using Mendeley (recommendation). The typing format will be in the same format as the IEEE journals and transaction format.

4. Reviewing of Manuscripts

Every submitted paper is independently and blindly reviewed by at least two peer-reviewers. The decision for publication, amendment, or rejection is based upon their reports/recommendations. If two or more reviewers consider a manuscript unsuitable for publication in this journal, a statement explaining the basis for the decision will be sent to the authors within six months of the submission date.

5. Revision of Manuscripts

Manuscripts sent back to the authors for revision should be returned to the editor without delay (maximum of two weeks). Revised manuscripts can be sent to the editorial office through the same online system. Revised manuscripts returned later than one month will be considered as new submissions.

6. Editing References

- **Periodicals**
  J.K. Author, “Name of paper,” Abbrev. Title of Periodical, vol. x, no. x, pp. xxx-xxx, Sept. 2013.

- **Book**
  J.K. Author, “Title of chapter in the book,” in Title of His Published Book, xth ed. City of Publisher, Country or Nation: Abbrev. Of Publisher, year, ch. x, sec. x, pp xxx-xxx.

- **Report**
  J.K. Author, “Title of report,” Abbrev. Name of Co., City of Co., Abbrev. State, Rep. xxx, year.

- **Handbook**
  Name of Manual/ Handbook, x ed., Abbrev. Name of Co., City of Co., Abbrev. State, year, pp. xxx-xxx.

- **Published Conference Proceedings**
  J.K. Author, “Title of paper,” in Unabbreviated Name of Conf., City of Conf., Abbrev. State (if given), year, pp. xxx-xxx.

- **Papers Presented at Conferences**
  J.K. Author, “Title of paper,” presented at the Unabbrev. Name of Conf., City of Conf., Abbrev. State, year.

- **Patents**
  J.K. Author, “Title of patent,” US. Patent xxxxxxx, Abbrev. 01 January 2014.

- **Theses and Dissertations**
  J.K. Author, “Title of thesis,” M.Sc. thesis, Abbrev. Dept., Abbrev. Univ., City of Univ., Abbrev. State, year. J.K. Author, “Title of dissertation,” Ph.D. dissertation, Abbrev. Dept., Abbrev. Univ., City of Univ., Abbrev. State, year.

- **Unpublished**
  J.K. Author, “Title of paper,” unpublished. J.K. Author, “Title of paper,” Abbrev. Title of Journal, in press.

- **On-line Sources**
  J.K. Author. (year, month day). Title (edition) [Type of medium]. Available: http:// www.(URL) J.K. Author. (year, month). Title. Journal [Type of medium]. volume(issue), pp. if given. Available: http://www.(URL) Note: type of medium could be online media, CD-ROM, USB, etc.

7. Editorial Adress

Jl. Scientia Boulevard, Gading Serpong
Tangerang, Banten, 15811
Email: **ultimacomputing@umn.ac.id**

Paper Title

Subtitle (if needed)

Author 1 Name[1], Author 2 Name[2], Author 3 Name[2]
[1] Line 1 (of affiliation): dept. name of organization, organization name, City, Country
Line 2: e-mail address if desired
[2] Line 1 (of affiliation): dept. name of organization, organization name, City, Country
Line 2: e-mail address if desired

Accepted on mmmmm dd, yyyy
Approved on mmmmm dd, yyyy

***Abstract*—This electronic document is a "live" template which you can use on preparing your Ultima Computing paper. Use this document as a template if you are using Microsoft Word 2007 or later. Otherwise, use this document as an instruction set. Do not use symbol, special characters, or Math in Paper Title and Abstract. Do not cite references in the abstract.**

***Index Terms*—enter key words or phrases in alphabetical order, separated by semicolon (;)**

I. INTRODUCTION

This template, modified in MS Word 2007 and saved as a Word 97-2003 document, provides authors with most of the formatting specifications needed for preparing electronic versions of their papers. Margins, column widths, line spacing, and type styles are built-in here. The authors must make sure that their paper has fulfilled all the formatting stated here.

Introduction commonly contains the background, purpose of the research, problem identification, and research methodology conducted by the authors which been describe implicitly. Except for Introduction and Conclusion, other chapter's title must be explicitly represent the content of the chapter.

II. EASE OF USE

A. Selecting a Template

First, confirm that you have the correct template for your paper size. This template is for Ultima Computing. It has been tailored for output on the A4 paper size.

B. Maintaining the Integrity of the Specifications

The template is used to format your paper and style the text. All margins, column widths, line spaces, and text fonts are prescribed; please do not alter them.

III. PREPARE YOUR PAPER BEFORE STYLING

Before you begin to format your paper, first write and save the content as a separate text file. Keep your text and graphic files separate until after the text has been formatted and styled. Do not add any kind of pagination anywhere in the paper. Please take note of the following items when proofreading spelling and grammar.

A. Abbreviations and Acronyms

Define abbreviations and acronyms the first time they are used in the text, even after they have been defined in the abstract. Abbreviations such as IEEE, SI, MKS, CGS, sc, dc, and rms do not have to be defined. Abbreviations that incorporate periods should not have spaces: write "C.N.R.S.," not "C. N. R. S." Do not use abbreviations in the title or heads unless they are unavoidable.

B. Units

- Use either SI (MKS) or CGS as primary units (SI units are encouraged).
- Do not mix complete spellings and abbreviations of units: "Wb/m2" or "webers per square meter," not "webers/m2." Spell units when they appear in text: "...a few henries," not "...a few H."
- Use a zero before decimal points: "0.25," not ".25."

C. Equations

The equations are an exception to the prescribed specifications of this template. You will need to determine whether or not your equation should be typed using either the Times New Roman or the Symbol font (please no other font). To create multileveled equations, it may be necessary to treat the equation as a graphic and insert it into the text after your paper is styled.

Number the equations consecutively. Equation numbers, within parentheses, are to position flush right, as in (1), using a right tab stop.

$$\int_0^{r_2} F(r,\phi)\, dr\, d\phi = [\sigma r_2 / (2\mu_0)] \qquad (1)$$

Note that the equation is centered using a center tab stop. Be sure that the symbols in your equation have been defined before or immediately following the

equation. Use "(1)," not "Eq. (1)" or "equation (1)," except at the beginning of a sentence: "Equation (1) is ..."

D. Some Common Mistakes

- The word "data" is plural, not singular.
- The subscript for the permeability of vacuum μ_0, and other common scientific constants, is zero with subscript formatting, not a lowercase letter "o."
- In American English, commas, semi-/colons, periods, question and exclamation marks are located within quotation marks only when a complete thought or name is cited, such as a title or full quotation. When quotation marks are used, instead of a bold or italic typeface, to highlight a word or phrase, punctuation should appear outside of the quotation marks. A parenthetical phrase or statement at the end of a sentence is punctuated outside of the closing parenthesis (like this). (A parenthetical sentence is punctuated within the parentheses.)
- A graph within a graph is an "inset," not an "insert." The word alternatively is preferred to the word "alternately" (unless you really mean something that alternates).
- Do not use the word "essentially" to mean "approximately" or "effectively."
- In your paper title, if the words "that uses" can accurately replace the word using, capitalize the "u"; if not, keep using lower-cased.
- Be aware of the different meanings of the homophones "affect" and "effect," "complement" and "compliment," "discreet" and "discrete," "principal" and "principle."
- Do not confuse "imply" and "infer."
- The prefix "non" is not a word; it should be joined to the word it modifies, usually without a hyphen.
- There is no period after the "et" in the Latin abbreviation "et al."
- The abbreviation "i.e." means "that is," and the abbreviation "e.g." means "for example."

IV. Using the Template

After the text edit has been completed, the paper is ready for the template. Duplicate the template file by using the Save As command, and use the naming convention as below

ULTIMATICS_firstAuthorName_paperTitle.

In this newly created file, highlight all of the contents and import your prepared text file. You are now ready to style your paper. Please take note on the following items.

A. Authors and Affiliations

The template is designed so that author affiliations are not repeated each time for multiple authors of the same affiliation. Please keep your affiliations as succinct as possible (for example, do not differentiate among departments of the same organization).

B. Identify the Headings

Headings, or heads, are organizational devices that guide the reader through your paper. There are two types: component heads and text heads.

Component heads identify the different components of your paper and are not topically subordinate to each other. Examples include ACKNOWLEDGMENTS and REFERENCES, and for these, the correct style to use is "Heading 5."

Text heads organize the topics on a relational, hierarchical basis. For example, the paper title is the primary text head because all subsequent material relates and elaborates on this one topic. If there are two or more sub-topics, the next level head (uppercase Roman numerals) should be used and, conversely, if there are not at least two sub-topics, then no subheads should be introduced. Styles, named "Heading 1," "Heading 2," "Heading 3," and "Heading 4", are prescribed.

C. Figures and Tables

Place figures and tables at the top and bottom of columns. Avoid placing them in the middle of columns. Large figures and tables may span across both columns. Figure captions should be below the figures; table heads should appear above the tables. Insert figures and tables after they are cited in the text. Use the abbreviation "Fig. 1," even at the beginning of a sentence.

TABLE I. TABLE STYLES

Table Head	**Table Column Head**		
	Table column subhead	***Subhead***	***Subhead***
copy	More table copy		

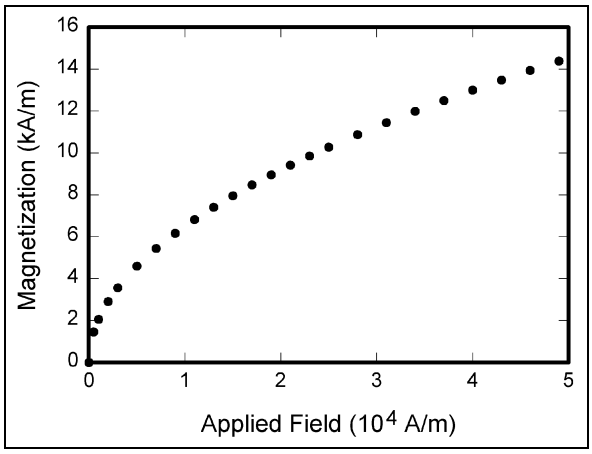


Fig. 1. Example of a figure caption

V. Conclusion

A conclusion section is not required. Although a conclusion may review the main points of the paper, do not replicate the abstract as the conclusion. A conclusion might elaborate on the importance of the work or suggest applications and extensions.

Appendix

Appendixes, if needed, appear before the acknowledgment.

ACKNOWLEDGMENT

The preferred spelling of the word "acknowledgment" in American English is without an "e" after the "g." Use the singular heading even if you have many acknowledgments. Avoid expressions such as "One of us (S.B.A.) would like to thank ... ." Instead, write "F. A. Author thanks ... ." You could also state the sponsor and financial support acknowledgments here.

References

The template will number citations consecutively within brackets [1]. The sentence punctuation follows the bracket [2]. Refer simply to the reference number, as in [3]—do not use "Ref. [3]" or "reference [3]" except at the beginning of a sentence: "Reference [3] was the first ..."

Number footnotes separately in superscripts. Place the actual footnote at the bottom of the column in which it was cited. Do not put footnotes in the reference list. Use letters for table footnotes.

Unless there are six authors or more give all authors' names; do not use "et al.". Papers that have not been published, even if they have been submitted for publication, should be cited as "unpublished" [4]. Papers that have been accepted for publication should be cited as "in press" [5]. Capitalize only the first word in a paper title, except for proper nouns and element symbols.

For papers published in translation journals, please give the English citation first, followed by the original foreign-language citation [6].

[1] G. Eason, B. Noble, and I.N. Sneddon, "On certain integrals of Lipschitz-Hankel type involving products of Bessel functions," Phil. Trans. Roy. Soc. London, vol. A247, pp. 529-551, April 1955. (*references*)

[2] J. Clerk Maxwell, A Treatise on Electricity and Magnetism, 3rd ed., vol. 2. Oxford: Clarendon, 1892, pp.68-73.

[3] I.S. Jacobs and C.P. Bean, "Fine particles, thin films and exchange anisotropy," in Magnetism, vol. III, G.T. Rado and H. Suhl, Eds. New York: Academic, 1963, pp. 271-350.

[4] K. Elissa, "Title of paper if known," unpublished.

[5] R. Nicole, "Title of paper with only first word capitalized," J. Name Stand. Abbrev., in press.

[6] Y. Yorozu, M. Hirano, K. Oka, and Y. Tagawa, "Electron spectroscopy studies on magneto-optical media and plastic substrate interface," IEEE Transl. J. Magn. Japan, vol. 2, pp. 740-741, August 1987 [Digests 9th Annual Conf. Magnetics Japan, p. 301, 1982].

[7] M. Young, The Technical Writer's Handbook. Mill Valley, CA: University Science, 1989.

UMN

UNIVERSITAS
MULTIMEDIA
NUSANTARA

ISSN 2355-3286

UMN
UNIVERSITAS
MULTIMEDIA
NUSANTARA

PRESS

Universitas Multimedia Nusantara
Scientia Garden Jl. Boulevard Gading Serpong, Tangerang
Telp. (021) 5422 0808 | Fax. (021) 5422 0800