

Dashboard Design of Automatic Weather Station Measurement in Yogyakarta Climatology Station using Node-Red and Grafana

Alfin Syarifuddin Syahab¹

¹Climatology Office of Yogyakarta, Indonesian Agency for Meteorology Climatology and Geophysics, Yogyakarta, Indonesia

*Corresponding author: alfin.syahab@bmkg.go.id

ABSTRACT

The Indonesian Agency for Meteorology Climatology and Geophysics (BMKG) has installed a weather parameter measurement network using a sensor network called the Automatic Weather Station (AWS). The equipment records weather parameters such as air temperature, relative humidity, atmospheric pressure, solar radiation intensity, rainfall, wind speed, and wind direction. At the BMKG regional unit, automatic weather observation systems are not yet fully utilized due to limited network infrastructure and inadequate supporting software. Moreover, local stations lack real-time monitoring tools for equipment conditions, as existing national systems remain general and insufficiently detailed. The AWS measurement display is needed for forecasters to validate extreme weather and climate events. In addition, real-time data is used by technicians as a tool performance check and data quality control. Therefore, the study aims to design an interactive display on AWS to support technicians in monitoring equipment conditions and assist forecasters in reporting extreme weather and climate events as a service to the community. The data visualization display system is designed using the Node-Red and Grafana programming tools connected to the MySQL database management system. The results show that Node-Red can display weather data with one-minute intervals for seven AWS locations in the Special Region of Yogyakarta in the form of graphs and gauges. Meanwhile, Grafana can display weather parameters in the form of gauges and graphs at one point of the Yogyakarta Special Region Climatology Station.

Keywords: Automatic Weather Station, Dashboard, MySQL, Node-Red, Grafana.

1. INTRODUCTION

The Meteorology, Climatology, and Geophysics Agency (BMKG) plays a vital role in observing and monitoring weather and climate conditions across Indonesia (Fatoni et al., 2024). These services include the distribution of public and special information to the government and the community, in order to support safety and environmental sustainability (Marby et al., 2025). BMKG receives weather information from monitoring stations located across the country and gathers data using sensors, records it using data loggers, and sends this information via GSM (Global System for Mobile Communications) networks to the server (Widayanti & Fajarrohman, 2024).

Sensor networks are pivotal in environmental data collection, enabling BMKG to analyze weather patterns and predict natural disasters. Integrating internet of things (IoT) technology enhances these systems' efficiency and accuracy, making them essential for decision-making in sectors affected by climate change. IoT-based sensor networks facilitate real-time data collection and analysis for monitoring weather parameters (Ashfaq & Nur, 2024) (Ogunbunmi et al., 2024). Automatic Weather Stations (AWS) are automated systems that monitor various meteorological parameters, such as atmospheric pressure, rainfall, relative humidity,

temperature, wind speed, and wind direction. BMKG currently operates 368 AWS units across Indonesia, both within and outside station units (Santoso et al., 2023).

Previous research has designed a climate data monitoring system with the results being able to send weather sensor data every five minutes and is only limited to one site in Sumberbrantas village (Muslim et al., 2021). Another study published in Nigeria, the design of a weather monitoring system was carried out with a simpler number of weather parameter sensors and successfully sent data every 10 minutes (Adenekan et al., 2021). However, due to the complexity of information systems, including climate and weather monitoring, it is now more difficult to fully convey information to others because it is necessary to achieve consistency and maintain the order of this information (Mahdi et al., 2024). According this case, integrating Node-Red as a visualization platform can enhance the efficacy of monitoring and user-friendliness of data access. Even in a variety of environmental circumstances, the system consistently provides accurate and efficient temperature information. Additionally, this system's lightweight data administration, remote monitoring capabilities, and ease of integration make it a useful tool for supporting environmental condition control (Utomo & Izzaturrahmani, 2025). Moreover, Grafana is used as the front-end of the visualization application system because of its great processing flexibility (Anam et al., 2023). This helps reduce reliance on the national BMKG system, which contains too much data for local use.

At the Yogyakarta climatology station, automatic weather and climate observation systems have not been fully implemented by observers. The lack of adequate network infrastructure and supporting software is a factor hindering automated and digital observations. In addition, there is no real-time equipment condition monitoring system available within the scope of local observation stations outside the national system, which is still general and less informative. Currently, the monitoring system only shows data from the AWS at the Yogyakarta Climatology Office. By using Node-Red, data from 7 AWS sites across Yogyakarta Province can be collected.

This paper discusses the design and implementation of a weather monitoring system using an MQTT broker connected to Node-Red to collect data from several AWS sites. It also uses Grafana to display AWS data at the Yogyakarta Climatology Station in a more detailed and interactive way, with real-time updates every minute shown in numbers and graphs. This system aims to help technicians monitor equipment conditions and support observers and forecasters in validating extreme weather events.

2. METHODS

This section displays the main menu when the program is run on the Node-Red application, which will display the data visualization dashboard design process using nodes that are interconnected using the Javascript language. some main stages contain; publish and subscribe data, parse data, connect to MySQL, and connect to Grafana. Figure 1 shows block diagram of AWS data monitoring system.

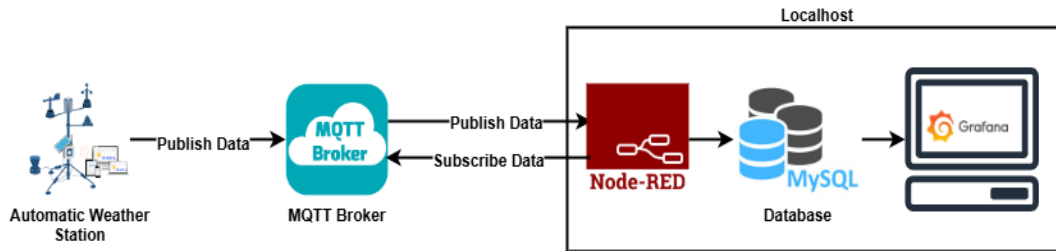


Figure 1. Block Diagram of AWS Data Monitoring System

The figure shows the data flow architecture of an automatic weather monitoring system of AWS. Weather data collected by the sensor is published to the MQTT Broker, a lightweight communication protocol commonly used in IoT systems. Furthermore, Node-Red acts as middleware that subscribes to data from the MQTT Broker, processes it, and then stores it in a MySQL database running on the localhost. The stored data can be visualized in real-time through Grafana, a visualization platform that allows users to monitor weather data interactively and informatively. Grafana obtains its data from the MySQL database, which contains information collected through client subscriptions to MQTT topics handled by the Node-Red application (Aditama et al., 2023).

One of Node-Red's key strengths lies in its visual programming approach, which lets users design and understand the flow of data through connected nodes, each with a clearly defined function (Omidi et al., 2023). Companies may visualize data taken from real-time object data, including sensor data, and turn it into information to help with decision-making with the help of Node-Red and Grafana which can improve the use of operational resources (Anam et al., 2023). The software used to design this system includes Node-RED v4.0.5, Node.js v22.11.0, and Grafana v11.5.1. AWS data is handled by seven sites spread across districts. Recorded sensor data is stored in a data logger with settings for an average of 10 minutes to the server. Table 1 shows AWS locations installed in Yogyakarta.

Table 1. The List of AWS Locations Installed in Yogyakarta Province

Station Name	ID Station	Location
AWS Staklim Mlati	160021	Mlati
AWS Stageof Yogya	10003	Gamping
AWS Kulon Progo	STA2061	Wates
AWS SMPK Kulon Progo	STA4014	Nanggulan
AWS Pakem	STA2095	Pakem
AWS Prambanan	STA2139	Prambanan
AWS UGM	160016	Depok

The data is transmitted to the server via FTP (File Transfer Protocol), and is also accessible through an MQTT (Message Queuing Telemetry Transport) broker, with an average interval of one minute between data transmissions. Real-time AWS data is important for technicians to check for sensor errors and monitor battery conditions. For weather observers, AWS displays help provide up-to-date weather information and confirm extreme weather events quickly. Next, there is Figure 2 to display flowchart of AWS Data Monitoring System.

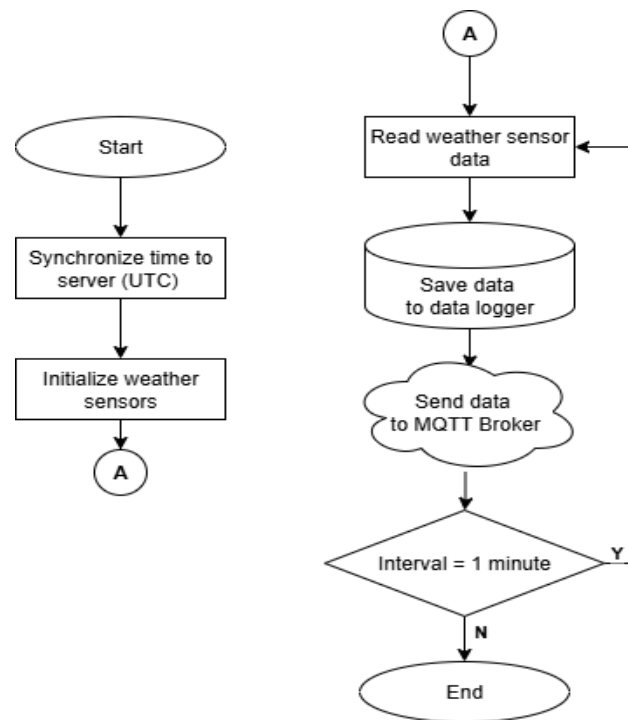


Figure 2. Flowchart of AWS Data Monitoring System

This flowchart illustrates the workflow of the AWS data monitoring system in the Special Region of Yogyakarta which functions to read, record, and send data from sensors to the server automatically and periodically. The process begins with system initialization through the Start step, followed by local time synchronization to the server so that data recording is accurate and consistent, especially using the UTC time reference. After that, the system initializes all sensors to be ready to read environmental data such as temperature, relative humidity, pressure, wind direction, wind speed, solar radiation, and rainfall. The data obtained from the sensors is then read and stored in the data logger as local storage. Furthermore, the data that has been stored is sent to the central server using the MQTT Broker communication protocol which is widely used in IoT systems. After the data sending process, the system will check whether the time interval has reached 1 minute. If so, the system will repeat the process of reading and sending sensor data. However, if not, the system will wait until the 1-minute time is met before repeating the process. This process continues continuously periodically, unless an emergency stop condition occurs, which is marked with the End block. Overall, this flowchart represents an automated monitoring system that runs in real-time, ensuring that environmental data is recorded and transmitted consistently every minute. To secure access in Node-RED, the settings.js file within the C directory is configured to enable adminAuth. In this file, a username and password are defined to control access to both the node flow editor and the user interface dashboard.

3. RESULTS AND DISCUSSIONS

3.1. NODE-RED DASHBOARD

The first stage in designing an AWS weather data monitoring system is to create a publish and subscribe data menu. Using inject node on Node-Red which is simulated as Data Logger

connected to MQTT out node as broker. In this process, it is called publish data from sensor data logger to MQTT server. The server targeted in this system is BMKG's central server for weather sensor data and credential access for users of the server data. Figure 3 depicts nodes to create subscriber function.

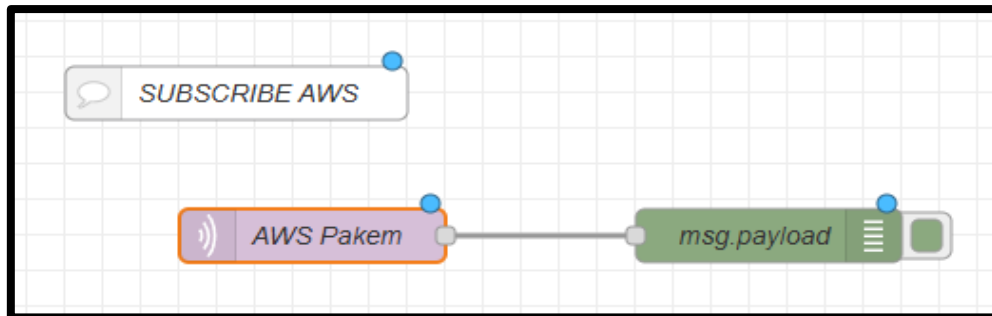


Figure 3. Subscribing Weather Data

The process of transmitting weather observation data to a database server via the MQTT protocol involves initializing a device topic that encodes the site's identity. This identity includes the device type, data logger type, province name, and device serial number, enabling unique identification of each observation station across the monitoring network. To verify that data has been successfully published through the MQTT broker, the system can be connected to a debug node, which allows real-time inspection of the data being transmitted at one-minute intervals. Each transmitted data packet contains detailed information, including the date and time of observation, site identification number, data logger temperature, battery voltage, and key meteorological parameters. These parameters consist of air temperature, relative humidity, air pressure, rainfall, solar radiation, wind speed, and wind direction. This structured approach ensures data traceability and supports reliable communication between devices in real-time weather monitoring systems.

The subsequent stage in the design of an AWS data monitoring system involves the development of a parsing mechanism to extract and visualize weather data on a dashboard interface. This process begins by configuring a JSON node to convert incoming data from a string format into a structured object. Following this, function nodes are implemented, each configured with multiple outputs to correspond to the various parameters provided by the AWS. A custom script, developed using JavaScript within the Node-RED environment, is then employed to specify which output parameters are to be displayed on the dashboard. These may include key weather variables such as air temperature, relative humidity, and rainfall recorded at the AWS site. To ensure the accuracy of the data parsing process, a debug node is utilized, allowing the developer to verify that each output is correctly interpreted and mapped. This process facilitates a clear and organized presentation of real-time weather data, as illustrated in Figure 4, which shows the parsed output of various weather parameters.

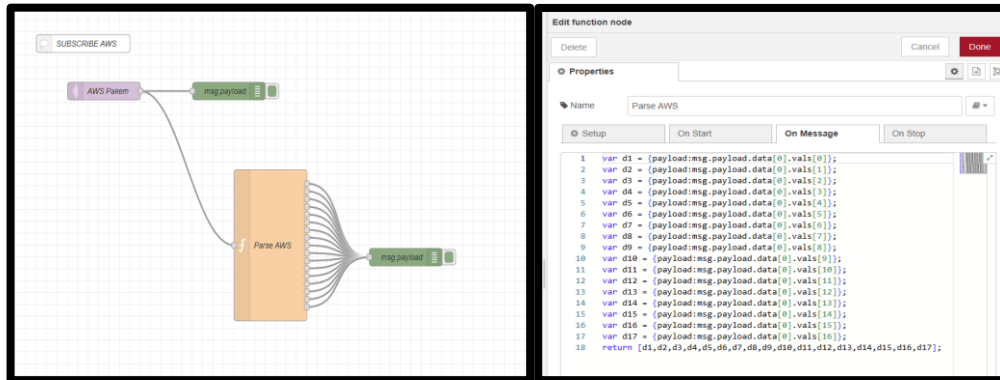


Figure 4. Parsing Data of Weather Parameters

The parsed data is then visualized using the dashboard palette feature in Node-RED, which enables the display of information as graphs, numerical values, text, and gauges based on user requirements. Each site display includes the site name, date and time in text format, a gauge for rainfall, and a graph for battery voltage. Figure 5 presents the complete set of nodes used in the AWS monitoring system.

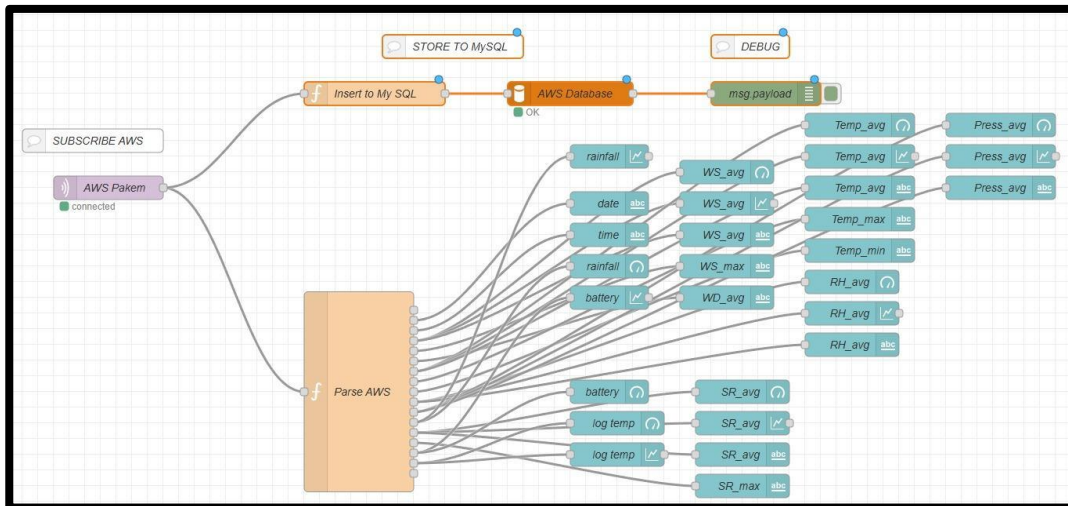


Figure 5. Nodes AWS Monitoring Dashboard using Node-Red

In developing an AWS data monitoring system on Node-Red, this research connects nodes, each with its own function. Starting with subscribed data, the node uses the data parse function to read weather sensor and device battery data. The data parse function allows data to be displayed using nodes for graphical, gauge, or text displays to provide detailed information on the display. Figure 6 displays user interface of rainfall data from rain gauge sensors in AWS network in Yogyakarta.



Figure 6. Display Data of Rain gauge sensors in AWS network in Yogyakarta

The design results show that rainfall data can be displayed in the Node-Red user interface, using a gauge feature that has a measurement range of 0 to 200 millimeters for rainfall units. Furthermore, a graph feature is added to facilitate viewing rainfall based on the time of occurrence. Detailed date and time information is also displayed in the user interface. Figure 7 shows user interface of temperature and humidity sensors from AWS network in Yogyakarta.

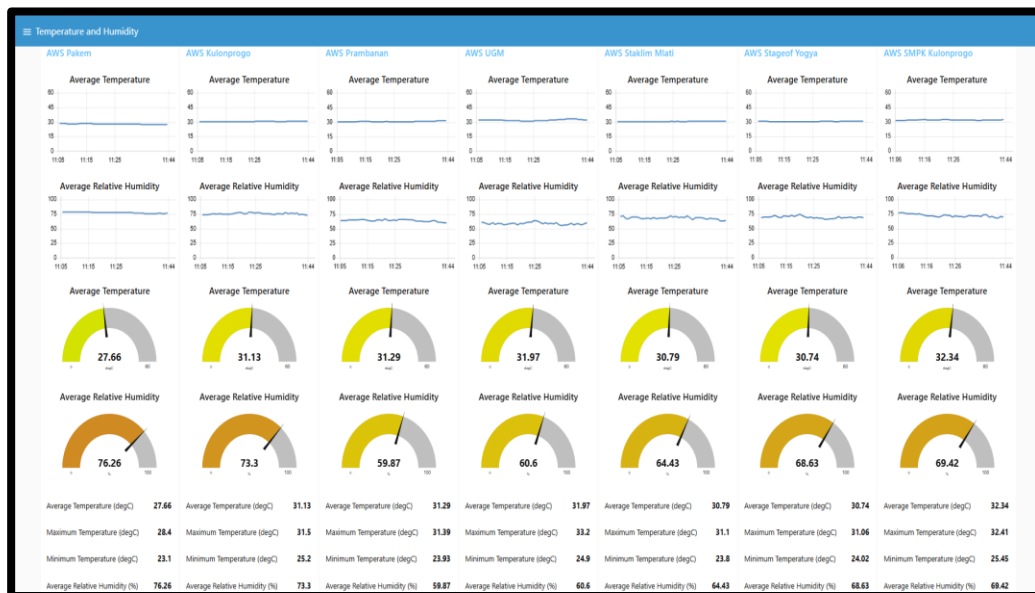


Figure 7. Display Data of Temperature and Humidity Sensors from AWS network in Yogyakarta

The resulting nodes in the user interface display graphs and gauges for air temperature in degree of celsius and relative humidity in percentage measurements. Additionally, there is additional information in the form of maximum and minimum temperatures. Figure 8 give information about display data of wind speed and direction sensors from AWS network in Yogyakarta.



Figure 8. Display Data of Wind Speed and Direction Sensors from AWS network in Yogyakarta

The wind speed and wind direction measurements with meter per second unit were obtained from an anemometer sensor mounted at a height of 10 meters on the AWS equipment. The parameters obtained from the sensor were wind speed, maximum wind speed, and wind direction, which were successfully displayed in the Node-Red user interface. Figure 9 provides the result in user interface of air pressure sensors from AWS network in Yogyakarta.

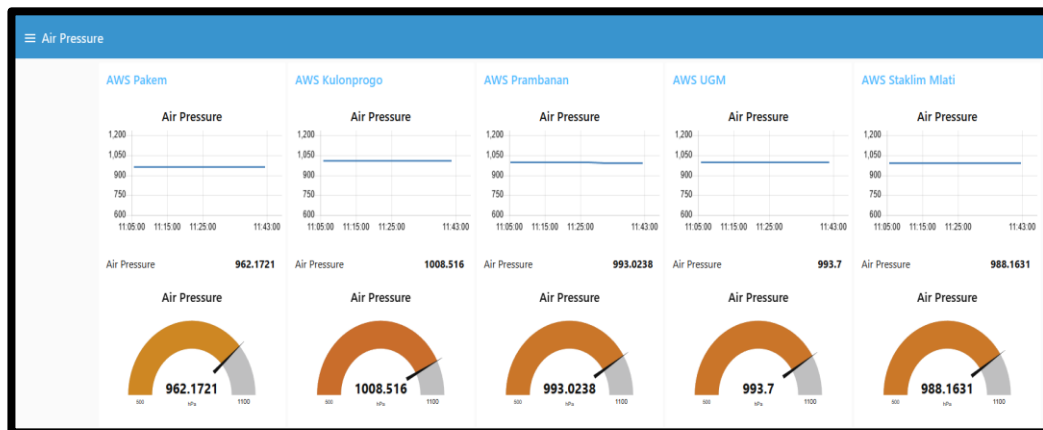


Figure 9. Display Data of Air Pressure of AWS network in Yogyakarta

The results of air pressure measurements from the barometer sensor can be displayed in the Node-Red monitoring system in the form of graphs and gauges. Figure 10 displays data of solar radiation sensors from AWS network in Yogyakarta.



Figure 10. Display Data of Solar Radiation of AWS network in Yogyakarta

In measuring solar radiation, the unit used is W/m^2 . The parameters that can be displayed on the user interface are the solar radiation value and the maximum solar radiation value in the form of graphs and gauges. Figure 11 depicts user interface of battery and logger temperature data sensors from AWS network in Yogyakarta.



Figure 11. Display Data of Battery and Logger Temperature of AWS network in Yogyakarta

In addition to the weather parameters displayed, the 12-volt battery for the AWS instrument power supply is displayed in voltage units, and the temperature in Celsius units is displayed on the data logger component. This helps monitor the performance and condition of AWS instrumentation remotely.

3.2. CONNECTING MYSQL DATABASE

The XAMPP application is utilized to operate the database management system, serving as a data server through its integrated Apache and MySQL modules. These components are

essential for designing and managing the AWS database system, which is structured to store sensor data collected from multiple AWS sites. Within the database, individual tables are created for each site, containing columns for timestamp, site name, and various weather parameters.

Before establishing a connection between Node-RED and MySQL, it is necessary to install the node-red-node-mysql palette. In the Node-RED flow, each AWS site's subscriber is linked to an inject node, which is connected to a JSON node, followed by a function node. Figure 12 provides nodes of MySQL connection from Node-Red.

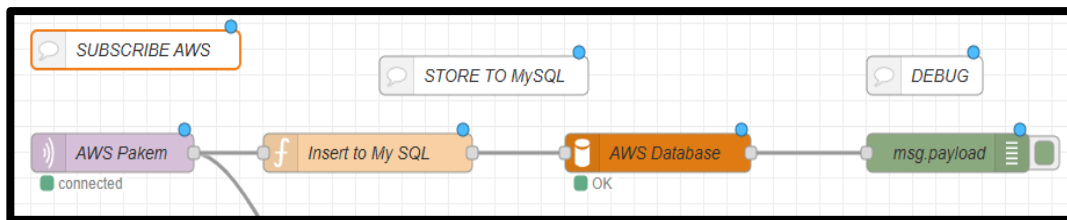


Figure 12. Nodes of MySQL connection from Node-Red

The function node contains a script written in JavaScript, which generates a command to insert data into the corresponding AWS table in the MySQL database. This script ensures that data is mapped accurately to the respective columns based on the weather parameters. Subsequently, the function node is connected to a MySQL node, where the database address, host name, and port number must be properly configured. Once deployed, the connection status can be verified. Figure 13 illustrates MySQL database for each site of AWS connection to Node-RED.

Table	Action	Rows	Type	Collation	Size	Overhead
aws_kulonprogo	Browse Structure Search Insert Empty Drop	584	InnoDB	utf8mb4_general_ci	96.0 Kib	-
aws_mlati	Browse Structure Search Insert Empty Drop	586	InnoDB	utf8mb4_general_ci	96.0 Kib	-
aws_pakem	Browse Structure Search Insert Empty Drop	573	InnoDB	utf8mb4_general_ci	96.0 Kib	-
aws_prambanan	Browse Structure Search Insert Empty Drop	564	InnoDB	utf8mb4_general_ci	96.0 Kib	-
aws_smpk_kulonprogo	Browse Structure Search Insert Empty Drop	573	InnoDB	utf8mb4_general_ci	96.0 Kib	-
aws_stageof_yogya	Browse Structure Search Insert Empty Drop	572	InnoDB	utf8mb4_general_ci	96.0 Kib	-
aws_tbl	Browse Structure Search Insert Empty Drop	0	InnoDB	utf8mb4_general_ci	16.0 Kib	-
aws_ugm	Browse Structure Search Insert Empty Drop	565	InnoDB	utf8mb4_general_ci	96.0 Kib	-
8 tables	Sum	4,017	InnoDB	utf8mb4_general_ci	688.0 Kib	0 B

Figure 13. Database of AWS network in Yogyakarta

The MySQL database portal can be accessed through phpMyAdmin to view the databases created and the data successfully entered into the MySQL server. In this experiment, an AWS database named aws_db was created, containing seven tables, aws_kulonprogo, aws_mlati, aws_pakem, aws_prambanan, aws_smpk_kulonprogo, aws_stageof_yogya, and aws_ugm. Each site will contain recorded weather parameters and battery conditions. Table 2 displays the sample of structure and contents of the AWS database in MySQL.

Table 2. Sensor Data in MySQL Database of AWS network in Yogyakarta

date	time	site	rain	ws	wd	temp	rh	press	sr	batt
18/02/2025	07:17	AWS Pakem	0	1.018	104	28.29	66.2	960.328	141.6	11.68
18/02/2025	07:18	AWS Pakem	0	3.515	123.5	28.29	66.68	960.259	143.1	11.68
18/02/2025	07:19	AWS Pakem	0	3.891	111.9	28.27	67.65	960.084	152.6	11.68
18/02/2025	07:20	AWS Pakem	0	2.638	190.9	28.19	68.72	960.23	163.6	11.68
18/02/2025	07:21	AWS Pakem	0	1.807	165.5	28.21	68.8	960.222	181.3	11.68
18/02/2025	07:22	AWS Pakem	0	1.448	226.1	28.25	68.57	960.212	221.6	11.69
18/02/2025	07:23	AWS Pakem	0	1.190	169.3	28.37	69.06	960.227	239.2	11.69

From the results of this MQTT-based database delivery, it can answer previous research from Muslim et al. (2021) that the interval can be increased closer from a 5-minute interval to denser data for a 1-minute data interval and for one weather measurement site to more sites to a network of seven weather observation locations.

3.3. CONNECTING GRAFANA

To connect Grafana with AWS data, the process begins by running the Node-RED application and activating the Apache and MySQL services via XAMPP. Grafana is then accessed through a web browser at localhost on port 3000. A new data source is added by selecting MySQL, entering the local host address and the corresponding database name. Once the connection is verified, the configuration is saved and tested. Figure 14 shows the data source setup in Grafana for MySQL-AWS integration.

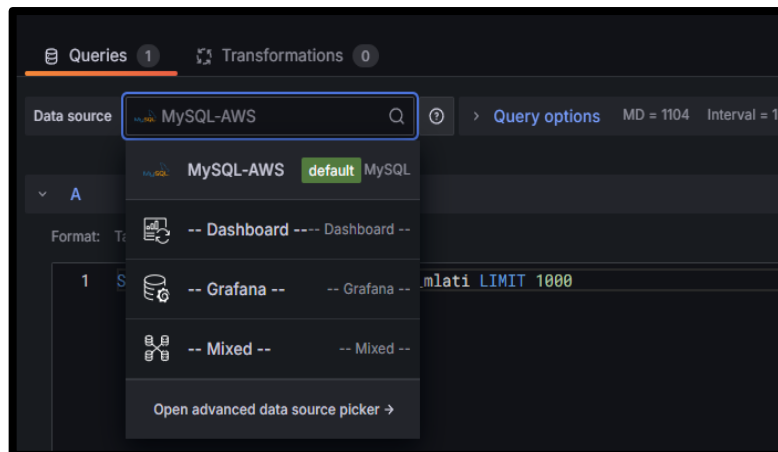


Figure 14. Configuration of Data Source in Grafana to connect MySQL-AWS

In this system design, the visualization is implemented to display AWS weather data originating from the Yogyakarta Climatology Station. A new dashboard interface is developed, with individual panels created for each weather parameter to be presented, ensuring that the data is organized and easily interpretable for monitoring and analysis purposes. Figures 15 illustrate the AWS monitoring dashboard at the Yogyakarta Climatology Office.



Figure 15. Monitoring of Weather Parameters in Yogyakarta Climatology Office using Grafana

On the Grafana dashboard, weather parameters can be displayed using a variety of text panels, numbers, gauges, and graphs. Different panel colours are used for each weather parameter to facilitate easy differentiation. Figure 16 provides dashboard monitoring of instrumentation in Yogyakarta Climatology Office using Grafana.



Figure 16. Monitoring of Instrumentation in Yogyakarta Climatology Office using Grafana

In another panel group, there is an AWS instrument monitoring dashboard that has parameters for the power supply battery, the battery on the data logger component, and the temperature of the data logger component in graphs and numbers. This also represents an improvement over previous research by Adenekan et al. (2021), which created a weather display for temperature, humidity, wind speed, and rainfall parameters. This study also displays more diverse data, including wind direction, air pressure, and solar radiation. Furthermore, the system in this study was designed to monitor equipment performance

through parameters such as the power supply battery, component batteries, and data logger component temperature.

4. CONCLUSION

In the design and implementation of the weather monitoring system on AWS in the Special Region of Yogyakarta using Node-red and Grafana has been successfully used to build a real-time AWS data visualization system. This research also builds on previous studies related to weather monitoring systems from Adenekan et al. (2021) and Muslim et al. (2021) by increasing the number of weather observation points. This research also improves data storage performance for tighter intervals, at 1 minute. Furthermore, the system also improves the diversity of weather parameters and adds monitoring for device performance, including the power supply battery, component batteries, and data logger component temperature.

The advantages of the system using the tools used can present an interactive, flexible display and support fast monitoring. The system has an impact and contribution to the operational activities of technicians, observers, and forecasters in the BMKG work area, especially the Yogyakarta Climatology Station. Suggestions for future research development are integration with an early warning system. Furthermore, the system can be added with automatic notifications for data anomalies, and then this concept can be replicated to other regions in Indonesia.

REFERENCES

- Adenekan, O. A., Sodunke, M. A., Olateju, A. I., & Bello, O. (2021). Weather Variables Monitoring System with A Developed Graphical User Interface using Internet of Thing. *International Journal of Innovations In Engineering Research And Technology*, 8(3), 55–65. <https://repo.ijert.org/index.php/ijert/article/view/2932>
- Aditama, G. K. L., Wijaya, S. K., Siagian, S. B., & Nugroho, W. (2023). Developing an IoT-Based Digital Climate Station at Dramaga Climate Observational Station. *IOP Conference Series: Earth and Environmental Science*, 1167(1). <https://doi.org/10.1088/1755-1315/1167/1/012033>
- Anam, K., Rofi, D. N., & Meiyanti, R. (2023). Monitoring System for Temperature and Humidity Sensors in the Production Room Using Node-Red as the Backend and Grafana as the Frontend. *Journal of Systems Engineering and Information Technology (JOSEIT)*, 2(2), 68–76. <https://doi.org/10.29207/joseit.v2i2.5222>
- Ashfaq, M., & Nur, S. (2024). IoT Sensor Networks- Orchestrating Connectivity, Efficiency, and Intelligence Across Diverse Domains. *International Journal of Innovative Research in Computer Science and Technology*, 12(3), 154–161. <https://doi.org/10.55524/ijirest.2024.12.3.26>
- Fatoni, A. R., Bahtiar, A., Faridatussafura, N., Hermanto, A., & Aulady, M. F. N. (2024). The application of PPE at the Juanda meteorological station (BMKG). *Public Health Risk*

- Assesment Journal*, 1(2), 53–59. <https://doi.org/10.61511/phraj.v1i2.2024.316>
- Mahdi, R. A., Hamed, H. A., & Latif, H. K. (2024). A review of weather conditions monitoring system based on iot. *BIO Web of Conferences*, 97, 1–17. <https://doi.org/10.1051/bioconf/20249700091>
- Marby, I., Meutia, I. F., Hermawan, D., & Yulianto, Y. (2025). Evaluating BMKG Public Information Services in Flood Disaster Areas Using ITO (Input-Throughput-Output) Model Approach. *The Eastasouth Journal of Social Science and Humanities*, 2(02), 250–266. <https://doi.org/10.58812/esssh.v2i02.384>
- Muslim, M. A., Setyawan, R. A., Basuki, A., Razak, A., Hario, F. P., & Fernando, E. (2021). *IOT Based Climate Monitoring System*. <https://doi.org/10.1088/1755-1315/746/1/012044>
- Ogunbunmi, S., Taiwo, A. A., Oladosu, J. B., Hakeem, S., Inaolaji, F. A., Olanokanmi, U. G., Azeez, A. I., Tajudeen, W. A., Christian, C. N., Samuel, A. O., Adeleke, O. J., & Enabulele, E. C. (2024). Internet of things weather monitoring system. *World Journal of Advanced Research and Reviews*, 22(2), 2099–2110. <https://doi.org/10.30574/wjarr.2024.22.2.1647>
- Omidi, S. A., Baig, M. J. A., & Iqbal, M. T. (2023). Design and Implementation of Node-Red Based Open-Source SCADA Architecture for a Hybrid Power System. *Energies*, 16(5). <https://doi.org/10.3390/en16052092>
- Santoso, B., Ryan, M., Wicaksana, H. S., Ananda, N., Budiawan, I., Mukhlis, F., & Kurniadi, D. (2023). Predictive Maintenance Automatic Weather Station Sensor Error Detection using Long Short-Term Memory. *Ultima Computing : Jurnal Sistem Komputer*, 15(2), 41–51. <https://doi.org/10.31937/sk.v15i2.3403>
- Widayanti, & Fajarrohman, A. (2024). Design and Implementation of a Weather Monitoring System Base on IoT Blynk Display and Solar Panel. *Sunan Kalijaga Journal of Physics*, 6(2), 47–52. <https://doi.org/10.14421/physics.v6i2.5209>