IMPLEMENTATION AND ANALYSIS OF THE INTERNET OF THINGS SYSTEM FOR ELECTRICAL ENERGY MONITORING AT INSTITUT TEKNOLOGI TELKOM PURWOKERTO

I Ketut Agung Enriko¹, Mas Aly Afandi², Herryawan Pujiharsono³, Fikri Nizar Gustiyana⁴, Hedi Krishna⁵, Filbert H. Juwono⁶

¹,²,³Faculty of Telecommunications Engineering and Electrical Engineering, Institut Teknologi Telkom Purwokerto, Indonesia
⁴Faculty of Electrical Engineering, Telkom University, Indonesia
⁵Indonesia Telecommunication & Digital Research Institute, Indonesia
⁶Connected Intelligence Research Group, University of Southampton, Malaysia

Email: ¹enriko@ittelkom-pwt.ac.id

(Article received: May 03, 2023; Revision: May 22, 2023; published: June 26, 2023)

Abstract

Measurement of electric power usage is carried out using simple measuring instruments and the recording is still manual so that the data obtained is not real-time and accurate. This research aims to implement an electrical energy monitoring system using the Internet of Things (IoT) to obtain real-time information related to electrical energy in the education industry. This research uses an Industrial Grade Power Meter to get a more accurate measurement value. To connect the Power Meter device with the IoT system, this research uses Modbus RS485 communication and a mini PC to process data from the meter, so that the data can be sent to a server using the MQTT communication protocol, and displayed on the Dashboard. The test results of this study indicate that the monitoring system can be implemented and the system runs well with end-to-end measurement results. From the measurement results, the current value (3 phase average) has an average deviation of 0.001 Amperes, Voltage (3 phase average) has an average deviation of 0.519 V, Power factor has an average deviation of 0.012, Active power has a deviation average of 0.000 kW, reactive power with an average deviation of 0.000 kVAR, apparent power with an average deviation of 0.000 kVA and frequency with an average deviation of 0.124 Hz. Then the MQTT protocol has a quality of service with index 4 based on TIPHON standardization on delay, throughput, and packet loss parameters, and index 3 based on TIPHON standardization on jitter parameters.

Keywords: Internet od Things, Monitoring, MQTT, Power, RS485.

1. INTRODUCTION

Currently, technology is developing rapidly in various scientific fields. Humans continue to strive to develop and research the latest technologies to make it easier for humans themselves. One of them is in the field of technology regarding the Internet of Things (IoT).

The Internet of Things has been widely applied in several fields of science and industry, such as in the fields of health, informatics, and geography, and in everyday life, it can also be applied. At the opening ceremony of the 2018 Indonesia Industrial Summit, the Ministry of Industry launched the Industry 4.0 Roadmap [1]. The main technologies supporting Industry 4.0 are artificial intelligence, 3D printing technology, robotic and sensor technology, human-machine interface, and the internet of things. Thus the use of IoT is in line with government policies to build a globally competitive manufacturing industry.

The use of electric power in the industrial sector so far can only be seen using the kWh meter measuring instrument distributed by PLN. One of the industrial sectors that consumes large amounts of electrical energy is the education sector which consumes 72% of electrical energy [2]. This tool does not provide detailed information about how much electricity is connected to the kWh meter. When viewed from historical data, sometimes a change in power usage suddenly drops due to machine failure or unknown maintenance in the education sector. This will result in a waste of electric power. Because of this, monitoring tools are needed for consumers who can show the use of electric power, so they can monitor the use of electric power [3]. To be able to view electricity usage data directly, a system is needed that is designed to display electricity usage data and its characteristics on devices such as computers and Android which can be accessed anywhere, anytime as long as the device is connected to the Internet. In this way, historical data on electricity use can be recorded in an orderly manner, comparing the readings of the PLN meter system with the data obtained from the designed system [4].

In Indonesia itself, the development of smart meter devices has been used in several places, one of
which is in Musi Banyuasin, Palembang. PT Muba Elektrik Power, a Regional Owned Enterprise of the Musi Banyuasin Regency, makes it easy for 48,000 customers to monitor electricity usage through the Smart Electric Muba application [5]. With the application of smart meters, it is hoped that the Education sector can monitor the use of electrical energy through applications, especially in the Educational Area of the Telkom Purwokerto Institute of Technology.

Electric power is defined as the rate of delivery of electrical energy in an electric circuit. In AC power networks with sinusoidal waveforms, several types of power are known, including active power, reactive power and apparent power. [6].

Internet of Things is composed of 2 words, namely, Internet and Things. Internet itself means interconnection and networking [7]. Internet Of Things or commonly referred to as IoT is a concept that aims to be able to expand the benefits of internet connectivity that is connected continuously. [8].

Modbus is an application layer messaging protocol from Modicon, located at level 7 of the OSI model, that provides client/server communication between devices connected on any type of bus or network. [9].

Message Queuing Telemetry Transport is a lightweight publish/subscribe-based protocol specifically designed for communication on low power devices or lightweight messages. [10]. A lightweight publish-subscribe message-based protocol requires a message broker, where the broker is responsible for distributing messages to interested clients based on message topics [11].

Quality of Service (QoS) is the ability of a network to provide a better service for traffic services that pass through it. QoS is an end-to-end architectural system and is not a network feature [12].

2. RESEARCH METHOD
A. Tools and Materials Used
In this Thesis Report, several stages of the workings of the tool that will be used will be explained.

1) Tools
a. Computer/PC In this study, the computer or PC was a PC with the HP 14s-dk0148AU. Which uses an AMD Rezen 5 processor with DDR48GB memory. With SSD storage of 512GB NVme. A PC with this specification is used for report generation, system integration, and QoS measurement.

b. Mikrotik Access point
With it, a router operating system can be created using home devices that can maintain 2GHz and 5GHz dBi. This proxy operates by connecting to the Winbox application speeds. Supported by wireless chip models QCA9880 and QCA9588 with antenna gain specifications of 2-2.5.

The Mikrotik access point used in this study has a display with components that can be used, namely, there is a CPU space of 720 MHz and 128 MB of RAM and a flash size of 16 MB, has five Ethernet ports with SFP space, and an output port and a USB port for a 3G modem. /4G as shown in Figure.2. In this study the authors use a proxy Access Point as an Access Point Wifi IoT Gateway.

c. Node-Red
Node-RED is a browser-based tool for creating Internet of Things (IoT) applications whose visual programming environment makes it easy for users to create applications as “flow”.

Fig. 1. PC HP 14s-dk0148AU

Fig. 2. Mikrotik Access Point
This flow is formed from interconnected nodes where each node performs certain tasks. Although Node-RED was designed for the Internet of Things (IoT), it can also be used for general purposes and various applications. It is named “Node” because it is implemented as a node application but from the customer's point of view, it is only the details of the internal implementation. In this study, the authors use node-red to integrate data from the power meter to the IoT Gateway and create a local dashboard.

Fig. 3. Node-RED

d. Software Wireshark

Wireshark software that functions as a network protocol analyzer. In this measurement methodology, Wireshark, which is useful as a network analyzer, is useful for connecting clients to communicate with each other, and packets are sent from client1 (sender) to client2 (recipient) and vice versa, which is done via a server. In this study, the authors used Wireshark to measure the quality of service on the MQTT protocol. MQTT is easy to deploy and can communicate IoT data efficiently. MQTT supports messaging between devices to the cloud and cloud to devices.

Fig. 4. Software Wireshark

2) Material Used
a. Schneider PM2100

This research used the PM2100 as an implemented power meter in this study. The PM2100 is a digital meter that offers comprehensive 3-phase electrical instrumentation and load management facilities in a compact and rugged package. The PM2100 offers energy monitoring and electricity cost management application features. All meters in the PM2100 series comply with Class 1, or Class 0.5S accuracy standards and deliver high quality, reliability, and affordability in a compact, easy-to-install format. The PM2100 series meters support many features, some of the features are listed below:

a) LED display screen: Intuitive self-navigation using three display LED buttons, with three rows of values together. Two LED columns light up on both sides of the gauge's front panel indicating its parameter name is displayed.

b) Energy calculation and balancing

c) Measurement of True Power Factor and Displacement Power factor

d) Modbus RTU protocol support with RS485 interface

e) And others, which are more complete, can be seen in the PM2100 datasheet in the Appendix [17].

Fig. 5. Power Meter Schneider PM2100

b. Schneider Current Transformer 100/5 METSECT5CC010

To be able to measure the current flowing, the author uses a tool called Current Transform. Schneider Current Transformer METSECT5CC010 is a measuring tool that functions to convert the magnetic field into an electric current that can be processed by PM2100 with specifications reading Current 5A, Maximum voltage rating 720V, Frequency 50/60Hz with IEC Standard 61869-2 and CT 100/100 ratio 5 [18].

Fig. 6. Schneider Current Transformer 100/5 METSECT5CC010
c. Mini Circuit Breaker (MCB) 3 Phase
   MCB has a function as a circuit breaker towards the load direction. Disconnecting the current can be done manually or automatically. To do this manually, that is by changing the toggle switch that is in front of the MCB after that the MCB will cut off the electric current. Meanwhile, it will automatically occur if an overcurrent is detected due to an overload of electricity usage or a short circuit. In this study, the authors used the MCB as a safety device in the circuit.

![MCB 3 Phase](image)

Fig. 7. MCB 3 Phase

**d. Raspberry Pi 3B+**

Raspberry Pi, often abbreviated as Raspi, is a single-board computer (SBC) the size of a credit card that can be used to run office programs, and computer games, and as a media player for high-resolution video. The Raspberry Pi was developed by a non-profit foundation, the Raspberry Pi Foundation, which is fronted by a number of developers and computer experts from the University of Cambridge, England. Raspberry Pi 3 Model B+ is the latest version of the mini-computer released by the Raspberry Pi as a development of the previous Raspberry Pi version and as a successor to the Raspberry Pi 3 Model B which has already been released. Physically, the Raspberry Pi 3 Model B and Raspberry Pi 3 Model B+ are the same so that for the case the Raspberry Pi 3 Model B version can be used. Some of the advantages and improvements to the Raspberry Pi 3 Model B+ compared to the previous ones are as follows:

- Broadcom BCM2837B0, Cortex-A53 (ARMv8) 64-bit SoC @ 1.4GHz.
- 1GB LPDDR2 SDRAM.
- 2.4GHz and 5GHz IEEE 802.11.b/g/n/ac wireless LAN, Bluetooth 4.2, BLE.
- Gigabit Ethernet over USB 2.0 (maximum throughput 300 Mbps)
- Extended 40-pin GPIO headers.
- Full-size HDMI.
- 4 USB 2.0 ports.

Based on the specifications above, the author uses Raspberry Pi as an IoT gateway in this study.

e. RS485 to USB Converter
   This device aims to convert the RS485 communication interface to a USB port that is applied to the Raspberry Pi so that it can be compatible with existing USB 2.0 ports and there is no need to add a RS 485 shield to the Raspberry Pi. The author uses this device to convert RS485 communication to USB on IoT Gateway.

![Converter RS485 to USB](image)

Fig. 9. Converter RS485 to USB

**B. Research Flow**

The research was carried out in several stages, namely the stages starting from the study of the determination of the literature study, system design, system testing, and analysis of the results of the tests that had been carried out.

![Research Flowchart](image)

Fig. 10. Research Flowchart

This research requires a series of stages, from collecting literature studies to the conclusion and recommendation stages. A series of research stages is necessary to achieve the performance of the research process. The first stage is starting with finding sources from research that has been done before. The sources used are sources based on scientific journals, proceedings, and books. Literature study functions as a reference provider so that new concepts can be found that can be developed in research.
Next determine the research location that will be tested, after the test location is determined, the author determines what parameters will be tested related to the system that is made. Then determine the research scheme or scenario, research must determine how the research will be carried out later, so it is necessary to design a research scenario. After the research testing scenario is determined. In this study several parameter tests will be carried out, starting from testing PM2100 data acquisition for each parameter, and testing QoS performance with QoS standards 0, 1, and 2 on MQTT, to find out how good the quality of the protocol used on the system is. get, then do the Hardware and Software design with the tools and materials previously mentioned.

After the hardware and software have been designed, the next stage is the installation of the device at the test.

C. System Planning

![System Block Diagram]

In Figure 11 it can be explained that the design of this tool begins with installing the metering device to the Existing Electrical panel, then the metering device is connected to the IoT Gateway with the RS 485 interface which is converted to USB. The Raspberry Pi uses the Modbus RTU Protocol to acquire data from the PM2100. After the data has been acquired, the Raspberry Pi then forwards the data to the MQTT broker using the publishing method via a WiFi interface that is connected to the Internet using Node-red. After that, PC users connected to the Internet can access the data by subscribing to Node-red and displaying it on a simple Dashboard.

![Block Diagram of System Input and Output]

From Figure 12 above it can be seen that the input from this research is the value of the quantity read by PM2100 from the existing electrical panel. After all, parameters can be read by the PM2100, then the data will be acquired by the IoT Gateway. The IoT Gateway here acts as a bridge that delivers data parameter values and sends them to the internet via a WiFi network. The data will then be displayed in the form of a simple dashboard which is the output of this research.

Here are some explanations of the components in the Block Diagram above:

a) PM2100 functions to acquire electrical parameter data in the form of Current, Voltage, Power (Active, Reactive, Apparent), Power Factor, and Frequency.
c) Raspberry Pi is a device that acquires data from PM2100 and forwards the data to the MQTT broker to be displayed on the Dashboard using Node-Red.

d) Data acquisition from PM2100 to Raspberry Pi uses RS485 interface and Modbus RTU communication protocol.

e) Sending data from the Raspberry Pi to the Broker via the internet using WiFi.

f) The communication protocol used when sending data from the Raspberry Pi to the Dashboard uses the MQTT protocol.

g) The dashboard functions to display real-time data sent by the IoT Gateway.

3) Hardware Design

The design of the entire device is a hardware design and software design on the system created. The hardware design of this system consists of all schematic drawings of the interface design between the PM2100 and the Existing Panel and the PM2100 interface with the Raspberry Pi.

a) The PM2100 interfaces with the Existing Panel

The interface of the PM2100 with the existing panel can be seen in Figure 13.

![PM2100 Interface Wiring Diagram with Existing Panel](image)

![PM2100 Converter Rs485 to USB](image)

![Fig. 14. The PM2100 interfaces with the Raspberry Pi](image)

To connect the interface between PM2100 and Raspberry Pi, a conversion device is needed to convert the RS485 interface to USB as shown in Figure 3.14. For detailed port connections, see Table 2.

<table>
<thead>
<tr>
<th>PM2100 Converter Rs485 to USB</th>
<th>Raspberry Pi</th>
</tr>
</thead>
<tbody>
<tr>
<td>RS485 A</td>
<td>-</td>
</tr>
<tr>
<td>RS485 B</td>
<td>-</td>
</tr>
<tr>
<td>USB Male</td>
<td>USB Female</td>
</tr>
</tbody>
</table>

b) The Interface between PM2100 with Raspberry Pi

The interface between PM2100 and Raspberry Pi. The interface description of the PM2100 with the Raspberry Pi is shown.

c) Software Design

Figure 15 describes the flowchart of the software planning that is made on this system. There are 3 main parts in software planning including IoT gateway, MQTT Broker, and PC user as Dashboard. The flow starts from the IoT Gateway device that requests data transmission to the PM2100 device, if the PM2100 sends a response to a data request, then the next data is coded in the form of String data or sentences. The data sent from PM2100 is data in the form of a buffer. After the data for each parameter is obtained and encoded into the form of the String data type, then the data is combined so that it becomes a message package that will be sent to the MQTT Broker. Simultaneously with data acquisition from PM2100, IoT Gateway also configures MQTT connectivity to connect to the intended Broker. If the IoT Gateway is connected, then the data packets that were previously sent directly to the MQTT Broker. On the Dashboard side, the PC device connects to MQTT Broker. If the device is connected, the device retrieves the data that has been sent by the IoT gateway to the MQTT Broker, then displays it in the form of a Dashboard.
3. RESULT AND DISCUSSION

A. SYSTEM DESIGN RESULT

In this study, there were several designs including the results of hardware design including meter installation to existing panels, IoT Gateway installation, and integration between Meter and IoT Gateway. Then the design of the software includes the display on the Dashboard. The installation results can be seen in Figure 16.

combined so that it becomes a message package that will be sent to the MQTT Broker. Simultaneously with data acquisition from PM2100, IoT Gateway also configures MQTT connectivity to connect to the intended Broker. If the IoT Gateway is connected, then the data packets that were previously sent directly to the MQTT Broker. On the Dashboard side, the PC device connects to MQTT Broker. If the device is connected, the device retrieves the data that has been sent by the IoT gateway to the MQTT Broker, then displays it in the form of a Dashboard.

a) PM2100 Interface Installation Results with Existing Panels

In Figure 17 there are 3 devices for the Pm2100 interface and the existing panel, each of which functions as:

a. The green box is an MCB that functions as a scavenger and breaker of electricity which is measured from the existing panel.
b. The red box is PM2100 which functions as a measuring instrument for electrical energy on the existing panel.
c. The yellow box is a CT device that functions to read the current on each 3-phase cable line (R,S,T)
b) PM2100 Interface Installation Results with Raspberry Pi

In Figure 18 there are 3 devices for the Pm2100 interface and the existing panel, each of which functions as:
1. Raspberry Pi green box that functions as an IoT Gateway.
2. The red box is the RS485 communication interface on the PM2100.
3. The yellow box is the Rs485 to USB Converter device which functions to convert the RS485 interface to a USB port.

c) Flow Node-Red on IoT Gateway

The flow design and configuration of Node-Red on the IoT Gateway can be seen in Figure 19.

Based on Figure 19 there are 7 flow segments which can be explained as follows:
1. Segment 1: using the Modbus Node which functions to Request data parameters that are measured using the Modbus RTU protocol from the RS485 interface connected to PM2100. Modbus RTU communication configuration can be seen in Figure 4.5

The Modbus node configuration consists of parameter names, Unit id (ID number on PM2100), Function Code (Using a Holding Register according to PM2100 Specifications), Address Bus (parameter address code measured according to specifications), Quantity (amount of data buffer taken from PM2100), Poll Rate (request Data on PM2100), and Modbus RS485 server configuration.

Based on Figure 21, the server configuration consists of the server name, communication type, Serial Port (port connected to the USB converter on the Raspberry Pi), Serial Type, Baud rate, Unit Id (Number Id on PM2100), Time Out, and Request Time Out.

2. Segment 2: use the Node function which functions to convert data from the buffer data type to a decimal number data type

3. Segment 3: using the Node template which functions to make the data form for each parameter into a JavaScript Object Notation (JSON) data format. The JSON data format has a "key": "value" structure, where the key value is configured on the template node, and the value is obtained from the value of each parameter requested.

4. Segment 4: uses Node join which functions to combine 7 parameters with the JSON data type.
5. Segment 5: use the Node template which functions to organize JSON data resulting from the merger of segment 4.
6. Segment 6: using the MQTT Publish Node which functions to send data from segment 5 using the MQTT publish protocol with server configuration.
7. Segment 7: uses a debug node that functions to display data connected to this node, namely data from segment 5.

B. SYSTEM TEST RESULT
a) Validation Results of Electrical Parameter Values on the IoT Gateway
Validation is carried out by comparing the value of the reading value on the IoT Gateway with the value on the Display Power Meter for 1 minute with the duration of sending data every second. The value on the Power Meter display can be seen in Figure. 22

![Fig. 22 Server configuration on Modbus Nodes](image)

In Figure 22 it can be seen that the parameter values are based on box number:
1. Voltage (Average 3 phase)
2. Current (Average 3 phase)
3. Power Factor (Average 3 phase)
4. Apparent Power
5. Active Power / Real power
6. Reactive Power / Reactive power
7. Frequency

Based on the measurement results, the researcher obtained 50 data on measurements with a duration of 1 minute as in Appendix 1, and produced an average deviation between the data on the Power Meter and the IoT Gateway as shown in Table 4.5.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
<th>Average deviation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Voltage</td>
<td>0.012</td>
<td></td>
</tr>
<tr>
<td>Current</td>
<td>0.004</td>
<td></td>
</tr>
<tr>
<td>Power Factor</td>
<td>0.000</td>
<td></td>
</tr>
<tr>
<td>Apparent Power</td>
<td>0.000</td>
<td></td>
</tr>
<tr>
<td>Active Power</td>
<td>0.000</td>
<td></td>
</tr>
<tr>
<td>Reactive Power</td>
<td>0.000</td>
<td></td>
</tr>
<tr>
<td>Frequency</td>
<td>0.124</td>
<td></td>
</tr>
</tbody>
</table>

Based on Table 4.5, the current value (3-phase average) has an average deviation of 0.006 Amperes, Voltage (3-phase average) has an average deviation of 0.221 V, Power factor has an average deviation of 0.088, Active power has an average deviation of 0.005 kW, Reactive power has an average deviation of 0.000 kVAR, Apparent power has an average deviation of 0.004 kVA and frequency has an average deviation of 0.07 Hz. The results of these deviations can be caused by several factors such as the difference in time when collecting data between the Power meter and the IoT Gateway on a scale of seconds, fluctuations in electrical parameters that cannot be controlled, and human error when measuring.

b) End to End Electrical Parameter Value Validation Results

![Fig. 23 Electrical value on Display Power Meter (End to End Measurement)](image)

In Figure 23 it can be seen that the parameter values are based on the box number:
1. Voltage (Average 3 phase)
2. Current (Average 3 phase)
3. Power Factor (Average 3 phase)
4. Apparent Power
5. Active Power / Real power
6. Reactive Power / Reactive power
7. Frequency

The data on the dashboard is displayed in the form of a gauge meter for each measured parameter. The dashboard can only be accessed on a computer or PC that has node-red software installed and also has a flow that has been configured for the needs of the User Interface (user interface) so that it makes it easier for users to view data from power meter readings.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
<th>Average deviation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Voltage</td>
<td>0.012</td>
<td></td>
</tr>
<tr>
<td>Current</td>
<td>0.004</td>
<td></td>
</tr>
<tr>
<td>Power Factor</td>
<td>0.000</td>
<td></td>
</tr>
<tr>
<td>Apparent Power</td>
<td>0.000</td>
<td></td>
</tr>
<tr>
<td>Active Power</td>
<td>0.000</td>
<td></td>
</tr>
<tr>
<td>Reactive Power</td>
<td>0.000</td>
<td></td>
</tr>
<tr>
<td>Frequency</td>
<td>0.124</td>
<td></td>
</tr>
</tbody>
</table>

Based on Table 4., the current value (3-phase average) has an average deviation of 0.001 Amperes, Voltage (3-phase average) has an average deviation of 0.519 V, Power factor has an average deviation of 0.012, Active power has an average deviation of 0.000 kW, Reactive power has an average deviation of 0.000 kVAR, Apparent power has an average deviation of 0.000 kVA and frequency has an average deviation of 0.124 Hz. The results of these deviations can be caused by several factors such as the time difference when collecting data between the Power meter and the dashboard on a scale of seconds, uncontrollable fluctuations in electrical parameters, end-to-end packet travel delays, and human error during measurement. Based on the test results, it can be concluded that the electricity monitoring system was successfully implemented because the data from the power meter was successfully sent to the internet with values that almost match the data on the dashboard with deviations as shown in Table 4.
c) Quality of Service (QoS) Test Results

1) Delay

From the data captured on Wireshark that has been exported in CSV form, then the data is processed by performing calculations. Calculations are made using the delay formula (5) which has been explained in the previous chapter, namely:

The following is an example of calculating the delay in the MQTT protocol using formula (5), to calculate the delay:

<table>
<thead>
<tr>
<th>NO</th>
<th>Source</th>
<th>Destination</th>
<th>Protocol</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>100.100.100.253</td>
<td>TCP</td>
</tr>
<tr>
<td>2</td>
<td>34.236.123.240</td>
<td>100.100.100.253</td>
<td>TCP</td>
</tr>
<tr>
<td>3</td>
<td>34.236.123.240</td>
<td>100.100.100.253</td>
<td>TCP</td>
</tr>
<tr>
<td>4</td>
<td>34.236.123.240</td>
<td>100.100.100.253</td>
<td>TCP</td>
</tr>
<tr>
<td>5</td>
<td>34.236.123.240</td>
<td>100.100.100.253</td>
<td>TCP</td>
</tr>
</tbody>
</table>

The first step is to copy the time data into t1 and t2, then do the calculation by reducing the value of t2 with t1. We call the result of the difference delay1. Then calculate the average delay from the existing data, namely using the average function on Ms. Excel. Then the data will automatically appear what is the average value of the delay.

a. Publish Method

<table>
<thead>
<tr>
<th>NO</th>
<th>Time</th>
<th>Source</th>
<th>Destination</th>
<th>Protocol</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>52.202.162.140</td>
<td>100.100.100.253</td>
<td>TCP</td>
</tr>
<tr>
<td>2</td>
<td>34.236.123.240</td>
<td>100.100.100.253</td>
<td>TCP</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>34.236.123.240</td>
<td>100.100.100.253</td>
<td>TCP</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>34.236.123.240</td>
<td>100.100.100.253</td>
<td>TCP</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>34.236.123.240</td>
<td>100.100.100.253</td>
<td>TCP</td>
<td></td>
</tr>
</tbody>
</table>

b. Subscription Method

Based on the results of measuring the average delay on the subscribe method with variations in data size and distance, it can be seen in the table.

When compared with the TIPHON standard, the results of the quality of service parameter jitter produced by the MQTT publish protocol are very good. It can be seen in Table .8 that the average jitter value generated by the MQTT protocol on the subscribe method has a value below 1 ms, which indicates that the jitter is very good.

2) Jitter

From the jitter calculation sample above, it can be seen that the formula for finding the jitter is the value of delay 2 - delay 1. Then after looking for the delay value between packets, then do the calculation of the total delay. Only then can the jitter value be searched by entering formula (6), namely the total delay divided by the packet received minus 1.

a. Publish Method

When compared with the TIPHON standard, the results of the quality parameter jitter produced by the MQTT publish protocol are good. It can be seen in Table .10 that the average jitter value generated by the MQTT protocol in the publishing method has a value below 1 ms, which indicates that the jitter is good.

b. Subscription Method

When compared with the TIPHON standard, the results of the quality parameter jitter produced by the MQTT subscribe protocol are good. It can be seen in Table .11 that the average jitter value generated by the MQTT protocol in the Subscribe method has a value below 1 ms, which indicates that the jitter is good.

3) Throughput

The throughput value can be seen in the average bytes in Wireshark because the value of the average byte is the result of the division between the bytes and the TimeSpan. However, the results obtained from
average bytes are still in bytes, so if you want to get results in bits per second (bps), you can perform calculations using equation (8).

Fig. 22 Throughput Calculations in Wireshark

a. Publish Method

Table. 12. Average jitter of subscribe method

<table>
<thead>
<tr>
<th>Range (m)</th>
<th>44 Byte</th>
<th>56 Byte</th>
<th>80 Byte</th>
<th>128 Byte</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>519,012</td>
<td>568,470</td>
<td>667,262</td>
<td>779,909</td>
</tr>
<tr>
<td>10</td>
<td>518,912</td>
<td>568,426</td>
<td>667,255</td>
<td>782,763</td>
</tr>
<tr>
<td>15</td>
<td>519,045</td>
<td>566,232</td>
<td>667,267</td>
<td>782,719</td>
</tr>
<tr>
<td>20</td>
<td>519,592</td>
<td>568,512</td>
<td>666,268</td>
<td>782,733</td>
</tr>
</tbody>
</table>

It can be seen in Table .12 that at each distance, the larger the data packet sent, the greater the measured throughput. When compared with the TIPHON standard, the results of the quality of service parameter throughput produced by the MQTT publish protocol are very good. It can be seen in Table .12 that the throughput value generated by the MQTT protocol in the publishing method has a value above 100 bps, which indicates that the throughput is very good.

b. Subscription Method

Table. 13. Packet loss publish method

<table>
<thead>
<tr>
<th>Range (m)</th>
<th>44 Byte</th>
<th>56 Byte</th>
<th>80 Byte</th>
<th>128 Byte</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>10</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>15</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>20</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

When compared with the TIPHON standard, the results of the quality of service parameter packet loss produced by the MQTT publish protocol are very good. It can be seen in Table .14 that the packet loss value generated by the MQTT protocol in the publishing method has a value of no more than 0%, which indicates that the packet loss is very good.

b. Subcription Method

Table. 14. Packet loss publish method

<table>
<thead>
<tr>
<th>Range (m)</th>
<th>44 Byte</th>
<th>56 Byte</th>
<th>80 Byte</th>
<th>128 Byte</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>10</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>15</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>20</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

When compared with the TIPHON standard, the results of the quality of service parameter packet loss generated by the MQTT subscribe protocol are very good. It can be seen in Table .15 that the packet loss value generated by the MQTT protocol on the subscribe method has a value of no more than 0%, which indicates that the packet loss is very good.

4). CONCLUSION

Based on the discussion regarding the Implementation and Analysis of the Internet of Things System for Monitoring Electrical Energy at the Telkom Purwokerto Institute of Technology, several conclusions can be obtained as follows:

1. Researchers have succeeded in designing an Internet of Things System for Monitoring Electrical Energy at the Telkom Purwokerto Institute of Technology because electrical
parameter data can be viewed via the Dashboard via the Internet.

2. The results of the design of the Internet of Things System for Monitoring Electrical Energy at the Telkom Institute of Technology Purwokerto can be implemented properly which is marked by the successful transmission of End to End data with a current value (3 phase average) have an average deviation of 0.006 Amperes, Voltage (average 3 phase) has an average deviation of 0.221 V, Power factor has an average deviation of 0.088, Active power has an average deviation of 0.005 kW, Reactive power has an average deviation of 0.000 kVAR, Apparent power has an average deviation of 0.004 kVA and the frequency has an average deviation of 0.072 Hz.

3. Service quality in the delay, throughput, and packet loss categories for the publish and subscribe methods has an index value of 4 according to the TIPHON standard, while the jitter category for the publish and subscribe method has an index value of 3 according to the TIPHON standard.

REFERENCES


