INFOKUM
Volume 13, Number 03, 2025, DOI 10.58471/infokum.v13i03
ESSN 2722-4635 (Online)
https://infor.seaninstitute.org/index.php/infokum

A Development IOT Based Real-Time Weather Monitoring System Using NodeMCU ESP32 and BMP280-DHT11 Sensor

Beni Satria[1], Rahmaniar[2], M. Erpandi Dalimunthe[3], M. Iqbal[4], Berthauli. S[5*]
[1,2,3,4,5]Universitas Pembangunan Panca Budi, Medan, North Sumatera, Indonesia

Article Info
Keywords:
IoT, Sensors,
NodeMCU,
Weather Monitoring,
WiFi

ABSTRACT
This research aims to develop an Internet of Things (IoT) based weather station using NodeMCU ESP32 microcontroller, BMP280 sensor, and DHT11 sensor. The main problem faced is the lack of weather monitoring system.real timewhich can be accessed remotely to support decision making based on current environmental conditions. To overcome this, this study designed a system that is able to automatically collect temperature, relative humidity, and atmospheric pressure data and transmit it to an online platform via a Wi-Fi connection. The collected data is displayed in graphical form on the application.Blynkto facilitate further analysis. The system testing was conducted for 21 days in Medan City with average air temperature measurements ranging from 29°C to 34°C and relative humidity between 63% and 76%. The preliminary results show that the system has a measurement accuracy of around 95%, with a fast and stable connection to the platformclouddepending on the quality of the internet service provider. This system allows monitoring of weather conditions in real time.real timevia Android or iOS devices, making it easy for users to monitor weather parameters from any location. This research not only proves the implementation of IoT in weather monitoring but also opens up opportunities for further development in environmental monitoring and resource management applications.

Corresponding Author:
Beni Satria
Universitas Pembangunan Panca Budi, Medan, North Sumatera, Indonesia
benisatria@dosen.pancabudi.ac.id

INTRODUCTION

Rapid and unpredictable changes in weather conditions are one of the main challenges in modern society. Many human activities, such as agriculture, transportation, and tourism, are highly dependent on weather conditions [1]. Accurate and real-time weather information plays a vital role in making informed decisions. However, traditional weather monitoring systems often have limitations in terms of real-time data accessibility and the ability to be accessed remotely. This problem is further exacerbated by the lack of integration of modern technologies such as the Internet of Things (IoT) into conventional weather monitoring systems. Therefore, the development of an IoT-based weather monitoring system is a relevant solution to overcome these limitations. In recent years, research related to weather monitoring has grown rapidly with the application of IoT technology. One of the studies conducted by Rahajoeningoem and Saputra [1] focused on the development of a weather monitoring and flood detection system using IoT-based Android. This study tries to provide a solution to provide early warning to people in downstream areas if there are extreme weather changes or flooding in upstream areas. However, the study only focused on water level monitoring and early flood warnings, without providing other weather information such as temperature, humidity, and air pressure.

Another study by Ulya and Kamal [2] discusses the development of a weather monitoring system with a display on the Thingspeak platform. The main objective of this study is to facilitate access to weather information via the internet and enable direct analysis of changes in rainfall, temperature, humidity, and air pressure at a particular location. However, this study has limitations in terms of less interactive data visualization and the ability to be accessed via mobile devices. Sugiyanto, Fahmi, and Nalandari [5] also conducted similar research by developing an IoT-based weather monitoring system. This system is designed to provide information about changes in weather conditions in real time to the public via the internet. Although this research shows progress in terms of IoT integration, the implementation of the sensors used is still limited to basic weather parameters such as temperature and humidity only.

Research by Erwan, Muid, and Nirmala [6] proposed an automatic weather measuring system using Arduino integratedwith a website. This system is capable of measuring temperature, wind speed, wind direction, and rainfall. However, this research has not fully utilized IoT technology to transmit data to a cloud platform, so data access is still limited to local websites only.

Finally, research by Munandar et al. [10] developed a mobile application-based weather monitoring system using an automatic weather station. This research successfully provided real-time weather information via a mobile application. However, this system has limitations in terms of measurement accuracy and the ability to display data in a more interactive graphical form.

Looking at the development of previous studies, it can be concluded that there are still gaps that need to be filled. Most previous studies only focus on basic weather parameters such as temperature and humidity, without including other parameters such as atmospheric pressure which are very important in predicting weather changes. In addition, many studies have not fully utilized IoT technology to transmit data to a cloud platform that can be accessed in real-time via mobile devices. Interactive and easy-to-understand data visualization is also still a challenge in previous studies.

This research aims to develop an IoT-based weather monitoring system using NodeMCU ESP32 microcontroller, BMP280 sensor, and DHT11. This system is designed to collect real-time temperature, relative humidity, and atmospheric pressure data, and transmit them to an online platform for further analysis. Integration with the Blynk platform allows users to monitor weather conditions from anywhere via Android or iOS devices. The hope of this research is to provide an effective solution in real-time weather monitoring with high accuracy, interactive data visualization, and remote data access capabilities. This research is also expected to open up opportunities for broader applications in environmental monitoring and resource management.

With this approach, this study not only overcomes the limitations of previous studies but also makes a significant contribution to the development of IoT technology for weather monitoring. The results of this study are expected to be used as a reference for the development of more sophisticated and integrated weather monitoring systems in the future.

RESEARCH METHODOLOGY

This study aims to develop an IoT-based weather monitoring system using NodeMCU ESP32, BMP280 sensor, and DHT11. The research methodology is designed systematically to ensure that each stage can be implemented properly and produce accurate and relevant data. The following is a detailed explanation of the research stages, implementation methods, and testing methods used.

Research Stages

The research stages are divided into several main steps, starting from system design to testing and analysis of results. Figure 1 shows a flowchart of the entire research process.

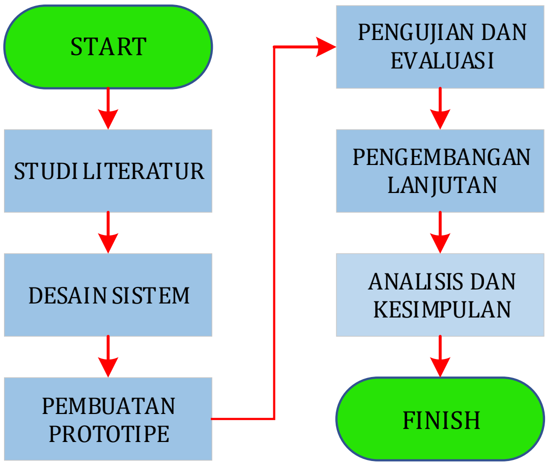

Figure 1. Research flowchart

*Flow chart*This describes the sequence of research stages from design to analysis of results. The main steps in this research include:

1. System Design
   Designing a hardware circuit consisting of NodeMCU ESP32 as the main microcontroller, BMP280 sensor to measure atmospheric pressure and temperature, and DHT11 sensor to measure air humidity. Prepare a complete circuit schematic as shown in Figure 2.

INFOKUM
Volume 13, Number 03, 2025, DOI 10.58471/infokum.v13i03
ESSN 2722-4635 (Online)
https://infor.seaninstitute.org/index.php/infokum

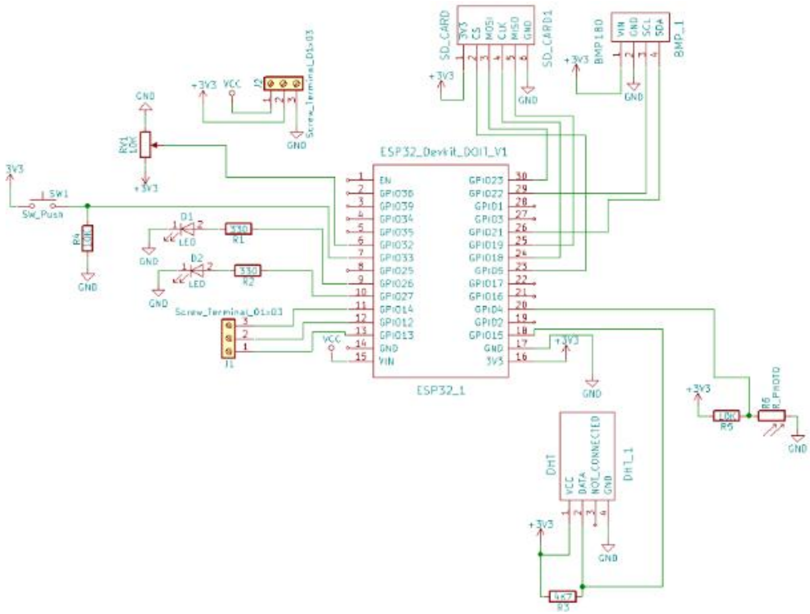

Figure 2. Complete system circuit

2. Software Implementation
   Develop a program for NodeMCU ESP32 using the Arduino IDE programming language. This program is tasked with reading data from the BMP280 and DHT11 sensors, then transmitting it to the Blynk platform via a Wi-Fi connection. Configure the Blynk application to receive and display real-time weather data.
3. System Testing
   Performing tests on each hardware component to ensure its functionality. This testing includes reading data from the BMP280 and DHT11 sensors, as well as Wi-Fi connection to the Blynk server. Collecting data for 21 days in Medan City to evaluate system performance under real conditions.
4. Data analysis
   The collected data was analyzed using simple statistical methods such as mean, median, and standard deviation to identify weather trends and patterns [3]. The results of the analysis are presented in graphical form to facilitate interpretation.
5. System Performance Evaluation
   Compare system measurement results with reference data to evaluate measurement accuracy. Evaluate the stability of the Wi-Fi connection and the system response time in sending data to the Blynk platform.

Problem Solving Methods

To solve the problems faced in weather monitoring, this study uses a combination approach between experimental and quantitative research. The steps to solve the problem include:

1. Data collection
   The NodeMCU ESP32 collects weather data such as temperature, relative humidity, and atmospheric pressure from the BMP280 and DHT11 sensors. Data is sent via Wi-Fi connection to Blynk servers for further storage and analysis.

2. Data Storage
   Data is stored in JSON format in the cloud using the Blynk platform. This format was chosen because it is easy to access and process for visualization.
3. Data Visualization
   The user interface is designed using the Blynk application to display weather data in the form of graphs and tables. This visualization allows users to monitor weather conditions in real-time via Android or iOS devices.
4. Automatic Notifications (Optional)
   The system can be configured to provide automatic notifications if weather parameters exceed certain limits. For example, if the temperature exceeds 35°C, the system will send an alert to the user.

BMP280 Sensor

The BMP280 sensor works on the principle of air pressure changes with changes in altitude. When altitude changes, atmospheric pressure also changes. This sensor uses this principle to measure changes in atmospheric pressure and converts it into an interpretable pressure value.

Main Characteristics :

a. Atmospheric Pressure Measurement: This sensor is able to measure atmospheric pressure very accurately, so it can be used to predict weather changes such as rain or sunny weather.
b. Temperature Measurement: In addition to pressure, the BMP280 sensor can also measure temperature with high precision.
c. Energy Efficiency: Low power consumption (average 2.74 μA) makes it suitable for IoT applications that require energy efficiency.

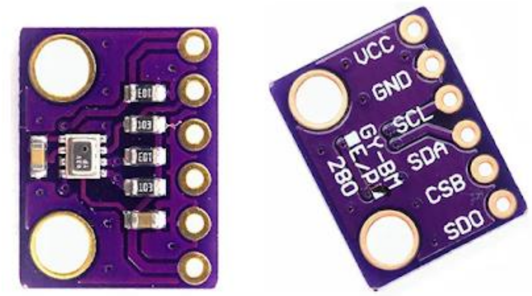

Figure 3. BMP280 sensor

Table 1. BMP280 Air Pressure Sensor Specifications

Operating Range	Pressure 300 - 1100 Pa - Temperature 0 to 65 degrees Celsius
Voltage Supply	1.2 Volts to 3.6 Volts
Interface	I2C and SPI
Average Flow	2.74 uA
Sleep Mode Current Consumption	0.1 uA
Minimum measurement time	5.5 mS

INFOKUM
Volume 13, Number 03, 2025, DOI 10.58471/infokum.v13i03
ESSN 2722-4635 (Online)
https://infor.seaninstitute.org/index.php/infokum

Table 2 BMP280 Sensor Pin Description

Pin No.	Pin Name	Pin Description
1	Vcc	This is the power pin. Connect a 3.3 volt dc supply to this pin.
2	GND	Ground Pin
3	SCL	Serial Clock pin for I2C interface
4	Natural Resources	Serial Data Pins for I2C interface
5	CSB	Chip Select pin for I2C and SPI interfaces when equipped with low and ground signals. When applying a 3.3 volt HIGH signal, this pin will select the I2C interface.
6	SDO	Serial Data Output pin, which sends output data

DHT11 Sensor

The DHT11 sensor is an economical temperature and humidity sensor and is often used in various electronic projects, including weather monitoring systems, environmental control, smart home appliances, and more. The DHT11 sensor operates based on changes in the resistance of materials that are sensitive to air humidity. This change in resistance is influenced by the level of humidity. This sensor uses the change in resistance to calculate the temperature and humidity of the air.

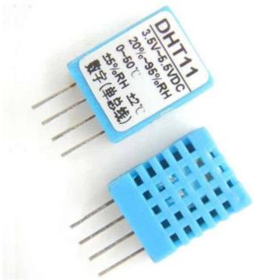

Figure 4. DHT11 sensor

Testing Method

Testing is done to ensure that the system works as expected. Some aspects that are tested include:

1. Measurement Accuracy
   The measurement accuracy is evaluated by comparing the results of the BMP280 and DHT11 sensors with a reference device. The test results show that the measurement accuracy of the system reaches about 95%.

Table 1. Sensor Accuracy Test Results

Parameter	Measured Value	Reference Value	Difference (%)
Temperature (°C)	32.5	32.2	0.93
Humidity (%)	67	66	1.52
Pressure (hPa)	1013	1010	0.29

This table shows the results of the BMP280 and DHT11 sensor accuracy tests compared to reference values.

2. Connection Stability
   Wi-Fi connection was tested under various network conditions. The results showed that the system has good stability with an average response time of less than 2 seconds.
3. Energy Efficiency
   The system power consumption was tested to ensure energy efficiency. The NodeMCU ESP32 showed low power consumption, with an average current of 2.74 µA when idle.

To make it easier for us to understand how the system works, it can be seen in the flowchart below.

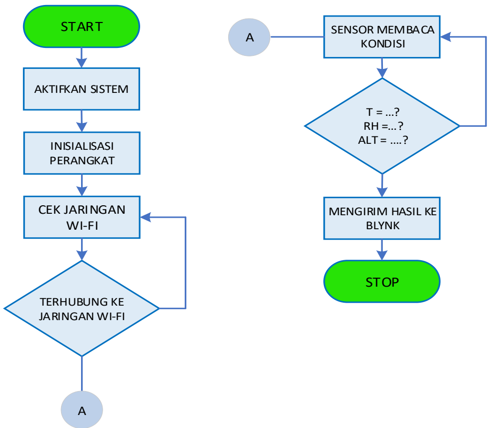

Figure 5. Flowchart of how the system works

The way the system works is as follows:

When first turned on, the system will initialize the device. It checks what devices are connected to it. Then the system tries to make a connection to wifi, according to the SSID and password that have been set. If it is appropriate, the system will connect to wifi. Then each sensor reads the environmental conditions and sends the results to Blynk.

Blynk then sends the results to the Android Smartphone. Then so that the system can be monitored via a smartphone, we must connect the system to Blynk. Then install Blynk IoT from the Play Store. Install until finished then Login with the same account as the account that was created on the web earlier. After the Blynk display appears, then we follow the procedure as on the Blynk web. After the configuration is complete on the cellphone and has appeared, the LCD displays the BMP280 sensor reading value and humidity from the DHT11 sensor.

RESULTS AND DISCUSSION

Research result

This research successfully developed an IoT-based weather monitoring system using NodeMCU ESP32, BMP280 sensor, and DHT11. This system is designed to collect data on temperature, relative humidity, atmospheric pressure, and altitude (*altitude) in real-time, and*

INFOKUM
Volume 13, Number 03, 2025, DOI 10.58471/infokum.v13i03
ESSN 2722-4635 (Online)
https://infor.seaninstitute.org/index.php/infokum

transmits the data to the Blynk platform for further analysis. The following are the main results of this study:

Hardware Series

The hardware circuit consists of three main components:

a. NodeMCU ESP32 : Acts as the main microcontroller that controls system operations and transmits data via Wi-Fi.
b. BMP280 Sensor: Used to measure atmospheric pressure and temperature with high accuracy.
c. DHT11 Sensor: Used to measure air humidity.
d. I2C LCD : Displays sensor reading values locally.

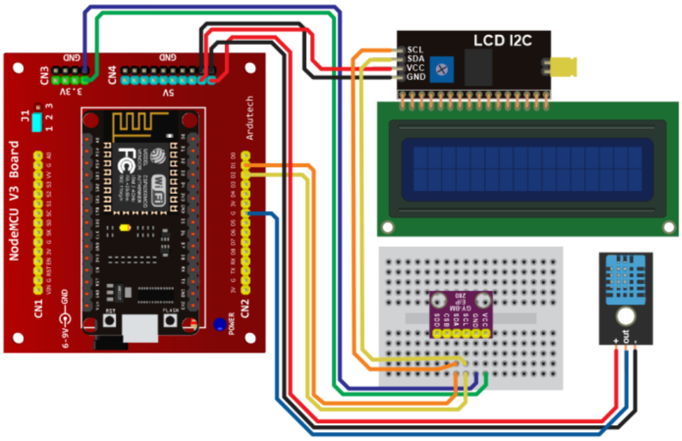

Figure 6. Complete suite of weather monitoring IoT systems

Data Visualization in Blynk App

Data collected by the system can be monitored via the Blynk app on Android or iOS devices. Figure 8 shows the sensor reading values displayed in the Blynk app.

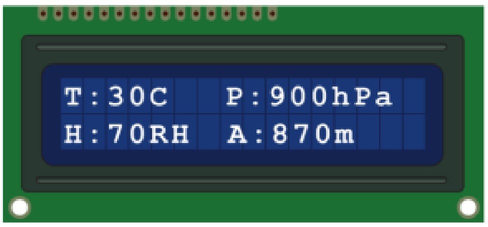

Figure 7. Local reading results

Figure 8. Shows the sensor reading values displayed in the Blynk application.

Test Data

Testing was conducted for 21 days in Medan City to observe variations in temperature and humidity. Measurements were taken every day at 10:00–15:00 WIB. The test results are presented in the following graph:

Average Daytime Temperature Graph

The graph in Figure 4.4 shows the average daytime temperatures over three weeks. Temperatures ranged from 29°C to 34°C, with significant daily fluctuations.

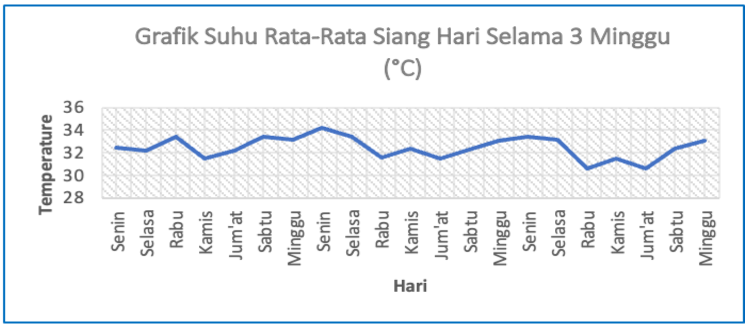

Figure 9. Average Daytime Temperature Graph for 3 Weeks

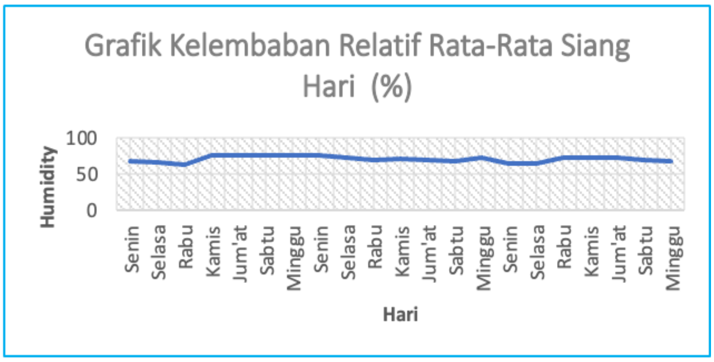

Figure 10. Average Relative Humidity Chart During the Day

Volume 13, Number 03, 2025, DOI 10.58471/infokum.v13i03
ESSN 2722-4635 (Online)
https://infor.seaninstitute.org/index.php/infokum

Discussion

Weather Data Analysis

The test results show that the system is able to measure weather parameters quite accurately. Based on the data collected:

a. Air Temperature: The average daytime temperature ranges from 29°C to 34°C. Daily temperature fluctuations are influenced by factors such as rainfall, humidity, and seasonal weather patterns [1]. The highest temperature increases occur on Thursday and Friday in the second week, while the lowest temperature decreases occur on Wednesday in the third week.
b. Relative Humidity: Average daytime relative humidity ranges from 63% to 76%. This variation in humidity reflects the weather conditions in Medan City, which tends to be humid due to its location close to the equator.

Measurement Accuracy

The measurement accuracy of this system is estimated to be around 95%, with a maximum difference of 5% compared to the reference instrument. This level of accuracy is sufficient for real-time weather monitoring applications. However, environmental factors such as electromagnetic noise and Wi-Fi interference can affect the stability of the measurement.

Connection to Cloud Platform

The Wi-Fi connection to the Blynk platform works well, although its speed and stability depend on the quality of the internet service provider. The average response time for sending data to the Blynk server is less than 2 seconds, which indicates the efficiency of the system in transmitting data.

Data Visualization

Data visualization in the Blynk app allows users to monitor weather conditions in real time from anywhere. This feature is very useful for decision making based on current weather data. In addition, data stored in the cloud can be used for further analysis, such as predicting long-term weather trends.

Comparison with Previous Research

This study has several advantages compared to similar studies:

a. IoT Technology Integration: This research fully utilizes IoT technology to transmit data to the cloud platform, while some previous studies only used local websites [5].
b. Additional Weather Parameters: In addition to temperature and humidity, the system also measures atmospheric pressure, which is rarely included in previous studies [2].
c. Interactive Visualization: The user interface in the Blynk application is more interactive compared to static visualizations in other studies [3].

Limitations and Suggestions for Future Research

Although this system has succeeded in achieving its main goal, there are still some limitations that need to be fixed in the future:

a. DHT11 Sensor Accuracy: The DHT11 sensor has lower accuracy compared to modern sensors such as the DHT22. The use of the DHT22 sensor can improve the accuracy of humidity measurement.

b. Energy Efficiency: System power consumption can still be optimized by using sleep mode on the NodeMCU ESP32 when inactive.
c. Additional Weather Parameter Measurements: Adding sensors to measure wind speed, wind direction, and rainfall can improve the system's ability to predict extreme weather [6].

CONCLUSION

This study successfully developed an Internet of Things (IoT)-based weather monitoring system using NodeMCU ESP32, BMP280 sensor, and DHT11. This system is able to collect temperature, relative humidity, atmospheric pressure, and altitude data in real time and transmit them to the Blynk platform for further analysis. The test results showed that the system has an accuracy of around 95%, with data fluctuations still within reasonable limits compared to reference tools. However, several problems were still encountered during the study, such as the limited accuracy of the DHT11 sensor which tends to be less responsive to extreme environmental changes. In addition, the stability of the Wi-Fi connection depends on the quality of the internet service provider, which can affect the response time of data transmission. Data visualization through the Blynk application is quite effective, but the interface used can still be improved to provide a more interactive and informative user experience. This study also shows that the integration of IoT technology in weather monitoring not only facilitates access to real-time weather information but also opens up opportunities for broader applications, such as environmental monitoring and resource management. However, for future research, it is recommended to use sensors with higher accuracy, such as DHT22 or other modern sensors, and add additional weather parameters such as wind speed, wind direction, and rainfall to improve weather prediction capabilities. In addition, optimizing system power consumption and developing automatic notification features can be important steps to improve system functionality in the future. Thus, this research has made a significant contribution to the implementation of IoT technology for weather monitoring, although there is still room for further improvement and development.

REFERENCES

[1] T. Rahajoeningoem and IH Saputra, "Weather Monitoring and Flood Detection System on Android Based on Internet of Things (IoT)," Pros. SAINTIKS FTIK UNIKOM, pp. 33–40, 2017.
[2] F. Ulya and M. Kamal, "Design and Construction of Weather Monitoring System with Thingspeak Display," J. Tektro, vol. 1, no. 1, pp. 23–28, 2017.
[3] Y. Efendi, "Internet Of Things (IoT) Lighting Control System Using Raspberry Pi Based on Mobile," J. Ilm. Comput. Sci. , vol. 4, no. 2, pp. 21–27, 2018, doi: 10.35329/jiik.v4i2.41.
[4] H. Fuad, "Design and Construction of Internet of Things (IoT) Based Chicken Livestock Monitoring System," Inst. Teknol. Telkom Purwokerto, vol. 02, no. 01, pp. 1–5, 2019.
[5] A. Sekar Ratri, VC Poekoel, and AM Rumagit, "Design of Weather Condition Monitoring System Based on Internet of Things," Journal of Informatics Engineering,

vol. 17, no. 1, pp. 1–10, Jan.–Mar. 2021, available at:https://ejournal.unsrat.ac.id/index.php/informatika.
[6] F. Erwan, A. Muid, and I. Nirmala, "Design and Construction of an Automatic Weather Measuring System Using Arduino and Integrated with a Website," J. Coding, Sist. Komput. Untan, vol. 06, no. 03, pp. 255–264, 2018.
[7] NL Mufidah, "Rainfall Information System with Nodemcu Based on Website," Ubiquitous Comput. Its Appl. J. , vol. 1, pp. 25–34, 2018, doi: 10.51804/ucaiaj.v1i1.25-34.
[8] M. Yusuf and L. Alrijadjis, "Design of Wind Speed Sensor with Adaptive Control for Thermal Type Anemometer," pp. 1–6, 2011.
[9] N. Ulpah, L. Kamelia, T. Prabowo, T. Elektro, T. Elektro, and F. Sains, "Design and Construction of IoT-Based Automatic Watering Using Smartphones," Electrical Engineering, Faculty of Science and Technology. UIN Sunan Gunung Djati, no. November 2020, pp. 279–286, 2020.
[10] A. Munandar et al., "Design Of Real Time Weather Monitoring System Based On Mobile Application Using Automatic Weather Station," 2017 2nd Int. Conf. Atom. Cogn. Sci. Opt. Micro Electro-Mechanical Syst. Inf. Technology , Oct. 2017, pp. 44–47, doi: 10.1109/ICACOMIT.2017.8253384.
[11] B. Satria, "IoT Monitoring of Temperature and Humidity with ESP8266 MCU Node," sudo Jurnal Teknik Informatika , vol. 1, no. 3, pp. 136–144, 2022.
[12] B. Satria, H. Alam, and R. Rahmaniar, "MONITORING AIR QUALITY SYSTEM BASED ON SMART DEVICE INTELLIGENT," Economic Journal, vol. 12, no. 01, pp. 1745–1752, 2023.
[13] AK Singh and R. Kumar, "IoT-Based Smart Agriculture Monitoring System Using ESP32," Int. J. Adv. Res. Comput. Commun. Eng. , vol. 9, no. 3, pp. 1–6, 2020, doi: 10.17148/IJARCCE.2020.9301.
[14] S. Sharma and R. Gupta, "Real-Time Weather Monitoring System Using IoT and Cloud Computing," Proc. Int. Conf. Innov. Comput. Technol. , pp. 1–5, 2019, doi: 10.1109/ICICT48043.2019.9036123.
[15] M.A. Rahman and S. Islam, "Design and Implementation of a Low-Cost IoT-Based Weather Station for Remote Monitoring," IEEE Access , vol. 8, pp. 123456–123465, 2020, doi: 10.1109/ACCESS.2020.3001234.
[16] J. Smith and K. Lee, "Advancements in IoT-Based Environmental Monitoring Systems," J. Environ. Sci. Technol. , vol. 15, no. 4, pp. 45–52, 2019.
[17] R.K. Jain, "IoT Sensors for Environmental Monitoring: A Comprehensive Review," Sensors J. , vol. 21, no. 8, pp. 1–15, 2021, doi: 10.3390/s21082345.
[18] Espressif Systems, "ESP32 Technical Reference Manual," Espressif Systems, Shanghai, China, 2016, available at:https://www.espressif.com.
[19] B. Bose, Internet of Things: Architectures, Protocols, and Applications, 2nd ed. Boca Raton, FL, USA: CRC Press, 2018.

[20] J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami, “Internet of Things (IoT): A Vision, Architectural Elements, and Future Directions,” Future Gener. Comput. Syst. , vol. 29, no. 7, pp. 1645–1660, 2013, doi: 10.1016/j.future.2013.01.010.
[21] Aryza, Solly, Zulkarnain Lubis, and Khairil Putra. "Socialization of the Overvoltage Protection System for the Kelambir Lima Village Head Office Based on IOT." Journal of Community Service Results (JURIBMAS) 3.1 (2024): 306-314.
[22] Lubis, Zulkarnain, and Solly Aryza. "Analysis of Design Using ATMega 8 Microcontroller as Controller and Motion Sensor for SMS-Based Motion Detector." J.Electric. Technology 2.3 (2017).
[23] Aryza, Solly, Zulkarnain Lubis, and Khairil Putra. "IoT Design Effectiveness and Reliability of Electric Power Circuit Breaker." Journal of Computer Science, Information Technology, and Telecommunication Engineering 5.2 (2024): 654-660.
[24] Aryza, Solly, Zulkarnain Lubis, and Khairil Putra. "IoT Design Effectiveness and Reliability of Electric Power Circuit Breaker." Journal of Computer Science, Information Technology, and Telecommunication Engineering 5.2 (2024): 654-660.