Jurnal Teknik Informatika Vol. 16 No. 1, April 2023 (89-99)
ISSN: p-ISSN 1979-9160 (Print)| e-ISSN 2549-7901 (Online)
DOI: https://doi.org/10.15408/jti.v16i1.31832

JURNAL TEKNIK INFORMATIKA

Homepage : http://journal.uinjkt.ac.id/index.php/ti

Real-Time Monitoring of Gas Fields: Prototype at Pt Gamma Energi Pratama Bogor

Firman Adi Nur Fatin[1], Mukhamad Nurkamid[2], Rizkysari Meimaharani[3], Ahmad Bagus Maskula[4]

[1,2,3] Department of Informatics Engineering, Faculty of Engineering, Muria Kudus University
[4] Department of Engineering, PT Gamma Energi Pratama

[1,2,3] Jl. Lingkar Utara UMK, Gondangmanis, Bae, Kudus, Indonesia
[4] CRC Blok C. 1, Jl. Mayor Oking Jayaatmaja No.63, Ciriung, Cibinong, Bogor, Indonesia

Article:
Accepted: April 16, 2022
Revised: May 13, 2022
Issued: June 15, 2022

***Correspondence Address:**
firmanadinurfatin@gmail.com

ABSTRACT

PT Gamma Energi Pratama is a company engaged in the instrumentation sector. One of the projects undertaken by PT Gamma Energi Pratama is on an oil and natural gas drilling site. The oil drilling process involves a tool called a Gas Jack Compressor originating from the United States. At first, the technician there used the local panel that came with the compressor. But procuring spare parts takes a long time. At the same time, the needs in the field demand to be met immediately. Therefore, the Programmable Logic Controller (PLC) was chosen as a special microcontroller device that can access the compressor via the Modbus Protocol. PLCs can also be connected to Supervisory Control and Data Acquisition (SCADA) applications via Ethernet. This solution monitors data from sensor readings installed on the Gas Jack Compressor. The system is already running with its use only on the local scope. For the development of the system so that monitoring can be carried out in real-time and online, it needs to be linked to flow control devices, database systems, and interfaces for data visualization. Thus, monitoring gas fields can be done in real-time online.

Keywords: *condition monitoring; data visualization; gas jack compressor; IoT; online monitoring; PLC; SCADA*

Jurnal Teknik Informatika Vol. 16 No. 1, April 2023 (89-99)
ISSN: p-ISSN 1979-9160 (Print)| e-ISSN 2549-7901 (Online)
DOI: https://doi.org/10.15408/jti.v16i1.31832

1. INTRODUCTION

The oil and gas industry has experienced significant advancements in the integration of technology in its operations. The use of technology has brought numerous benefits, such as convenience, increased efficiency, and reduced manual labor. Remote-controlled equipment and robots have become crucial components in performing tasks such as drilling and blasting. To further improve the management of operations in the gas field, it is vital to implement a real-time online monitoring system. The Gas Jack Compressor used in pumping crude oil [1] has various sensors that monitor temperature and pressure in the gas field. These sensors are connected to a Programmable Logic Controller (PLC), which manages and receives the sensor data[2]–[6]. The connection between the PLC and the sensors is established via the Modbus Protocol [2], [6], [7]. The received data is then transmitted to a Supervisory Control and Data Acquisition (SCADA) system which is integrated with the local internet network for efficient monitoring and control [2], [7]–[9].

One possible solution to achieve real-time online monitoring is using the MQTT protocol. The MQTT protocol can connect the SCADA system with a flow control program and a MySQL database[10]. These three components can be used on a local device or with the database component on an online server. The SCADA system sends payload data on specific topics to be received by a flow control program which converts the data into SQL queries. These SQL queries are executed to enter the sensor reading data into the database. This data can be stored locally or in the cloud. Cloud database services provide the ability to place the databases that will be monitored online[11], making it possible to monitor the data from anywhere in real-time[12]. The processed data is then visualized using data visualization tools, making it easier to understand and use by interested parties within the company to improve efficiency[5], [13], [14].

This research aims to design and implement a Wireless Sensor Network (WSN) system that will monitor gas fields in real time, simultaneously from any location. The proposed system aims to increase the efficiency of gas field management and provide valuable information to relevant parties within the company.

2. METHODS

This section describes how the proposed system was built. Figure 1 shows the block diagram of the system consisting of various interconnected components to form a cohesive unit. The main element is the Gas Jack Compressor, equipped with pressure, temperature, level, and speed sensors to monitor the various parameters in the gas field. These sensors are connected to the Programmable Logic Controller (PLC) through the Modbus Protocol.

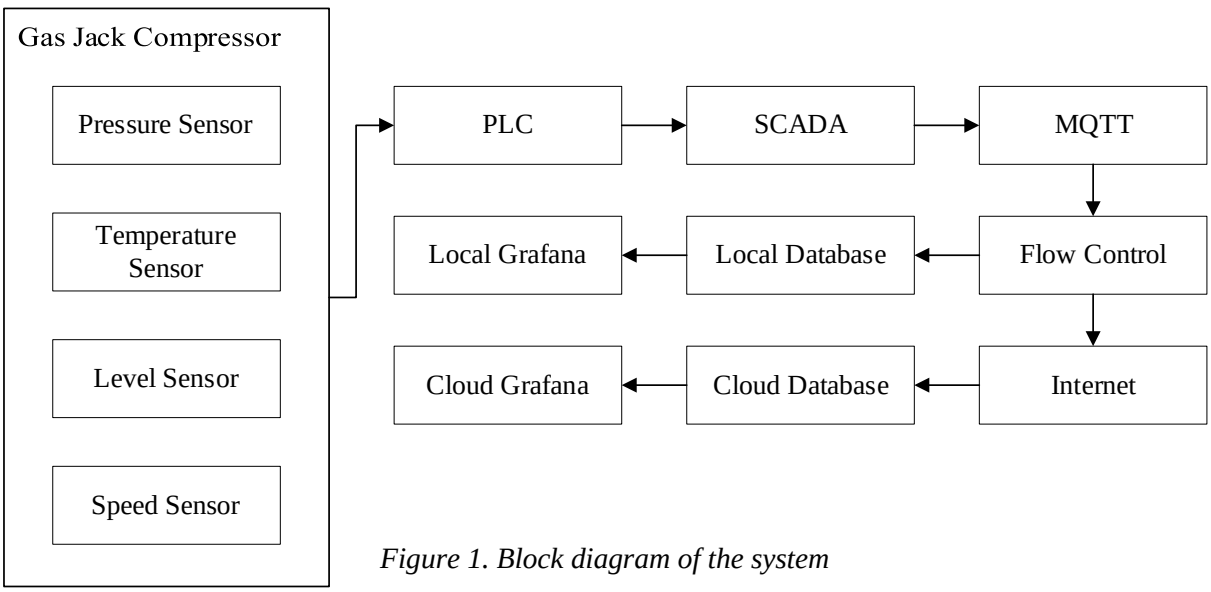

Figure 1. Block diagram of the system

Jurnal Teknik Informatika Vol. 16 No. 1, April 2023 (89-99)
ISSN: p-ISSN 1979-9160 (Print)| e-ISSN 2549-7901 (Online)
DOI: https://doi.org/10.15408/jti.v16i1.31832

The PLC and the SCADA system are connected via Ethernet, allowing the SCADA system to receive data from the PLC and sensors. An MQTT server is installed on the same computer as the SCADA system, along with Node-RED as the flow control, a local MySQL database, and Grafana for data visualization if needed locally. This computer is connected to the internet via a LAN network, enabling it to access the cloud database, which is read by Grafana Cloud for data analysis.

The interconnection of these components enables real-time monitoring of the gas field from anywhere through the integration of SCADA with the cloud databases, which can be accessed using data visualization tools.

Figure 2 shows the system flowchart diagram created. The sensor data from Gas Jack Compressor is recorded, read by PLC, transmitted to SCADA and displayed if required. The MQTT Protocol transfers data with a set topic to Node-RED for processing into SQL queries and stored in a local database. If internet access is available, data is also sent to a cloud database for reading and visualization by Grafana.

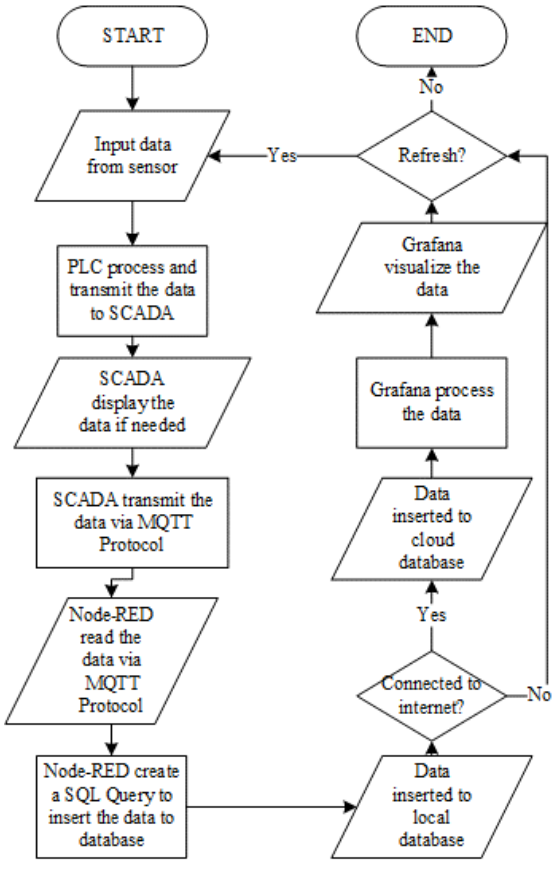

Figure 2. Flowchart diagram of the system

The software development process used the Agile Software Development method. According to Budiman, Sunariyo, and Jupriyadi in their research [15], the agile nature of the method is capable of accommodating rapid and continuous changes in software development in the industry. The following stages were performed:

2.1. Planning

In the planning phase of the software development process using the Agile Software Development Method, the team analyzes the needs and requirements of the system to be developed. The project manager and electrical engineer of PT Gamma Energi Pratama then determine the system concepts, schedule the activities, and decide on the specific parts of the system that will be worked on. The planning phase is a crucial step as it sets the direction for the development process and ensures that the final product meets the needs of the stakeholders. Additionally, the planning phase helps in identifying the resources needed for the project, estimating the cost and time required, and defining the scope of the project [16].

2.2. Implementation

In the implementation phase, the system design and technical plan produced in the planning phase are executed. In the hardware implementation stage, physical components such as Programmable Logic Controllers (PLC), Human-Machine Interfaces (HMI), and sensors are assembled and configured to work together as a system on the Gas Jack Compressor. The software implementation stage involves the development of code for the Supervisory Control and Data Acquisition (SCADA) system, which is a software system used to control and monitor industrial processes. The coding process involves writing the necessary instructions to control and monitor the hardware components in the system, such as the sensors and actuators. The goal of this phase is to create a working system that meets the requirements and specifications outlined in the planning phase [17].

2.3. Testing

During the testing phase, the system is evaluated to ensure that it meets the desired specifications and requirements set out in the planning phase. The black box method refers to

testing the system without any knowledge of the internal workings of the software, focusing solely on the input and output behavior [18]. This method allows the tester to evaluate the system from the user's perspective and to verify that the system functions as intended. If any issues are discovered during testing, they are documented and addressed in the maintenance phase. The successful completion of the testing phase is crucial to ensure that the system performs as expected and meets the needs of the users.

2.4. Documentation

The documentation process in this context includes creating various types of documentation, such as system requirement specifications, user manuals, technical manuals, and maintenance manuals. The system requirement specifications document the system's functional and non-functional requirements, while the user manuals provide guidance for users on how to use the system. Technical manuals provide detailed information about the system's design, architecture, and implementation, while maintenance manuals contain instructions for maintaining the system and troubleshooting common problems. These various types of documentation are essential for ensuring the system's long-term success and usability.

2.5. Deployment

The deployment process involves the installation of the system in the designated location, including the hardware and software components. Once the system is deployed, it is ready for use by end-users who will interact with the system to achieve their objectives. This process is usually carried out by a team of engineers or technicians who have the necessary skills and expertise to install and configure the system. It is important that the deployment process is carried out smoothly to ensure that the system operates effectively and meets the needs of the end-users.

2.6. Maintenance

The maintenance process also includes periodic inspections and troubleshooting to identify and fix any issues that may arise. The maintenance team may also perform preventive maintenance by regularly cleaning and servicing the hardware components to prevent problems from occurring. The software may also require periodic updates and patches to ensure optimal performance and security. The maintenance process is critical to ensure the system continues to function efficiently and effectively over time, avoiding downtime and costly repairs.

3. RESULTS AND DISCUSSION

In this section, we will present the results and discussion of our real-time monitoring prototype for gas fields at PT Gamma Energi Pratama Bogor. The results will cover the design and development of the prototype, including the user interface and monitoring features. We will also discuss the testing phase and evaluate the feasibility of the prototype as a real-time monitoring tool for gas fields. The discussion will focus on the effectiveness of the prototype in improving operational efficiency, identifying potential issues, and providing valuable insights for decision-making.

3.1. Preparation

Before starting the project, the team is carefully selected based on their expertise and understanding of the project's requirements. The project requirements are then gathered and analyzed through meetings with stakeholders. Clear goals and objectives are defined, and a project plan is created, outlining the timeline, milestones, and tasks. The plan also includes resource estimation and identifies potential risks with mitigation strategies.

3.2. Existing System

The existing system consists of a Gas Jack Compressor, Haiwell Card-Type PLC, and Weintek HMI. These three devices are currently undergoing a trial process in a testing workshop for implementation in the field. They will be connected with MQTT to enable online real-time remote monitoring.

The Gas Jack Compressor in the testing workshop features 4 sensors: pressure, temperature, level, and RPM. This device has 14 variables, divided into 5 pressure variables, 4 temperature variables, 3 level variables, and 1 RPM variable. The gas Jack Compressor is connected to both the PLC and the HMI.

Jurnal Teknik Informatika Vol. 16 No. 1, April 2023 (89-99)
ISSN: p-ISSN 1979-9160 (Print)| e-ISSN 2549-7901 (Online)
DOI: https://doi.org/10.15408/jti.v16i1.31832

Figure 3. Gas Jack Compressor

The PLC used in the system is the Haiwell AT16M0R series, a card-type PLC with dimensions of 40mm x 95mm x 65mm. It is designed for industrial use with harsh conditions, offering reliability and adaptability. The modular design allows for customization and easy expansion of I/O modules. Modular PLC is also flexible, maintainable, and easy to install, with a scalable architecture that reduces upgrade costs and meets constantly changing process requirements. Haiwell AT16M0R PLC can also function in harsh environments, including extreme temperatures and high vibration levels.

Figure 4. Haiwell Card Type PLC

The Weintek HMI MT8071iE1 is a 7-inch HMI with a resolution of 800x480 pixels. It features a fan-less cooling system and an LED backlight with a lifespan of over 30,000 hours. It has 128 MB of flash memory and RAM. The GUI on this HMI allows for easy control of a machine or process with buttons, sliders, and indicators for inputting commands and adjusting settings, as well as displaying real-time data and device status information. The HMI measures 200,3 mm x 146,3 mm at the front and 189,6 mm x 135,6 mm at the back. The back also features air ducts and multiple ports.

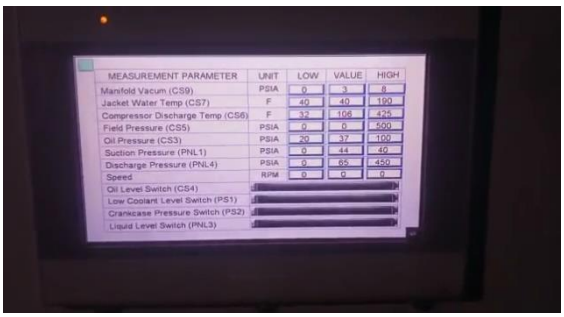

Figure 5. Weintek HMI MT8071iE1

3.3. SCADA Programming

To effectively control the PLC, the development of a SCADA program is necessary. This integration aims to facilitate data transfer from the PLC to the database through the SCADA program. The implementation process involves the following steps: the creation of a new SCADA project utilizing a PC platform runtime, integration of PLC devices, definition of PLC variables, the establishment of a data group incorporating the defined PLC variables, and configuration of the MQTT protocol within the SCADA system. In a new project that is created, it is necessary to add PLC devices that will be integrated. Haiwell PLC was chosen because the device is a device that has been used in the field. The device interface option of Ethernet (TCP/IP) was selected as the means of communication for the PLC device, as it will be connected to a local area network (LAN).

The variables created must be adjusted to the variables read from the sensor device attached to the Gas Jack Compressor. Variable names don't need to be very long. As long as it's recognizable, that's enough. Some variables use register type V (internal data register), and some others use register type X (external input relay) due to differences in data types. Type V register is selected for pressure, temperature, and RPM variables. Meanwhile, for level variables, register X is selected. This difference is due to differences in data types used by the PLC. All of these variables will channel output to devices outside of SCADA.

The MQTT devices used are still installed locally with the aim of cost efficiency. PLC variables are assigned to a data group to be transmitted via the MQTT Protocol. The data group name used is "local_MQTT".

Comprehensive data group naming is needed so that the data groups created are easy

to recognize and differentiate if there are other data groups. A tick is given in the remote report option, and MQTT is selected. In the group identifier, enter "gasjack1". The reporting service quality option is selected "For once Qos2". In the record mode section, Interval Record is chosen with an interval of 5 seconds. Channel Count is filled with the number of transmitted variables, namely 13.

The start index is filled with 1 because the variable index starts from the number 1. Adding variables can be done quickly via the Batch Setting button. All PLC variables can be selected to be included in the data group. If variables are omitted, they can be added at this step. This feature is very helpful, especially when the number of variables used in the project is quite a lot. After adding the variables used, it is highly recommended to recheck them to ensure there are no missing or missing variables and incorrect properties. If all variables are correct, click the OK button at the bottom.

The next step is to configure the data reporting server. The data will be sent to the local MQTT server. In server options, select MQTT. Server desc is filled with "mosquitto local". The project identifier is filled with "gasjack". On the host server, you can fill in the local address, namely 127.0.0.1 or "localhost". Fixed port number 1883. No need to tick Enable escalation verification.

3.4. Configuring MQTT

In this project, the chosen MQTT server is Mosquitto, a widely used communication facilitator in IoT environments. Mosquitto enables devices to send data or messages to a central message broker and enables other subscribed devices to receive these messages, enabling inter-device communication and data exchange without needing a direct connection. The installation process of Mosquitto is followed by executing the file "Mosquitto.exe". The verbose mode is recommended for displaying messages and logs and debugging. Upon activation of Mosquitto, data exchange can be initiated.

3.5. Database Selection

The selection of an appropriate database plays a crucial role in the success of any project. A suitable database ensures the efficient organization and structuring of data, allowing for easy retrieval, storage, and manipulation. This enables the project to handle large volumes of data effectively and ensures data integrity.

The performance of a project heavily relies on the performance of the underlying database. A well-chosen database provides efficient data retrieval and processing, resulting in faster response times. Additionally, a scalable database can accommodate the growth of data and user demands, allowing the project to scale smoothly without compromising performance.

The selection of MySQL as the preferred database for this research project was carefully considered in comparison to other database options such as Oracle, Microsoft Access, and SQL Server. Several reasons justify the choice of MySQL over these alternatives.

MySQL's open-source nature provides cost-effectiveness and accessibility, making it ideal for research projects with limited budgets. Its performance and scalability enable efficient handling of large datasets and high traffic loads, crucial for real-time monitoring systems. MySQL's compatibility with various operating systems, programming languages, and development frameworks allows seamless integration. It prioritizes data security through advanced authentication, access controls, and encryption. The large and active MySQL user community ensures quick issue resolution and access to extensive resources and best practices for successful implementation and maintenance.

Considering these factors, the selection of MySQL as the database for this research project was based on its cost-effectiveness, performance, scalability, compatibility, security features, and the support provided by its vibrant user community. These advantages make MySQL a suitable and advantageous choice for implementing the real-time monitoring prototype of gas fields at PT Gamma Energi Pratama Bogor, distinguishing it from alternatives like Oracle, Microsoft Access, and SQL Server.

Implementing the MySQL database in the cloud rather than locally for this research project has both advantages and disadvantages. One advantage of utilizing MySQL in the cloud is increased scalability and flexibility. Cloud-based databases allow for easy allocation of additional resources as needed, enabling seamless scaling to accommodate growing data volumes or increased user traffic. This

scalability eliminates the need for upfront hardware investments and provides a more dynamic and adaptable infrastructure.

Cloud-based MySQL databases can be accessed from anywhere with an internet connection, making it convenient for remote teams or collaborators to interact with the database. Furthermore, cloud providers offer high availability and reliability through redundant data centers and automated backup and disaster recovery mechanisms, ensuring continuous access to the database.

However, there are also some disadvantages to consider. One such disadvantage is potential concerns regarding data security and privacy. Storing sensitive research data in the cloud may raise apprehensions about data breaches or unauthorized access. It is crucial to select a reputable cloud provider and implement robust security measures, such as encryption and access controls, to mitigate these risks.

Another consideration is the cost implication of cloud-based solutions. While cloud services provide scalability and flexibility, they often involve ongoing subscription or usage-based fees. Depending on the project's budget and resource requirements, the cost of using a cloud-based MySQL database may outweigh the benefits compared to hosting the database locally.

3.6. Database Deployment

The database and Grafana visualization tool will be placed in the cloud for online access, with Amazon Web Services (AWS) cloud service selected. AWS Relational Database Services (RDS) is a managed service that allows easy setup and management of relational databases, with various features, including automatic backups, patching, failover, and scalability.

Steps to start using Amazon RDS on AWS include creating an AWS account, choosing a database engine, creating a new RDS instance, configuring the instance, and connecting to it. The Virtual Private Cloud (VPC) and public access should be considered when configuring RDS databases on AWS, with VPC allowing control of network access and improving security. VPC can also manage subnet groups, security groups, and endpoints to control access and protect sensitive data from the public internet.

The RDS instance deployment process can take several minutes. So once created, the database cannot be used immediately. If the RDS instance deployment process is complete, a notification will appear on the AWS RDS dashboard and in the status section, it says “Available” in green.

The database schema consists of multiple tables, each representing a node in the sensor network. Each table is designed to store data specific to the type of sensor present at the node, ensuring data is properly organized and not mixed. The data type used in each table is carefully selected to optimize storage efficiency. The resulting database structure facilitates easy data visualization.

3.7. Flow Control Coding

Node-RED was selected to set up the flow control software due to its open-source license and user-friendly interface and design flow. To install Node-RED, NodeJS must first be established. The installation process can be initiated through the Command Prompt, and once completed, the Node-RED interface can be accessed via a web browser on port 1880. The interface allows for program flow design and includes a log window for debugging purposes.

To connect Node-RED with MySQL, an extension or palette must be installed. This can be done through the Palette option in the Node-RED Settings menu. The “node-red-MySQL-node” palette is installed by searching for “MySQL” in the Install option.

An MQTT In node must be created in Node-RED to receive data from the MQTT server. This node holds the server connection information, and a set of nodes are required to obtain the data from the MQTT protocol and convert it into SQL queries. These nodes include the MQTT In node, Function node, Debug node, and MySQL node.

The data received from the MQTT server is stored in the “msg” object, and the values of each variable can be retrieved by calling the payload of the “msg” object, which will be converted to a parsed JSON object. The JSON object contains the terminal time, group name, and PLC variable attributes transmitted by SCADA.

The MQTT In node in Node-RED is designed to establish a connection to an MQTT broker and subscribe to messages from a specified topic. The topic can include MQTT

wildcards such as "+" for one level and "#" for multiple levels. To configure the connection to the MQTT broker, the user clicks on the pencil icon and inputs the necessary information, such as the MQTT server address and port number, the MQTT protocol type, and the client ID.

The security panel, which contains options for username and password, is optional. The messages panel allows for configurations for messages sent during different connection states, such as when the connection is established before it is disconnected and when it is disconnected abnormally.

The MQTT In node configuration also includes properties for server options, action options, topic name, QoS type, and output type. Multiple MQTT nodes can share the same broker connection if necessary.

The Node Function in Node-RED allows for the creation of custom JavaScript functions that process incoming messages. The messages are passed to the function as an object, known as "msg", with a property called "payload", which holds the message body. The function can return one or more message objects or choose not to return anything, halting the flow.

The "On Start" tab contains code that runs every time the node is started, and the "On Stop" tab contains code that executes when the node is stopped. If the code in "On Start" returns a Promise object, the node will not handle messages until the Promise has been resolved. The function node converts payloads into SQL queries for insertion into a database.

The code in the On Message panel retrieves the timestamp and payload of the monitored topic and converts it into an SQL query. The resulting query is passed to the MySQL node for execution and the debug node for display. The function node configuration includes the node name and setup panels for On Start, On Message, and On Stop.

The Debug node in Node-RED is utilized to display the properties of a selected message in the Debug sidebar tab and if desired, the runtime log. The default display is the payload attribute of the message object (msg. payload), but other properties, the full message or the resulting JSON expression, can also be configured. The Debug sidebar provides an organized view of the sent messages, making their structure easier to comprehend.

The sidebar also provides information about the receiving time, the source node, and the message type for each message. One can be redirected to the corresponding node in the workspace by clicking on the source node id. The output of nodes can be controlled by enabling or disabling their buttons. It is recommended to deactivate or eliminate unused Debug nodes for optimal performance.

The Debug node can be configured to send all messages to the runtime log or a limited (32 characters) version to the status text under the Debug node. The payload data can be accessed by calling the payload attribute of the message object and viewed in the Debug window. The Debug node screen is connected to the Function node "MySQL Query" so that the output from the Function node can be displayed through the Debug node screen.

The configuration of the Debug node provides control over the selection of the output object and its attributes, including options for routing the output to the debug window, system console, and node status with a 32-character limit. The configuration also includes an input for the node name at the bottom.

The MySQL node provides a means of accessing and querying MySQL databases within a Node-RED workflow. Upon execution, this node typically returns an array of result rows obtained through its query operations against a preconfigured database. The MySQL node in Node-RED has a designated database configuration and node name, which can be altered by clicking the pencil icon button to the right of the database option.

The configuration of the MySQL database node encompasses several parameters, including the server address (which may be specified as an IP address or hostname), port number, username, password, database name, time zone, charset, and node name. The time zone input follows the format of either a plus (+) or minus (-) symbol, followed by two-hour digits, a colon symbol (:), and two-minute digits (e.g., "+07:00" for UTC+7). The time zone input is left blank if the database server is located locally.

To represent both the local and cloud databases in Node-RED, two MySQL nodes need to be created. Representing each database requires configuring each so they can be properly represented in Node-RED. Creating these two nodes allows for efficient communication between the local and cloud databases, ensuring that the data can be

Jurnal Teknik Informatika Vol. 16 No. 1, April 2023 (89-99)
ISSN: p-ISSN 1979-9160 (Print)| e-ISSN 2549-7901 (Online)
DOI: https://doi.org/10.15408/jti.v16i1.31832

accessed appropriately and utilized. To ensure that the two databases are represented correctly, it is essential to carefully configure each database node in Node-RED.

Unlike the local MySQL node, the configuration of the MySQL cloud node must be adjusted to the RDS instance used. The host address can be found on the AWS RDS dashboard under the "Endpoints & ports" section. MySQL nodes that have been created can be connected with node functions. If the connection is successful, an indication will appear under the node. If it is unsuccessful, an error message will appear under the node indicating the type of error that occurred.

3.8. Data Visualization

Grafana Cloud provides online access to dashboards that can be used locally. To use the service, one must register, set up a data source, and create or import a dashboard. Accounts are registered and invited to become members of a stack, where the account information can be viewed and managed by an admin. Teams are configured to divide user types and provide preferences based on role. Supervisors monitor data analysis dashboards. Technicians monitor actual data dashboards.

3.9. Integration to Panel and Testing

This process is carried out with the aim of field trials. The entire program that has been made is installed on the PLC device connected to the HMI. If the program can communicate with the panel, then it is clear that the program can receive data from the PLC. This process can be seen in Figure 6.

Figure 6. Panel integration process

All program components created were tested under the supervision of the project manager from PT Gamma Energi Pratama. The changes requested are only on the Grafana dashboard display. The dashboard is expected to be able to display all variables in one screen display. This change applies to both actual data dashboards and data analysis results.

3.10. Result

After going through the trial process and necessary changes, the Gas Field Monitoring System Prototype at PT Gamma Energi Pratama was able to receive data from the Jack Compressor Gas sensor readings through the SCADA program, transmit it via the MQTT protocol, store the data in a local database and cloud database, and process the data becomes information by visualizing the data with various graphic forms in Grafana.

The data required by the technician is the latest. Therefore, the form of the dashboard must be able to present information in the form of the latest readings and the meaning of the data. The data can mean normal or not normal. In the case of abnormal data, the numbers may fall below the minimum or above the maximum limit. A specialized dashboard for the technician is shown in Figure 7

.

Figure 7. Grafana dashboard for technician

In contrast to technicians, supervisors need information from actual data that has been processed into information in the form of minimum numbers, maximum numbers, averages, and the number of errors in certain variables. This information helps know the device's condition based on the sensor readings. A specialized dashboard for supervisors is shown in Figure 8.

Jurnal Teknik Informatika Vol. 16 No. 1, April 2023 (89-99)
ISSN: p-ISSN 1979-9160 (Print)| e-ISSN 2549-7901 (Online)
DOI: https://doi.org/10.15408/jti.v16i1.31832

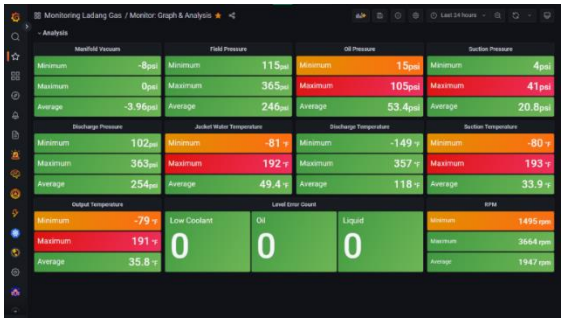

Figure 8. Grafana dashboard for supervisor

The minimum, normal, and maximum limits refer to the configurations that the field technician has determined. If the sensor reads a number outside its normal range, the Gas Jack Compressor will automatically turn off. Such automation is done to prevent damage to the device caused by abnormal conditions.

The implementation of the Kanban Agile methodology, utilizing the Trello platform, had a positive impact on the project from start to finish. The use of this Agile method allowed for adaptability within the project's constraints. Initially, the project planned to use Siemens PLC devices, but due to the lengthy procurement time, they were replaced with Haiwell PLC devices.

The project initially intended to utilize Adisra as the SCADA system. However, since Haiwell also provided SCADA capabilities, it was ultimately chosen as the preferred option. To overcome system limitations, MQTT was employed as a bridge between the SCADA system and the database, enabling seamless communication and data exchange.

The Kanban Agile method, coupled with the Trello platform, provided the project team with a clear visualization of tasks and their progress. This allowed for effective tracking and management of the project's workflow, ensuring timely completion of deliverables. Additionally, the flexibility offered by the Kanban methodology allowed the team to adapt and prioritize tasks based on changing requirements or unforeseen circumstances.

The use of Agile principles, specifically the Kanban approach, facilitated collaboration and transparency among team members. Regular meetings and discussions were held to address any challenges, share progress updates, and make necessary adjustments. This fostered a highly collaborative and responsive work environment, enabling efficient decision-making and problem-solving throughout the project's lifecycle.

CONCLUSION

The integration of Haiwell Cloud SCADA software, Node-RED, MySQL database, Grafana, and Amazon Relational Database Service enabled the reading and storage of data from a sensor on the Gas Jack Compressor into a graph. The data can be stored and visualized through local or cloud databases and Grafana. The company may improve cloud services as needed. To continue adapting to technological advancements, the authors suggest exploring options such as connecting the PLC directly to an MQTT server, using a better automation SCADA, self-managing the server, developing a predictive system, and adding an alerting feature for incidents. Further developments are strongly advised to concede with the affiliated company’s requirements.

REFERENCE

[1] A. N. Indrawan, W. Ardi, Halifah, and S. Mila, “Penggunaan VRU (Vapor Recovery Unit) untuk Mengurangi Emisi Gas Buang (Green House Effect) pada Lapangan ‘S,’” *Ikat. Ahli Tek. Perminyakan Indones.*, vol. 08, no. 035, pp. 1–10, 2008.

[2] S. Bakshi, G. Khairmode, N. Varkhede, and S. Ayane, “Monitoring and control of PLC based automation system parameters using IoT,” *Int. Res. J. Eng. Technol.*, vol. 06, no. 03, pp. 650–652, 2019.

[3] X. Liu, “Design of Remote Monitoring System based on Embedded Web Server,” *8th Int. Conf. Manag. Comput. Sci. (ICMCS 2018)*, vol. 77, pp. 619–623, 2018, doi: 10.4028/www.scientific.net/AMM.556-562.2599.

[4] A. Shabira and W. H. Mulyadi, “Penerapan Scada Pada Pengendali Dan Pemonitor Kecepatan Motor Prosiding Seminar Nasional Teknik Elektro Volume 7 Tahun 2022,” *Pros. Semin. Nas. Tek. Elektro*, vol. 7, pp. 69–72, 2022.

[5] C. Aravind, S. J. Suji Prasad, and M. Ponni Bala, “Remote monitoring and control of automation system with

internet of things," *Int. J. Sci. Technol. Res.*, vol. 9, no. 1, pp. 945–949, 2020.

[6] A. S. Shaikat, R. Tasnim, R. U. Saleheen, M. Rayhan, M. T. Khan, and R. Hasan, "A Real Time Electrical Load Distribution Monitoring and Controlling System based on PLC and Webserver," *Int. Conf. Energy Power Eng. Power Progress, ICEPE 2019*, 2019, doi: 10.1109/CEPE.2019.8726708.

[7] B. Fandidarma, I. Sunaryantiningsih, and A. Pratama, "Pengatur Suhu Ruangan Tertutup menggunakan PLC Schneider Twido Compact berbasis SCADA - Wonderware Intouch," *J. ELECTRA Electr. Eng. Artic.*, vol. 2, no. 2, p. 01, 2022, doi: 10.25273/electra.v2i2.12246.

[8] F. Ozen and M. A. Simsek, "Realization of A Building Automation System Using PLC and SCADA," *Int. J. Eng. Innov. Res.*, vol. 1, no. 1, pp. 28–34, 2019.

[9] E. S. Dewinta, H. Rachmat, and D. S. E. Atmaja, "Perancangan Supervisory Control and Data Acquisition (SCADA) pada Distribution Station Dan Pick & Place Station," *e-Proceeding Eng.*, vol. 8, no. 5, pp. 8435–8440, 2021.

[10] I. Harjanto, "IoT Gateway Menggunakan Protokol MQTT pada Perangkat Kendali Berbasis Modbus-RTU," *J. Ilm. Teknosains*, vol. 6, no. 1, pp. 12–19, 2020.

[11] W. Hofmann, S. Lang, P. Reichardt, and T. Reggelin, "A brief introduction to deploy Amazon Web Services for online discrete-event simulation," *Procedia Comput. Sci.*, vol. 200, no. 2019, pp. 386–393, 2022, doi: 10.1016/j.procs.2022.01.237.

[12] G. Chiesa, S. Cesari, M. Garcia, M. Issa, and S. Li, "Multisensor IoT Platform for Optimising IAQ Levels in Buildings through a Smart Ventilation System," *Sustain.*, vol. 11, no. 20, 2019, doi: 10.3390/su11205777.

[13] L. Feng, Y. You, W. Liao, J. Pang, R. Hu, and L. Feng, "Multi-scale change monitoring of water environment using cloud computing in optimal resolution remote sensing images," *Energy Reports*, vol. 8, pp. 13610–13620, 2022, doi: 10.1016/j.egyr.2022.09.134.

[14] A. S. Ali, C. Coté, M. Heidarinejad, and B. Stephens, "Elemental: An Open-Source Wireless Hardware and Software Platform for Building Energy and Indoor Environmental Monitoring and Control," *Sensors*, vol. 19, no. 4017, pp. 1–20, 2019, doi: doi:10.3390/s19184017.

[15] A. Budiman, S. Sunariyo, and J. Jupriyadi, "Sistem Informasi Monitoring dan Pemeliharaan Penggunaan SCADA (Supervisory Control and Data Acquisition)," *J. Tekno Kompak*, vol. 15, no. 2, p. 168, 2021, doi: 10.33365/jtk.v15i2.1159.

[16] S. Sudaryono, N. P. Lestari, and K. Gunawan, "Perancangan Virtual Assistant Entrepreneurship Menggunakan Metode Scrum," *J. Innov. Futur. Technol.*, vol. 2, no. 2, pp. 66–77, 2020, doi: 10.47080/iftech.v2i2.1021.

[17] M. Nurkamid and A. Widodo, "Penerapan Wireless Sensor Network Untuk Monitoring Lingkungan Menggunakan Modul ESP-WROOM32," *Ikraith-Informatika*, vol. 5, no. 3, pp. 72–78, 2021, [Online]. Available: http://jateng.tribunnews.com

[18] A. Alsaedi, A. Alhuzali, and O. Bamasag, "Effective and scalable black-box fuzzing approach for modern web applications," *J. King Saud Univ. - Comput. Inf. Sci.*, vol. 34, no. 10, pp. 10068–10078, 2022, doi: 10.1016/j.jksuci.2022.10.006.