IoT Implementation in Water Level Monitoring System Using ESP32 and Antares Platform

Diono[1] and Hendra Tri Guntoro[2]

∗ Politeknik Negeri Batam
Jurusan Teknik Elektro, Program Studi Teknik Mekatronika
Batam Center, Jl. Ahmad Yani, Kepulauan Riau 29461, Indonesia
E-mail: diono@polibatam.ac.id, hendratriguntoro@gmail.com

Abstract

This research develops an Internet of Things (IoT)-based water level monitoring system that can work in real-time both locally and online. The system is designed using an ultrasonic sensor integrated with an ESP32 microcontroller to process and send data to the Antares platform via the MQTT protocol. The obtained data can be displayed via the Antares dashboard and an I2C LCD as a local display, equipped with LED indicators and a buzzer as a warning when the water level is below the 65 cm threshold. To maintain communication stability, a static IP configuration is used. Test results show that the system has high accuracy with an average error of 0.2 cm, a delay of 129.15 ms, and no packet loss, thus being able to provide reliable performance. Thus, this system is considered effective, efficient, and suitable for application in monitoring water levels for various industrial needs.

Keywords : Internet of Things (IoT), ESP32, Antares, Ultrasonic Sensor, MQTT, Water Level Monitoring, Real-time Monitoring

1. Introduction

Water treatment is an integral part of the production process in the manufacturing industry, including at PT Simatelex. This system plays a crucial role in ensuring the availability of water that meets quality and quantity standards to support various operational activities, such as material washing, product testing, and other production needs. Optimal water treatment system performance significantly impacts the continuity of the production process, requiring continuous monitoring and maintenance. Without proper management, this system can become a critical point that causes production disruptions.

However, an undetected water tank failure resulted in downtime for several production lines. This situation was further exacerbated by the relatively long distance between the maintenance room and the water treatment area, which limited the ability to carry out regular, direct monitoring. To address this issue, an Internet of Things (IoT)-based water level monitoring system was designed and built that provides real-time data and sends automatic notifications to the maintenance room without requiring manual on-site inspection. During its development, sensor testing and calibration were also carried out to ensure the accuracy and consistency of water level readings under various operational conditions, so that the resulting data can represent the actual condition of the storage tank. The implementation of this system is expected to increase the speed of response to potential emergencies, minimize the risk of downtime, and support the stability and sustainability of the water treatment system operations in the PT Simatelex industrial environment more effectively and efficiently.

Several studies have been conducted on Internet of Things (IoT)-based water level monitoring systems to improve monitoring efficiency and provide early warnings of environmental conditions. However, some of these studies still have limitations, particularly the lack of integration of preventive maintenance (PM) schedule reminder features, which are crucial for maintaining the operational sustainability of industrial equipment.

Research conducted by Dicky Resta Kusuma and Munawaroh in a journal entitled "Design of IoT-Based Water Level Monitoring Equipment at PT. Usaha Gedung Mandiri" shows that IoT-based systems are effective for remote monitoring and increasing efficiency in water resource management in industrial environments.

Another related study is "An Internet of Things-Based Flood and Storm Detection System and Weather Monitoring," designed by Ida Bagus Made Lingga

Pradirta, I Nyoman Piarsa, and I Putu Arya Dharmaadi. This study uses ultrasonic sensors to measure water levels, making it an important reference in the development of an IoT-based water level monitoring system.

2. Metode

3.1 Design

The design of a water level monitoring system involves several stages: literature review as a reference collection process, hardware design, which includes the integration of mechanical and electrical components, and software design, which includes program and database creation. Next, system testing is conducted based on predetermined scenarios, followed by evaluation to assess the system's success rate. The next stage is data collection from the test results, which concludes with the preparation of a final report. The research design is shown in Figure 1.

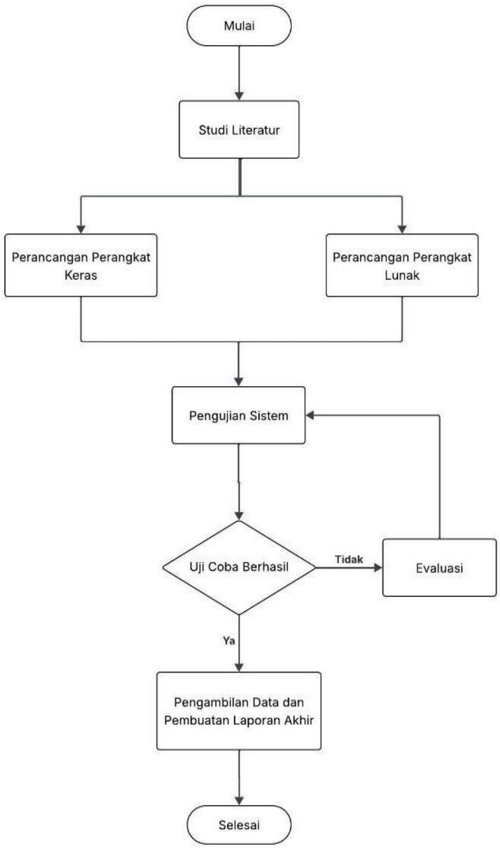


Figure 1. System Design Flowchart

2.1.1 Hardware Design

Hardware design is a key component in developing the system in this study. In the Water Level Monitoring system, the first ESP32 functions as a sensor unit that reads the water level using an ultrasonic sensor installed above a 1 m^3 stainless steel tank in the water treatment room on the third floor. The measurement data is then sent to an MQTT broker on the Antares platform via the internet. The second ESP32 acts as a receiver unit that subscribes to the data and displays it in real-time on an LCD located in the maintenance room on the ground floor. The system is also equipped with a buzzer as a warning indicator when the water level is below the specified minimum limit. With this IoT concept, users can monitor water conditions remotely in real time, thus supporting automatic monitoring and rapid decision-making to prevent the risk of water shortages. The overall system workflow is shown in the hardware design block diagram in Figure 2.

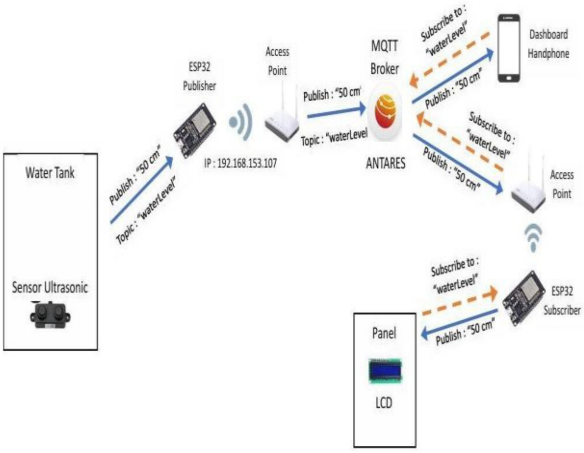


Figure 2. Watertreatment Monitoring Hardware Design

2.1.2 Software Design

The software design for the ESP32-based Water Level Monitoring system and the Antares platform begins with the process of reading ultrasonic sensor data to measure the water level. The height value is obtained by converting the reflected distance of the ultrasonic wave using the water level equation as the difference between the tank height and the sensor distance. The measurement data is then sent every 10 seconds to the Antares platform via the MQTT protocol by the ESP32 as the sending unit. After the data is stored on the server, the receiving ESP32 subscribes to the MQTT topic to obtain data in real-time. The data is then processed and displayed on a 20x4 I2C LCD, followed by buzzer activation if the water level is below the minimum limit of 65 cm. Water level information can be accessed by technicians via the LCD or the Antares application on mobile devices, thus supporting an automated and efficient monitoring system. The software workflow is shown in the design flowchart in Figure 3.

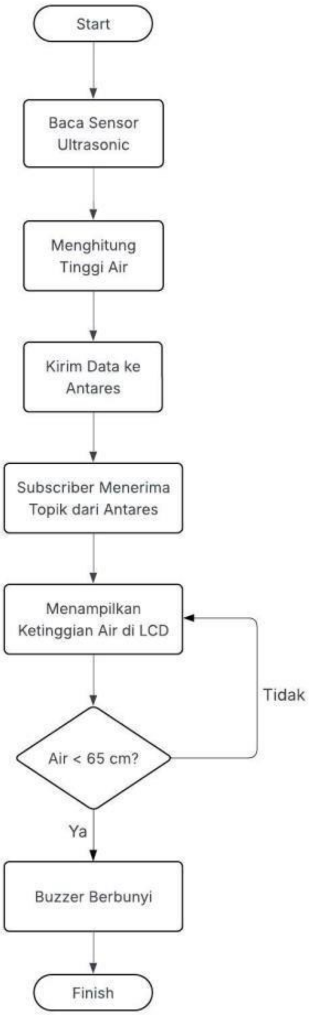


Figure 3. Software Design Flowchart

3. Results and Discussion

Before testing the IoT-based water treatment monitoring system using Antares, environmental conditions were first adjusted to suit the system's requirements. Testing was conducted in a water treatment installation room equipped with a storage tank, a filter unit with three main tubes, and a stainless steel tank as the water treatment process medium, as shown in Figure 4. The focus of the testing was on the stainless steel tank, which was fitted with sensors to directly monitor water conditions. All devices were placed in an easily accessible area to facilitate sensor installation, wiring, and integration with the IoT system. Furthermore, environmental conditions were maintained stable to ensure optimal testing and accurate data representation in the field.

Figure 4. Watertreatment

During the testing phase, an ultrasonic sensor was mounted on the top of a stainless steel tank, as shown in Figure 5, facing directly toward the water surface. This sensor works by emitting high-frequency sound waves, which are then reflected by the water surface and received back by the sensor to calculate their travel time. Based on this reflection time, the distance between the sensor and the water surface can be determined, thereby obtaining the water level value. This method allows for real-time measurements without direct contact with the water, making it safer and more resistant to humid or wet environmental conditions.

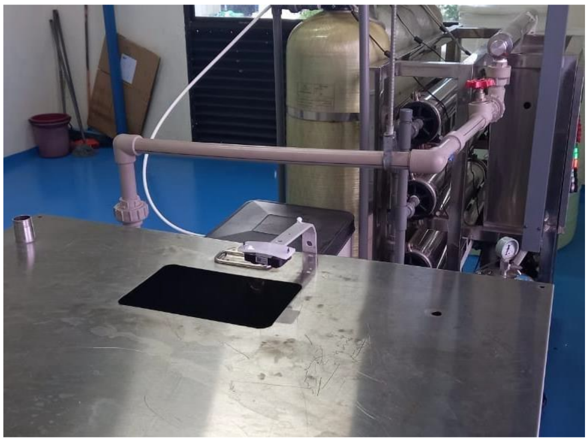


Figure 5. Ultrasonic Sensor Position

In the accuracy and calibration tests, a comparison method was used between the results of ultrasonic sensor readings and manual measurements using a meter, as shown in Figures 6 and 7. Manual measurements were carried out by lowering the meter into a stainless steel tub until it reached the water surface to obtain the actual height value. This value was then compared with the data generated by the ultrasonic sensor and sent to an IoT-based monitoring system. This stage aims to evaluate the level of sensor accuracy and carry out the calibration process if there is a difference in results between manual measurements and the sensor.

Figure 6. Manual Measurement Process

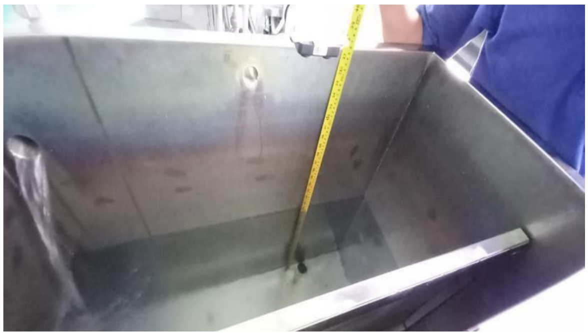


Figure 7. Meter Measuring Instrument

3.1 Testing Data Sending from ESP32 Publisher to ESP32 Subscriber

Data communication testing between the ESP32 Publisher and ESP32 Subscriber was conducted to ensure that data sent by the Publisher can be properly received by the Subscriber via the MQTT protocol on the Antares platform. The monitoring process was carried out using the serial monitor on the Arduino IDE and the I2C LCD display on the ESP32 Subscriber side. The results of the data reception test from the ESP32 Publisher to the ESP32 Subscriber are presented in Table 1.

Table I

Data Delivery Testing

Elemen		No.	Pengukuran Sistem (cm)	Tampilan di LCD (cm)	Status
Pengukuran dilakukan pada ketinggian air 80, 70, 60, 50 dan 40 cm	80	1	80,3	80,3	OK
		2	80,6	80,6	OK
		3	80,3	80,3	OK
		4	80,8	80,8	OK
		5	80,4	80,4	OK
		6	80,9	80,9	OK
		7	80,6	80,6	OK
		8	80,1	80,1	OK
		9	80,6	80,6	OK
		10	80,3	80,3	OK
	70	1	70,3	70,3	OK
		2	70,5	70,5	OK
		3	70,2	70,2	OK
		4	70,4	70,4	OK
		5	70,8	70,8	OK
		6	70,3	70,3	OK
		7	70,6	70,6	OK
		8	70,3	70,3	OK
		9	70,5	70,5	OK
		10	70,4	70,4	OK
	60	1	60,7	60,7	OK
		2	60,9	60,9	OK
		3	60,6	60,6	OK
		4	61,1	61,1	OK
		5	60,8	60,8	OK
		6	60,4	60,4	OK
		7	61	61	OK
		8	60,9	60,9	OK
		9	61,2	61,2	OK
		10	60,7	60,7	OK
	50	1	51	51	OK
		2	50,8	50,8	OK
		3	51,2	51,2	OK
		4	50,6	50,6	OK
		5	50,9	50,9	OK
		6	51,2	51,2	OK
		7	51	51	OK
		8	50,9	50,9	OK
		9	51,1	51,1	OK
		10	50,8	50,8	OK
	40	1	41	41	OK
		2	41,4	41,4	OK
		3	41,2	41,2	OK
		4	41,4	41,4	OK
		5	41	41	OK
		6	41,1	41,1	OK
		7	41,1	41,1	OK
		8	41,3	41,3	OK
		9	41,1	41,1	OK
		10	40,66	40,66	OK

3.4 Alarm Buzzer Warning Testing

The buzzer warning system was tested 10 times with two measurement conditions: at distances below 65 cm and above 65 cm. At distances below 65 cm, the buzzer is expected to be active as a warning indicator, and the test is considered successful if the buzzer emits a sound. Conversely, if there is no sound response, the test is considered a failure. At distances above 65 cm, the buzzer should be inactive, so the test is considered successful if no sound is produced, and failed if the buzzer remains active. The results of the active and inactive buzzer test conditions are presented in Table 2.

Table II

Alarm Buzzer Testing

No.	Pengujian	Kondisi Buzzer	Buzzer Alarm	
			<65 cm (On)	>65 cm (Off)
1	Pengujian 1	On	Berhasil	
		Off		Berhasil
2	Pengujian 2	On	Berhasil	
		Off		Berhasil
3	Pengujian 3	On	Berhasil	
		Off		Berhasil
4	Pengujian 4	On	Berhasil	
		Off		Berhasil
5	Pengujian 5	On	Berhasil	
		Off		Berhasil
6	Pengujian 6	On	Berhasil	
		Off		Berhasil
7	Pengujian 7	On	Berhasil	
		Off		Berhasil
8	Pengujian 8	On	Berhasil	
		Off		Berhasil
9	Pengujian 9	On	Berhasil	
		Off		Berhasil
10	Pengujian 10	On	Berhasil	
		Off		Berhasil
Berhasil berdasrkan kondisi			100%	100%
Total berhasil keseluruhan			100%	

3.4 Water Level Sensor Accuracy and Calibration Testing

Water level sensor accuracy and calibration testing was conducted to ensure compliance between ultrasonic sensor readings and actual field conditions. The testing method was carried out by comparing sensor data with manual measurements using a meter in a stainless steel tank to obtain error values. Next, the relationship between sensor measurement values and actual values was analyzed using linear regression. The regression results in Figure 8 show a coefficient of determination (R^2) close to 1, indicating a high level of consistency in sensor readings. This regression equation was then used as the basis for determining calibration factors to improve system accuracy.

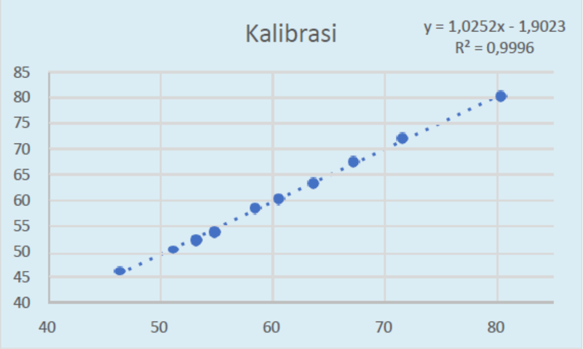


Figure 8. Linear Regression Graph

After applying the regression equation, the comparison results showed that the RMSE value after calibration decreased significantly compared to the pre-calibration condition, indicating an increase in sensor accuracy. This indicates that the calibration method using linear regression is effective in improving the accuracy of the water level measurement system. The results of the RMSE calculation after the calibration process are presented in Table 3.

Table III

Measurement After Calibration

No.	Measuring tool reference (cm)	Sensor Value (cm)	Error (cm)	Error Kuadrat
1	80,7	80,6	-0,1	0,01
2	77,6	77,4	-0,2	0,04
3	75,2	75	-0,2	0,04
4	72,6	72,7	0,1	0,01
5	69,5	69,6	0,1	0,01
6	68,6	68,4	-0,2	0,04
7	65,7	65,9	0,2	0,04
8	63,5	63,5	0	0
9	59,6	59,9	0,3	0,09
10	55,4	55,7	0,3	0,09
Jumlah				0,37
RMSE				0,19

3.4 QoS Testing

Quality of Service (QoS) testing is conducted based on predetermined scenarios using the Wireshark application. The initial testing phase begins by running the application to capture network traffic. This testing aims to obtain QoS parameters, including latency, delay, and packet loss values, as indicators of data communication performance on the system.

3.4.3 Latency Calculation (Delay)

Latency calculations were performed to determine the delay in sending data from the sensor to the server, which was obtained from the time interval between data sent and received based on the capture results in Wireshark. Next, the average delay value was calculated using Microsoft Excel with the average function, as shown in Table 4.

Table IV

Delay Calculation Using Ms. Excel

No.	Time	Source	Destination	Time 1	Time 2	Delay
1	0,00	182.160.55.57	192.168.234.107	0,00	0,25	0,25
2	0,25	192.168.234.107	182.160.55.57	0,25	0,52	0,27
3	0,52	192.168.234.107	182.160.55.57	0,52	0,61	0,09
4	0,61	182.160.55.57	192.168.234.107	0,61	0,61	0,00
5	0,61	182.160.55.57	192.168.234.107	0,61	0,83	0,22
6	0,83	192.168.234.107	182.160.55.57	0,83	0,92	0,09
7	0,92	182.160.55.57	192.168.234.107	0,92	1,15	0,23
8	1,15	192.168.234.107	182.160.55.57	1,15	1,21	0,07
...	...	...	...	...	...	...
289	37,74	182.160.55.57	192.168.234.107	37,74	38,10	0,36
290	38,10	192.168.234.107	182.160.55.57	38,10	38,25	0,16
291	38,25	182.160.55.57	192.168.234.107	38,25	38,26	0,01
292	38,26	182.160.55.57	192.168.234.107	38,26	38,26	0,00
293	38,26	182.160.55.57	192.168.234.107	38,26	38,36	0,10
294	38,36	192.168.234.107	182.160.55.57	38,36	38,48	0,13
295	38,48	182.160.55.57	192.168.234.107	38,48	38,48	0,00
296	38,48	182.160.55.57	192.168.234.107	38,48	38,49	0,01
297	38,49	192.168.234.107	182.160.55.57	38,49	38,59	0,10
298	38,59	182.160.55.57	192.168.234.107	38,59	38,62	0,03
299	38,62	192.168.234.107	182.160.55.57	38,62		
Total Delay						38,62
Rata-rata Delay						129,15

Based on the results of observations with Wireshark on the IP address and protocol that have been set, the latency (delay) value is obtained which is calculated using the following formula:

$$Latency = \frac{\sum_{i=1}^{N} (t_{terima,i} - t_{kirim,i})}{N} \times 1000 \qquad (1)$$

$$Latency = \frac{38{,}62}{299} \times 1000$$

$$Latency = 129{,}15\ ms$$

3.4.3 Jitter Calculation

Jitter is the variation in delay between data packets, with high values indicating irregular transmission and potentially impacting connection stability. The average jitter value was calculated using Microsoft Excel's average function, as shown in Table 5.

Table V

Jitter Calculation Using Ms.Excel

No.	Time	Source	Destination	Delay 1	Delay 2	Jitter
1	0,00	182.160.55.57	192.168.234.107	0,25	0,27	0,02
2	0,25	192.168.234.107	182.160.55.57	0,27	0,09	-0,18
3	0,52	192.168.234.107	182.160.55.57	0,09	0,00	-0,09
4	0,61	182.160.55.57	192.168.234.107	0,00	0,22	0,22
5	0,61	182.160.55.57	192.168.234.107	0,22	0,09	-0,13
6	0,83	192.168.234.107	182.160.55.57	0,09	0,23	0,14
7	0,92	182.160.55.57	192.168.234.107	0,23	0,07	-0,16
8	1,15	192.168.234.107	182.160.55.57	0,07	0,00	-0,07
...	...	...	...	...	...	...
289	37,74	182.160.55.57	192.168.234.107	0,36	0,16	-0,20
290	38,10	192.168.234.107	182.160.55.57	0,16	0,01	-0,15
291	38,25	182.160.55.57	192.168.234.107	0,01	0,00	-0,01
292	38,26	182.160.55.57	192.168.234.107	0,00	0,10	0,10
293	38,26	182.160.55.57	192.168.234.107	0,10	0,13	0,03
294	38,36	192.168.234.107	182.160.55.57	0,13	0,00	-0,13
295	38,48	182.160.55.57	192.168.234.107	0,00	0,01	0,01
296	38,48	182.160.55.57	192.168.234.107	0,01	0,10	0,09
297	38,49	192.168.234.107	182.160.55.57	0,10	0,03	-0,07
298	38,59	182.160.55.57	192.168.234.107	0,03		
299	38,62	192.168.234.107	182.160.55.57			
Total Jitter						0,22
Rata-rata Jitter						0,736

From the results of the analysis using Wireshark on the specified IP address and protocol, the jitter value is obtained which is then calculated using the following formula:

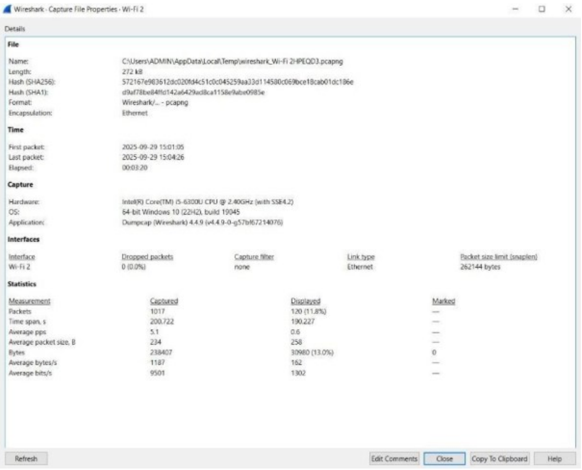
$$Jitter = \frac{\sum_{i=1}^{N} (D_{2,i} - D_{1,i})}{N} \times 1000 \qquad (2)$$

$$Jitter = \frac{0{,}22}{299} \times 1000$$

$$Jitter = 0{,}736\ ms$$

3.4.3 Packet Loss Calculation

The data analysis process was performed using the Wireshark application by first running a capture process on the network. After the capture process was completed, the "mqtt" filter was applied to the filter column to display data packets using the MQTT protocol. This step aims to focus the analysis only on relevant data traffic, as shown in Figure 9.



File

Name: C:\Users\ADMIN\AppData\Local\Temp\wireshark_Wi-Fi 2HPEQD3.pcapng
Length: 272 kB
Hash (SHA256): 572167e983612dc020fd4c51c0c045259aa33d114580c069bce18cab01dc186e
Hash (SHA1): d9af78be84ffd142a6429ad8ca1158e9abe0985e
Format: Wireshark/... - pcapng
Encapsulation: Ethernet

Time

First packet: 2025-09-29 15:01:05
Last packet: 2025-09-29 15:04:26
Elapsed: 00:03:20

Capture

Hardware: Intel(R) Core(TM) i5-6300U CPU @ 2.40GHz (with SSE4.2)
OS: 64-bit Windows 10 (22H2), build 19045
Application: Dumpcap (Wireshark) 4.4.9 (v4.4.9-0-g57bf67214076)

Interfaces

Interface	Dropped packets	Capture filter	Link type	Packet size limit (snaplen)
Wi-Fi 2	0 (0.0%)	none	Ethernet	262144 bytes

Statistics

Measurement	Captured	Displayed	Marked
Packets	1017	120 (11.8%)	—
Time span, s	200.722	190.227	—
Average pps	5.1	0.6	—
Average packet size, B	234	258	—
Bytes	238407	30980 (13.0%)	0
Average bytes/s	1187	162	—
Average bits/s	9501	1302	—

Figure 9. Packet Loss Value in Wireshark

To calculate packet loss, a formula is used that compares the number of data packets successfully received to the number of packets sent. This calculation aims to determine the percentage of data loss during the transmission process. Packet loss is an important parameter in evaluating network quality because it is directly related to the reliability of data communication. The lower the packet loss, the better the performance of the communication system used. To calculate packet loss, you can use the formula.

$$Packet\ Loss = \left(\frac{N_{Dikirim} - N_{Diterima}}{N_{Dikirim}}\right) \times 100\% \qquad (3)$$

$$Packet\ Loss = \left(\frac{120 - 120}{120}\right) \times 100\%$$

$$Packet\ Loss = 0\%$$

4. Conclusion

Based on the test results that have been carried out, the IoT-based Water Level Monitoring system is able to operate well and reliably in various test aspects. The data communication process between the ESP32 Publisher and ESP32 Subscriber via the MQTT protocol on the Antares platform shows a 100% success rate, with data matching between the serial monitor display and the I2C LCD without any differences. The buzzer warning system also functions according to the designed logic, with a 100% success rate in both active and inactive conditions based on a threshold of 65 cm. In addition, the calibration results using the RMSE method show an increase in sensor accuracy, where the displayed water level data is getting closer to the results of manual measurements. In terms of network performance, QoS analysis shows excellent results with a packet loss value of 0%, an average delay of 129.15 ms, and jitter of 0.736 ms. Thus, the developed system is proven to be effective, accurate, and stable, making it suitable for application as an IoT-based water level monitoring solution in industrial environments.

Acknowledgments

The author would like to thank all parties who have contributed and provided support in the implementation of this research, especially PT. Simatelex Manufactory Batam, for the facilities and opportunities provided during the research process until the preparation of the final report.

Referensi

[1] Y. Ratmini, V. Atina, and E. Purwanto, "Sistem Monitoring dan Peringatan Dini Banjir Berbasis Internet of Things (IoT)," *Jurnal Ilmiah Teknologi Informasi Asia*, vol. 19, no. 1, pp. 1–7, Feb. 2025, doi: 10.32815/jitika.v19i1.1103.

[2] K. Fajri, I. Santoso, and T. Prakoso, "ANALISIS KINERJA TRANSMISI DATA ANTAR NODE DAN NODE KE SERVER DENGAN PARAMETER QOS PADA JARINGAN SENSOR NIRKABEL."

[3] N. Syamsi, M. Ulfah, and D. Lesmideyarti, "Water Level Monitoring for Flood Early Mitigation Based on Internet of Things (IoT)," *TEPIAN*, vol. 5, no. 2, pp. 58–64, Jun. 2024, doi: 10.51967/tepian.v5i2.2504.

[4] I. Bagus, M. Lingga Pradirta, I. Nyoman Piarsa, I. Putu, A. Dharmaadi, and P. Korespondensi, "SISTEM PENDETEKSI BANJIR DAN BADAI ANGIN SERTA MONITORING CUACA BERBASIS INTERNET OF THINGS", doi: 10.25126/jtiik.202295983.

[5] Noptin Harpawi and Yogi Syaputra, "Monitoring Kualitas Udara dan Kontrol Air PurifierT," vol. 8, May 2022.

[6] I. Noval, A. Prahara, and I. R. Widiasari, "Implementasi Metode Received Signal Strength Indication dan Quality of Service Terhadap Analisis Kualitas Jaringan Wireless di CV Java Media Perdana Pati," *Jurnal Teknologi Informasi dan Komunikasi)*, vol. 7, no. 4, p. 2023, 2023, doi: 10.35870/jti.

[7] I. Sari Kusuma Wardhana, B. Agus Wardjiono, S. Jakarta STI, and S. Komputer, "Analisis Pengiriman Data Sensor dengan Jaringan Wireless Meggunakan Metode Quality of Service (QoS)," vol. 5, no. 2, pp. 371–383, 2022, doi: 10.31764/justek.vXiY.ZZZ.

[8] A. Fatakhunnaim[1], A. E. Jayati[2], and P. Muliandhi[3], "Analisis Kualitas Jaringan Wi-Fi di Lantai 7 Gedung Menara USM Menggunakan Ekahau Site Survey."

[9] M. K. Rihmi, G. Bintoro, M. A. Rahman, G. Puspito, and A. Muntaha, "ACCURACY ANALYSIS OF DISTANCE MEASUREMENT USING SONAR ULTRASONIC SENSOR HC-SR04 ON SEVERAL TYPES OF MATERIALS," *Journal of Environmental Engineering & Sustainable Technology JEEST*, vol. 11, no. 01, pp. 10–13, 2024, [Online]. Available: http://jeest.ub.ac.id