Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 8, Nomor 4, October 2024
DOI : https://doi.org/10.33395/sinkron.v8i4.13908
e- ISSN : 2541-2019
p- ISSN : 2541-044X

Rainfall Monitoring Using Aloptama Automatic Rain Gauge and The Network Development Life Cycle Method

Afandi [1], Wakhid Rokhayadi[2], Indri Budiarto[3], Taufan Hermawan[4], Kristiawan Nugroho[5]*
[1,2,3] Department of Information Technology, Fakultas Teknologi Informasi dan Industri Universitas Stikubank Semarang City, Indonesia
[1]afandi0022@mhs.unisbank.ac.id, [2]wakhidrokhayadi0026@mhs.unisbank.ac.id, [3]indribudiarto0025@mhs.unisbank.ac.id, [4]taufanhermawan0028@mhs.unisbank.ac.id, [5]kristiawan@edu.unisbank.ac.id

Submitted :July 19, 2024 | **Accepted** : Sept 8, 2024 | **Published** : Oct 11, 2024

Abstract: This study examines the role of rainfall data management in monitoring and mitigating natural disasters. The Meteorology, Climatology, and Geophysics Agency faces challenges in managing rainfall data between observation posts and the coordinating office. To enhance accuracy and efficiency, BMKG has utilized platforms such as Grafana, Node-RED, Xampp, and MQTT, which have proven effective. This research employs the Network Development Life Cycle (NDLC) model, encompassing stages of analysis, design, prototyping, simulation, implementation, monitoring, and management. By using ARG equipment and NDLC methods, the developed rainfall monitoring system is expected to provide accurate and reliable data. The results of this study aim to improve visual rainfall monitoring, understand rainfall patterns, predict floods, manage water resources, and implement mitigation measures. These findings can serve as a reference for government and institutions to make informed decisions in preventing climate-related disasters.
Keywords: Automatic Rain Gauge, Grafana, Monitoring System, Network Development Life Cycle, Rainfall.

INTRODUCTION

One of the essential components in climate observation is rainfall. To understand rainfall patterns, predict floods, manage water resources, and mitigate the impacts of climate change, accurate and efficient rainfall measurement is crucial. The task of Badan Meteorologi, Klimatologi, dan Geofisika (BMKG) is to observe weather, climate, earthquakes, and tsunami. The challenge in managing rainfall data is the data transmission system from rainfall observation posts spread throughout Central Java to the Central Java BMKG Office, which is not yet fully functional. As an agency responsible for monitoring weather, BMKG has an automated weather observation network using Automatic Rain Gauge (ARG) systems that only use tipping bucket sensors to measure rainfall(Ariffudin & Musa, 2022). The BMKG Coordinator of Central Java collaborates with related agencies and local governments to place ARG equipment throughout the region, totaling 45 observation points. It is expected that with cooperation from various agencies, the placement of these tools can be done effectively and optimally to meet weather monitoring needs in Central Java. However, in practice, these tools have not been able to provide real-time visualization of rainfall data at the ARG equipment locations.

Currently, BMKG uses FTP protocol technology to send data from automatic rainfall observations to the server, but some have switched to the HTTP get protocol. However, there are several issues when using these protocols. One issue is that it takes a long time to transmit weather observation data, and real-time storage on the main server experiences queuing. The FTP protocol also requires large bandwidth, with headers needing 8000 bytes(Ariffudin & Musa, 2022). By implementing IRK server engineering using IoT technology, the MQTT protocol, designed for low bandwidth devices, is implemented on TCP/IP and has a very small data service measurement of less than 2 bytes by applying the "machine-to-machine" (M2M) protocol concept to connect devices(Windryani, et.al, 2019). Additionally, a real-time monitoring data publication system for automatic rainfall observation is designed on the awscenter.bmkg.go.id portal using a web-based platform, requiring data users to log in to access the data. In providing public services to collaborative institutions, especially in terms of rainfall data visualization, the currently available equipment cannot provide an interactive dashboard that displays real-time rainfall data at the equipment locations. This shortcoming can affect the effectiveness of rainfall data monitoring and analysis as well as timely decision-making. Therefore, further development of the visualization and monitoring system is needed to meet the needs of institutions in presenting accurate and up-to-

*name of corresponding author

Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 8, Nomor 4, October 2024
DOI : https://doi.org/10.33395/sinkron.v8i4.13908
e- ISSN : 2541-2019
p- ISSN : 2541-044X

date information and facilitating faster and more precise responses to changing weather conditions. Grafana is used to support real-time data visualization for infrastructure and open-source application analysis.

The use of platforms such as Grafana, Node-RED, Xampp, and MQTT is considered highly reliable for monitoring rainfall. Grafana not only provides a good visualization of the outputs obtained but also displays information from the rain sensors and the condition of devices in the Aloptama ARG system. Additionally, these platforms have proven to be effective in monitoring rainfall, showing real-time data and accurate information. Previous research has demonstrated the reliability of ARG and the NDLC method in producing accurate, up-to-date, and dependable data. A study conducted by Widagdo et.all (2019), revealed that the NDLC method is highly effective in designing monitoring techniques and recording rainfall data using the Grafana platform. Furthermore, Maheswara et.all (2019) investigated the application of the NDLC method in the development and implementation of a GPS-based rain sensor system. Moreover, Wulandari et.all (2020) found through testing that this system has a high success rate in automatically detecting and watering plants, indicating that the NDLC method can be applied not only in computer networks but also in other automation systems. The design process of the NDLC system consists of components that define specific steps, phases, and stages. The implementation of the NDLC model includes steps such as design, analysis, implementation, prototype simulation, monitoring, and management. NDLC elaborates on the processes, stages, and continuous development of the system comprehensively. This is the key to understanding the process of developing a system. This study reinforces previous findings on the advantages of the NDLC method in the development of monitoring system.

LITERATURE REVIEW

According to research conducted by Widagdo et.all (2019) the NDLC method was used comprehensively in designing a rain gauge monitoring and recording system using the Grafana platform. With Grafana as the data visualization platform, the intensity of rainfall is automatically displayed in patterned graphs, making it easier to read and analyze. Every stage of NDLC is applied thoroughly and in detail, making it an effective and efficient choice for developing monitoring systems. In this study, NDLC was used in 100% of the overall monitoring system development process, ensuring all strategic business needs were met with an appropriate top-down approach. This research proves that NDLC is a reliable method for developing modern technology-based monitoring systems. Meanwhile, Maheswara et.all (2019) also affirmed the effectiveness of the NDLC method through evaluating its use in developing GPS systems for rainfall sensors. The evaluation results showed that the NDLC method proved effective and efficient, with an effectiveness rate of around 90%. Each development stage was carried out thoroughly and structured, resulting in an accurate, reliable, and cost-effective system. The resulting system was able to meet the research needs and objectives with periodic monitoring, ensuring the data was always up-to-date and accurate. Additionally, integrating the latest technology made it easier for users to access the monitoring system in real-time. Furthermore, Wulandari et.all (2020) found that this system had a high success rate in automatically detecting and watering plants, demonstrating that the NDLC method could be applied not only in computer networks but also in other automation systems. Based on the testing results, the IoT-based automatic watering system successfully detected and watered the Pothos plant with an 875% success rate. Temperature measurements showed an average error of 207%, while soil moisture measurements successfully identified soil conditions accurately. This system effectively maintained optimal soil moisture conditions, ensuring plants grew well without excessive manual intervention. According to Taufan et.all (2022), the implementation of the NDLC method was very effective in improving computer network accuracy. This process began with analyzing network needs and problems for proper planning. Network design involved creating an optimal topology and determining protocols and security measures. Prototype simulations using tools like GNS3 allowed testing the design before implementation, helping to identify and resolve potential issues. Implementation was carried out under strict supervision to ensure correct installation and configuration. Real-time monitoring ensured traffic and network performance supervision, allowing quick detection and resolution of issues. The management stage involved routine maintenance and policy management to maintain data integrity and network operations. The results showed a 20% increase in connection speed and a 15% reduction in error rates, resulting in a more stable, secure, and efficient network capable of meeting user needs and supporting organizational operations optimally. The latest research by Pratama et.all (2023) showed that implementing the NDLC method successfully enhanced wireless network security at PT Indotruck Utama. NDLC was applied systematically through several stages, including analysis, design, prototype simulation, implementation, monitoring, and management, to design an RADIUS-based authentication system using FreeRADIUS. The CHAP authentication method used proved effective in encrypting login information, thereby better protecting user data. Additionally, NDLC made it easier for administrators to manage users and authorization. The research results showed a 95% increase in network security, making NDLC a highly effective method for designing and implementing reliable network security systems. This shows that NDLC is a highly effective method for designing and implementing reliable network security systems.

Thus, various studies consistently show that the NDLC method is a very effective and efficient approach in various technological applications, whether for automatic monitoring systems, automation, or network security.

*name of corresponding author

Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 8, Nomor 4, October 2024
DOI : https://doi.org/10.33395/sinkron.v8i4.13908
e- ISSN : 2541-2019
p- ISSN : 2541-044X

METHOD

This research method uses NDLC system design, which uses Aloptama ARG (Automatic Rain Gauge). The NDLC method is used to build or design infrastructure networks that have performance statistics and network monitoring(T. Rahman & Nurdin, 2020). NDLC has components to determine phases, stages and steps, or mechanisms of certain processes. Stages: A comprehensive series of processes is established to ensure that the system being developed runs well and efficiently, including design, analysis, prototype simulation, implementation and monitoring and management. As part of the NDLC method for sustainable system development, the following stages are followed(Pratama, et.al, 2023):

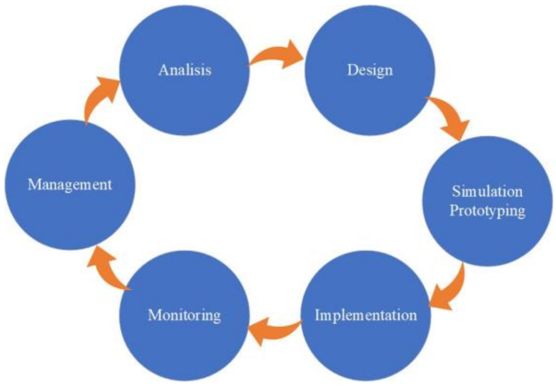


Fig 1. Network Development Life Cycle (Source: Pratama, et.al, 2023)

The flow of explanation of the research methods used so that the research can be carried out correctly and successfully as expected.

1. Analysis Phase: The initial stage of analysis includes identifying needs, current issues, user desires, and hardware requirements to be used. In other words, this step involves data collection to determine the problems and address existing obstacles. After defining the current system, the next step is to analyze the development of the system to be implemented.
2. Design Stage: Based on the previous data, the author creates a prototype design to be built. This can include data access design, topology structure design, layout visualization design, or other elements to provide an overview of the project being created.

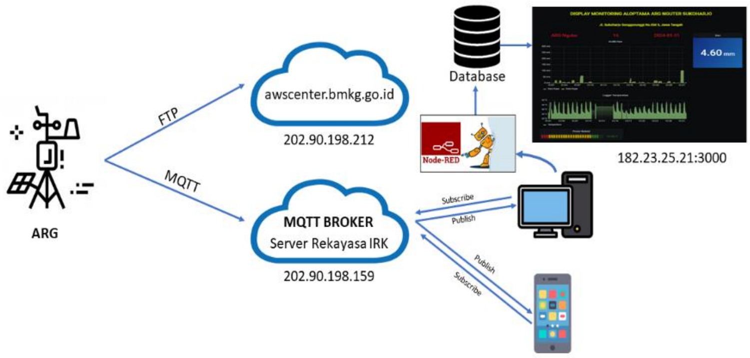


Fig 2. System Modeling Topology Design
(Source: BMKG, Communication Block 2021)

3. Simulation Prototype Stage: which involves developing a system implementation using Grafana and NodeRed, Xampp, MQQT tools intended to test the ability of the display design visualization system to meet current needs.
4. Implementation Stage: This stage takes a considerable amount of time. The implementation is carried out as planned and premeditated. The implementation process consists of several stages, such as installing Grafana tools, NodeRed, Xampp, and MQQT Broker. Next, the NodeRed platform is configured to synchronize with the MySQL database. Instructions are needed to retrieve the data, which is then separated by dashes in the log data. At this point, the impact of the development on the existing system is evaluated.
5. Supervision Stage:This stage is very important after implementation so that the visualization of rainfall monitoring displays using Aloptama ARG and the NDLC method is in according to the author's objectives at the beginning of the analysis. Grafana and NodeRed tools are used in this research to monitor rainfall levels and microcontroller conditions, so that anomalies can be detected and handled immediately.
6. Management Stage: This stage is very important, and special attention should be paid to policy issues because

*name of corresponding author

DOI : https://doi.org/10.33395/sinkron.v8i4.13908
e- ISSN : 2541-2019
p- ISSN : 2541-044X

maintenance and management activities are included in this stage. To ensure that the NDLC method and creating a Rainfall Monitoring visualization design using Aloptama ARG (Automatic Rain Meter) runs effectively and efficiently, this management stage is integrated with other stages in the NDLC. To ensure that the system that t has been built and functions well, and is durable and renewabl, policies must be created.

RESULT

The Automatic Rain Gauge (ARG) system, which utilizes a tipping bucket sensor with a measurement accuracy of 0.2 mm, demonstrates significant efficiency in monitoring rainfall intensity. The ARG-BMKG system comprises a dry battery, GPRS modem, solar panel, tipping bucket sensor, and datalogger, with data transmission to the central BMKG server occurring every 10 minutes. Initial data storage is conducted on a Campbell CR310 logger, followed by internet-based data transmission using a 4G GPRS modem to awscenter.bmkg.go.id and the IRK engineering server at BMKG Central Jakarta. Users must log in via the web to access weather observation data.

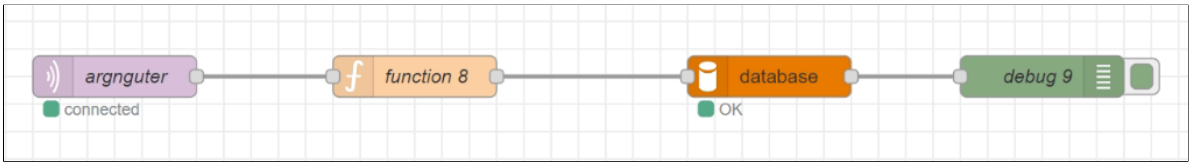


Fig 3. Workflow Diagram in Node-RED ARG

Figure 3 illustrates the workflow diagram implemented in Node-RED for research on monitoring rainfall intensity using an Automatic Rain Gauge (ARG) equipped with a tipping bucket sensor. In this system, rainfall data is transmitted via the MQTT protocol to a broker and received by the "argnguter" node in Node-RED. This node is responsible for subscribing to a specific topic from the MQTT broker and receiving messages containing rainfall data.

Once the data is received by the "argnguter" node, it is processed by the "function 20" node, which is a function node in Node-RED. This node allows for the writing of JavaScript scripts to process the data as required for the research, such as transforming the data format or performing specific calculations for further analysis. The processed data is then forwarded to the "database" node, which functions to store the data in the database system used in the research. This node ensures that the processed rainfall data can be accessed and analyzed further by researchers through structured and centralized storage. For monitoring and debugging purposes, the "debug 40" node is used to display the data output in the debug tab of the Node-RED interface. This node allows researchers to view and verify the data received and processed by the system, ensuring that each stage of the workflow is functioning correctly and that the generated data is accurate. The implementation of this workflow in Node-RED demonstrates how an MQTT-based rainfall monitoring system can be integrated with intuitive data processing software, supporting efficiency and accuracy in climate and rainfall monitoring research.

MQTT is a messaging protocol that features a simple publish/subscribe model, making it suitable for instruments with minimal bandwidth. With an MQTT header size of 2 bytes, the protocol can operate with minimal data bandwidth requirements(Ariffudin & Musa, 2022). The implementation of MQTT configuration enables sensor data storage in log files, which can be accessed through Node-RED after the synchronization stage. Node-RED facilitates the creation of applications focused on IoT through an intuitive workflow-based programming environment(Nițulescu & Korodi, 2020), interacting with the MQTT broker to manage message exchanges.

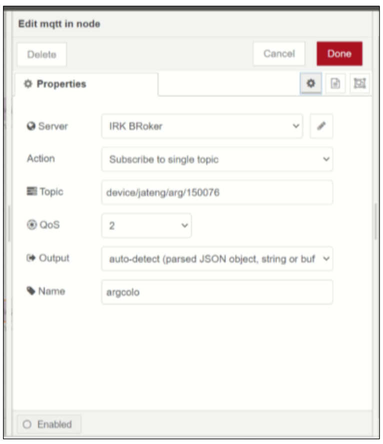


Fig 4. Configuration, Implementation of MQTT and Node-RED ARG

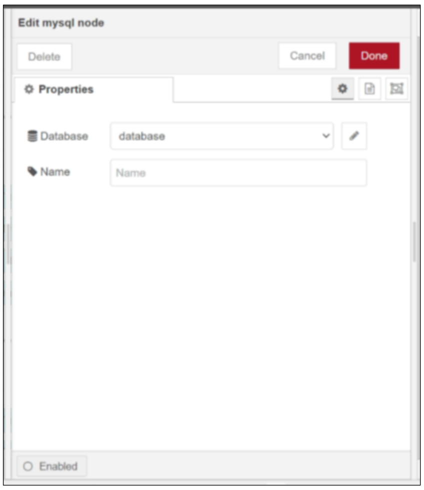
*name of corresponding author

Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 8, Nomor 4, October 2024
DOI : https://doi.org/10.33395/sinkron.v8i4.13908
e- ISSN : 2541-2019
p- ISSN : 2541-044X

Image 4 shows the user interface for editing an MQTT node in Node-RED.This node is configured to subscribe to an MQTT topic from a broker named "IRK Broker." Here is a detailed explanation of the configuration visible in the image:

- Server: The selected server for the subscription is "IRK Broker." This indicates that the node connects to the previously configured MQTT broker with that name. • Action: The chosen action is "Subscribe to single topic," meaning the node subscribes to a specific topic. • Topic: The topic to which this node subscribes is "device/jateng/arg/150076." In the context of MQTT, a topic is a path used to send and receive messages. This topic indicates that the node receives all messages sent to this path.
- QoS (Quality of Service): The chosen QoS value is 2. QoS determines the reliability of message delivery: QoS 2: Messages are delivered exactly once, with a stricter acknowledgment mechanism to prevent duplication.
- Output: The output is set to "auto-detect (parsed JSON object, string, or buffer)." This setting ensures that Node-RED automatically detects the type of data received (whether a JSON object, string, or buffer) and parses it as needed.
- Name: This node is named "argnguter," which helps identify the node in the Node-RED flow. Additionally, the node is enabled.

This configuration demonstrates how an MQTT node can be set up in Node-RED to listen for messages from a specific topic with a defined reliability level and automatically process the received messages based on their data type. Research related to this can explore the use of MQTT in IoT (Internet of Things) applications and how Node-RED can facilitate the integration and processing of data from various connected sensors and devices(Shahri, et al, 2022). Data insertion into the MySQL database is efficiently managed using JavaScript within Node-RED, with SQL commands ensuring accurate data storage and retrieval. The debugging tools in Node-RED allow for real-time message inspection, which is crucial for validating system performance.



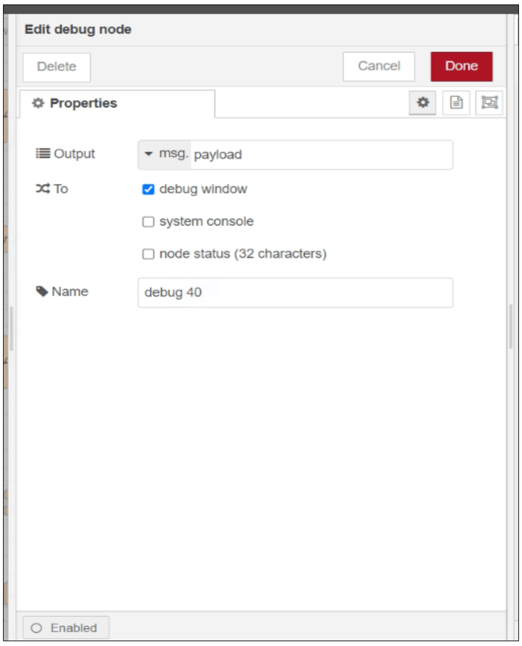

```
msg.topic = "INSERT INTO
tabel_argnguter(id,time_arg,rr_arg
,batt_arg,log _temp,site,date)" +
    "VALUES ('" + msg.payload.id +
"','"
    + msg.payload.time + "','"
    + msg.payload.rr + "','"
    + msg.payload.batt + "','"
    + msg.payload.log_temp + "','"
    + msg.payload.site + "','"    +
msg.payload.date + "');";
return msg;
```

Fig 5. Configuration of Functions and Database for ARG

Figure 5 illustrates the MySQL node property configuration for storing data into a database named 'database'. The right side of the figure displays the function code used to format the 'INSERT' SQL command to be inserted into the 'tabel_argnguter' table. This table includes columns 'id', 'time_arg', 'rr_arg', 'batt_arg', 'log_temp', 'site', and 'date'. The code within the function constructs the SQL command by extracting values from the message payload, which are 'id', 'time', 'rr', 'batt', 'log_temp', 'site', and 'date'. The function concatenates the SQL string with the payload values to form the complete 'INSERT INTO tabel_argnguter' command and then returns the updated message object. This configuration is crucial for automating the process of storing data received from sensors or other data sources into the MySQL database(Stojkoska, et.al, 2021), enabling efficient data storage and further analysis. This research demonstrates the application of IoT (Internet of Things) technology for automatic data collection and storage, which is highly relevant for ongoing studies.

*name of corresponding author

Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 8, Nomor 4, October 2024
DOI : https://doi.org/10.33395/sinkron.v8i4.13908
e- ISSN : 2541-2019
p- ISSN : 2541-044X




```
INSERT INTO
tabel_argnguter(id,time_arg,rr_arg
,batt_arg,log_temp,site,date)VALUE
S
('150076','12:23:00','0','12.75','
29.44','ARG Nguter','2024-08-04');
: msg.payload : ResultSetHeader
```

Figure 6. Debug Configuration 40 and ResultSetHeader Results

This node is configured to output the payload of messages to the debug window, providing instant visibility into the data being processed. Research indicates that the use of debug nodes enhances system development efficiency by expediting the process of identifying and resolving issues, ensuring that any changes in data flow can be promptly analyzed and addressed.

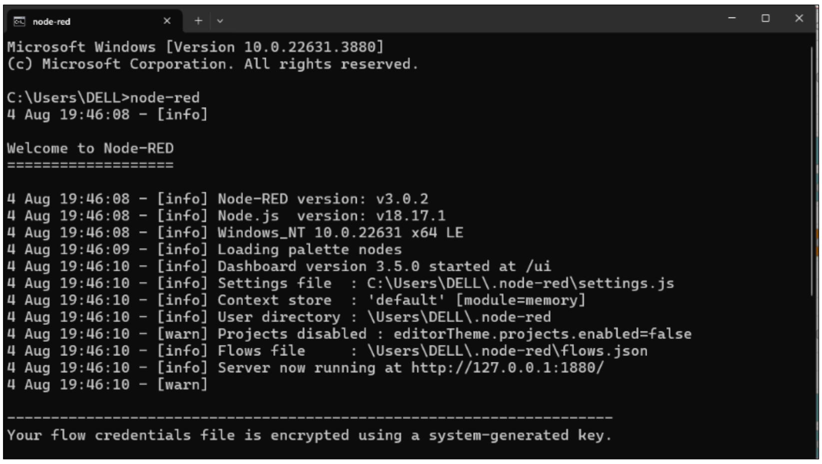


Fig 7. Running the Node-RED Web Server

After the installation and configuration of Node-RED are complete, the next step is to run the Node-RED web server to begin the rainfall monitoring process. To do this, users need to open the Command Prompt and type the command node-red. Once this command is executed, the Command Prompt window displays information about the Node-RED and Node.js versions being used, the operating system, and ongoing processes such as loading node palettes, dashboard versions, and the configuration files being used. In the example provided, the installed version of Node-RED is v3.0.2 and the Node.js version is v18.17.1, running on Windows 10. After all the necessary components are loaded, the Node-RED web server runs at the address http://127.0.0.1:1880/. Users can access the Node-RED dashboard through this web address to start monitoring the collected rainfall data. Additional information displayed in the Command Prompt window includes warnings about the flow credentials file being encrypted using a system-generated key. This step is crucial to ensure that all Node-RED components function correctly and are ready to be used for rainfall monitoring using the Automatic Rain Gauge (ARG) from Aloptama. With the web server running, users can configure and monitor rainfall data in real-time through the Node-RED dashboard interface.

*name of corresponding author

Monitoring Visualization Dashboard using Grafana. All sensor data and devices included in the database system, updated every minute, are used as data sources to be displayed on the Rainfall Monitoring screen using Loptama ARG through Grafana. Grafana is an open-source visual and analytical software (Rahman,et.al, 2020). Grafana automatically implements an Application Programming Interface (API) to connect to data sources. Once connected, the data source displays rainfall data, and queries are required to present the data on the desktop available in Grafana.

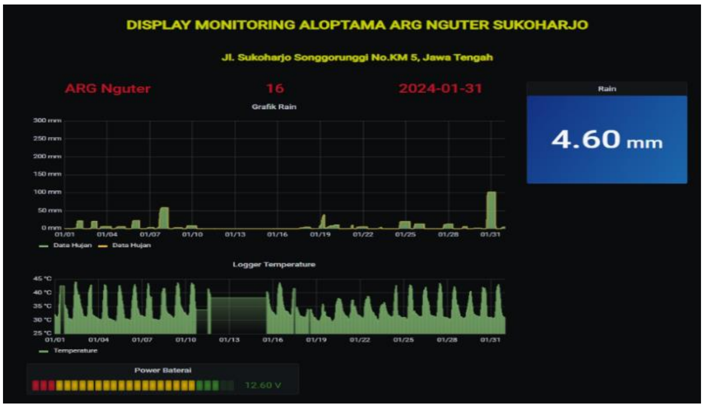
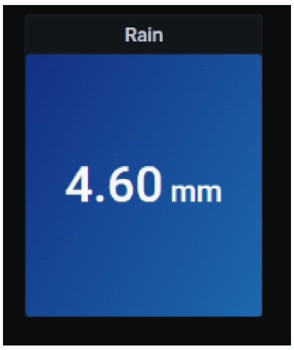


Fig 8. ARG Rainfall Monitoring Grafana
Display Visualization Dashboard

From the dashboard display above you can access using the IP address ttp://182.23.25.21:3000/. The dashboard above has an information panel that shows rainfall intensity, rainfall graph, battery power status, and engine temperature status recorded by the engine logger. To display the Rainfall Monitoring display via Grafana, data is input into the database system, and updated every minute, then used as a data source. Apart from data that is updated every minute from the rain sensor, which comes from the telegraph that has been embedded in the Raspberry Pi, the dashboard display panel takes data from the rain sensor and then displays a graph of rainfall intensity. Apart from that, devices such as batteries are also monitored via the battery power panel and a graphic display of the logger's temperature conditions to determine battery use and use. According to the Query dashboard display above, data from sensors is stored in a database, and grafana is used to process and access the data. This monitoring and control system can facilitate supervision in monitoring rainfall and the condition of ARG equipment(Fenny Vinola, 2020). When the pointer is not shown on the data display, the data is directly displayed as a graphic image with sequential time. However, when the pointer is directed at the Rain inspect_data panel and selected, the date and time columns appear, so the reader does not need to worry about being unable to read the data displayed on the Grafana screen. The grafana display is very reliable to use. Dashboards are great for displaying important information needed to achieve goals on one screen.

DISCUSSIONS

The panel was chosen because it can show real-time rain times and the form of movement of rain intensity, along with the time and date of data obtained in real-time. The panel was chosen because it meets the needs and makes it easier to read rainfall sensor data easily and precisely. Code scripts 13 describe data requests for the rainfall panel and rainfall graph.




```
SELECT
  waktu AS "time",
  rr_arg
FROM tabel_argnguter
WHERE
  waktu BETWEEN
FROM_UNIXTIME(1704042000) AND
FROM_UNIXTIME(1706720399)
ORDER BY waktu
```

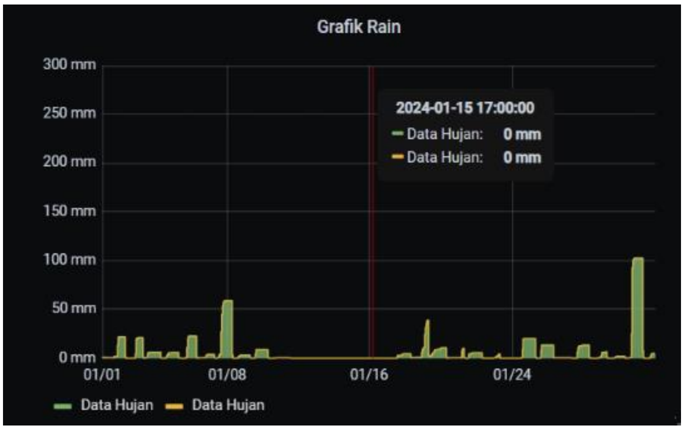
Fig 9. Command and display rainfall data

*name of corresponding author

Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 8, Nomor 4, October 2024
DOI : https://doi.org/10.33395/sinkron.v8i4.13908
e- ISSN : 2541-2019
p- ISSN : 2541-044X

Commands to access data in a database are called queries(Rachmawati, 2019). The query provides instructions for using the rain panel to display rainfall data. The latest rainfall amounts, whether increasing or decreasing, can be seen using the timeseries graph(Mulyani & Oktiawati, 2022).

. In the graphic image, the x-axis is the output of grouping the date and time columns, and the y-axis is the rainfall data collected from the rainfall field, which displays the level of rainfall intensity in data form. The current table comes from a MySQL database.




```
SELECT
  UNIX_TIMESTAMP(waktu) DIV
3600 * 3600 AS "time",
  site AS metric,
  avg(rr_arg) AS "rr_arg"
FROM tabel_argnguter
WHERE
  waktu BETWEEN
FROM_UNIXTIME(1704042000) AND
FROM_UNIXTIME(1706720399)
GROUP BY 1,2
ORDER BY UNIX_TIMESTAMP(waktu)
DIV 3600 * 3600
```

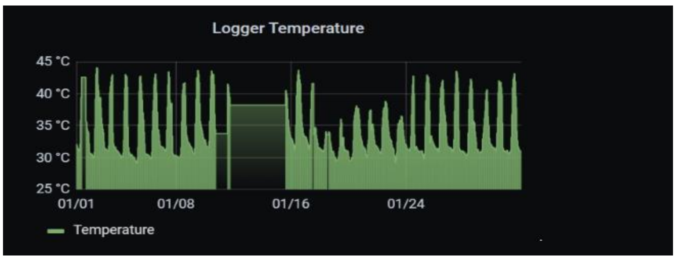
Fig 10. Command and displays rainfall graph

The query above displays the microcontroller battery usage data in the panel, making it easier to read. battery power panel to monitor and read battery usage on equipment. Raspberry and microcontroller observations are critical when developing various sensor mounting locations. This is especially important in vulnerable and hard-to-reach geographic environments. This can be used to solve problems if anomalies occur in writing rainfall data and current equipment conditions.


```
SELECT
  UNIX_TIMESTAMP(waktu) DIV 7200 *
7200 AS "time",
  avg(batt_arg) AS "batt_arg"
FROM tabel_argnguter
WHERE
  waktu BETWEEN
FROM_UNIXTIME(1704042000) AND
FROM_UNIXTIME(1706720399)
GROUP BY 1
ORDER BY UNIX_TIMESTAMP(waktu) DIV
7200 * 7200
```

Fig 11. Command and display battery level

For the query above to display temperature usage data from the logger microcontroller, the temperature logger panel is used to make it easier to read the logger temperature data. In developing a good and high quality ARG microcontroller installation location, it is very necessary considering the very diverse geographical locations. To record rainfall data and the condition of existing equipment.




```
SELECT
  UNIX_TIMESTAMP(waktu) DIV
3600 * 3600 AS "time",
  avg(log_temp) AS "batt_arg"
FROM tabel_argnguter
WHERE
  waktu BETWEEN
FROM_UNIXTIME(1704042000) AND
FROM_UNIXTIME(1706720399)
GROUP BY 1
ORDER BY UNIX_TIMESTAMP(waktu)
DIV 3600 * 3600
```

Fig 12. Command and displays graphics temperature

*name of corresponding author

Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 8, Nomor 4, October 2024
DOI : https://doi.org/10.33395/sinkron.v8i4.13908
e- ISSN : 2541-2019
p- ISSN : 2541-044X

This display shows rainfall and temperature data throughout the month of January. The rainfall chart, titled "Grafik Rain," illustrates the daily rainfall amounts in millimeters, with several peaks occurring around January 7th, 14th, 18th, and 31st. On the right side of the display, there is a box highlighting the current rainfall amount, which is 4.60 mm. The "Logger Temperature" section shows daily temperature fluctuations in degrees Celsius, with temperatures generally ranging from 30°C to 40°C and some peaks approaching 45°C. At the bottom of the display, there is a battery status indicator showing the current battery voltage at 12.60 V. This display provides comprehensive information about weather conditions and the status of the monitoring equipment throughout January. Overall, this monitoring display offers valuable data on rainfall, temperature, and the battery status of the monitoring equipment throughout January, which is crucial for environmental monitoring and ensuring the proper functioning of the monitoring equipment.

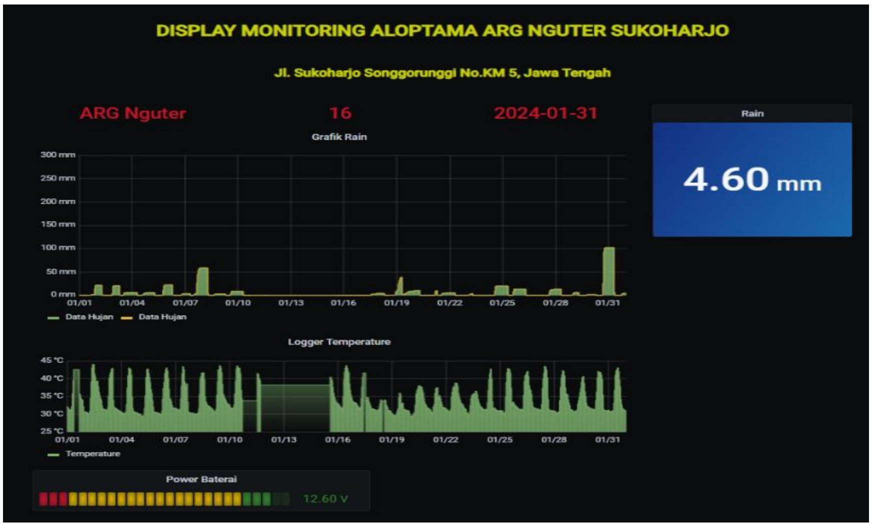


Fig 13. ARG Rainfall Monitoring Grafana
Display Visualization Dashboard

Table 1. Data monitoring test results

No	site	time	id	rr_arg	batt_arg	log_temp
1	ARG Nguter	01/31/2024 07:00:04	150079	4.6	12.8	31.75
2	ARG Nguter	01/31/2024 07:01:04	150079	4.6	12.82	31.79
3	ARG Nguter	01/31/2024 07:02:04	150079	4.6	12.75	31.74
4	ARG Nguter	01/31/2024 07:03:05	150079	4.6	12.73	31.92

Table 1. The test results show that the monitoring system successfully collected and displayed metric data in real-time within one minute.

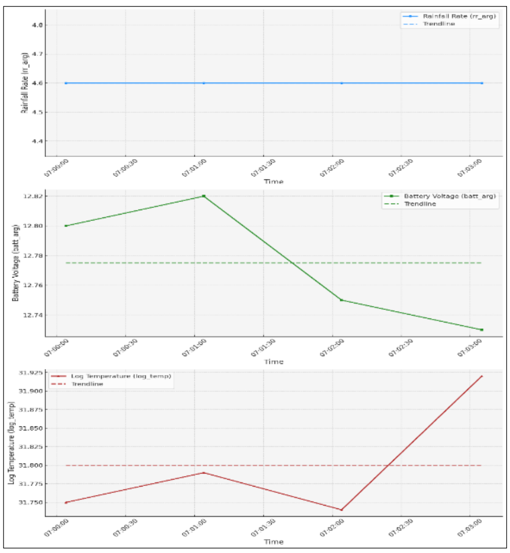


Fig 14. Test monitoring results

*name of corresponding author

e- ISSN : 2541-2019
p- ISSN : 2541-044X

CONCLUSION

The results of research regarding rainfall monitoring using Aloptama ARG and the Life Cycle Network Creation Method show that establishing a rainfall system with Grafana display visualization can make it easier to manage data, including monitoring and downloading data originating from sensors. Grafana can also provide microcontroller information with equipment that supports rainfall sensors. This is very helpful and is hoped to help improve visual monitoring of rainfall in the local area. It can also help people better understand rainfall patterns, flood predictions, water resource management, and mitigation measures as a basis for governments and organizations working together to avoid disasters related to climate change. Rain intensity patterns become easier to read with this modeling system because the data obtained is directly forwarded to a database, then formed into rain panels and patterned graphs.

REFERENCES

Ariffudin, A., & Musa, P. (2022). Analisa sistem komunikasi data berbasis Internet of Things (IoT) menggunakan metode PIECES pada Sistem Pengamatan Cuaca Otomatis di Badan Meteorologi Klimatologi dan Geofisika (BMKG). *Jurnal Meteorologi Dan Geofisika*, *23*(2), 81. https://doi.org/10.31172/jmg.v23i2.831

Fenny Vinola, A. R. S. (2020). 1-Sistem Monitoring dan Controlling Suhu. *Jurnal Teknik Elektro Dan Komputer*, *9*(2), 117–126.

Maheswara, P. L., Bayu, T. I., & Susetyo, Y. A. (2019). Perancangan Global Positioning System (GPS) Pada Sensor Curah Hujan Menggunakan Raspberry Pi. *Indonesian Journal of Modeling and Computing*, *2*(2), 9–16.

Mulyani, A., & Oktiawati, U. Y. (2022). Implementasi Arsitektur Serverless Internet of Things pada Monitoring Cold Chain. *Journal of Internet and Software Engineering*, *3*(1), 36–41.

Nițulescu, I. V., & Korodi, A. (2020). Supervisory Control and Data Acquisition Approach in Node-RED: Application and Discussions. *Internet of Things*, *1*(1), 76–91. https://doi.org/10.3390/iot1010005

Pratama, A. P., Sugiyanta, L., & Idrus, A. (2023). Design And Implementation of Freeradius as A User Manager on The Mikrotik Hotspot Network at PT Indotruck Utama using The NDLC (Network Development Life Cycle) Method. *International Journal of Information System & Technology Akreditasi*, *7*(2), 136–143.

Rachmawati, R. (2019). Analisis Kesalahan Menerapkan Bahasa Sql (Structure Query Language) Mata Kuliah Basis Data. *PRISMATIKA: Jurnal Pendidikan Dan Riset Matematika*, *1*(2), 27–34. https://doi.org/10.33503/prismatika.v1i2.431

Rahman, D., Amnur, H., & Rahmayuni, I. (2020). Monitoring Server dengan Prometheus dan Grafana serta Notifikasi Telegram. *JITSI : Jurnal Ilmiah Teknologi Sistem Informasi*, *1*(4), 133–138. https://doi.org/10.30630/jitsi.1.4.19

Rahman, T., & Nurdin, H. (2020). Abdul Hamid No.77, RT.8/RW.4, Cawang, Kramat Jati, Jakarta Timur 13630 1 Sekolah Tinggi Manajemen Informatika dan Komputer Nusa Mandiri. *Jl. Raya Jatiwaringin*, *5*(1), 23.

Shahri, E., Pedreiras, P., & Almeida, L. (2022). Extending MQTT with Real-Time Communication Services Based on SDN. *Sensors*, *22*(9), 1–19. https://doi.org/10.3390/s22093162

Stojkoska, B. R., Gjorshevski, H., & Mitreva, E. (2021). Finki Scholar, A Publications Database For Faculty Of Computer Science And Engineering Scholars. *Telfor Journal*, *13*(1), 47–52. https://doi.org/10.5937/telfor2101047r

Taufan, M., Zaen, A., Ramadhan, W., Sistem, F. R., Sumbawa, U. T., Raya, J., Maras, O., Alang, B., Hulu, M., Ntb, K. S., Informasi, T., Teknik, F., Mataram, U. M., Kh, J., Dahlan, A., Ntb, M., Informasi, S., Lombok, S., Lombok, P., … Ntb, M. (2022). *Penerapan Network Development Life Cycle (NDLC) Dalam Pengembangan Jaringan Komputer Pada Badan Pengelolaan Keuangan dan Aset Daerah (BPKAD) Provinsi NTB*. *XIV*(1).

Thias Widagdo, K., Bayu, I., & Susetyo, Y. A. (2019). Pemodelan Sistem Monitoring Sensor Curah Hujan Menggunakan Grafana. *ICM (Indonesian Journal of Computing and Modeling)*, *2*(2), 1–8.

Windryani, N. P., Bogi, N., & Mayasari, R. (2019). Analisa Perbandingan Protokol Mqtt Dengan Http Pada Iot Platform Patriot Comparison Analysis Between Mqtt and Http Protocol in Patriot Iot Platform. *-Proceeding of Engineering*, *6*(2), 3192–3199.

Wulandari, P. A., Rahima, P., & Hadi, S. (2020). Rancang Bangun Sistem Penyiraman Otomatis Berbasis Internet of Things Pada Tanaman Hias Sirih Gading. *Jurnal Bumigora Information Technology (BITe)*, *2*(2), 77–85. https://doi.org/10.30812/bite.v2i2.886

*name of corresponding author