

Analisis Kinerja Protokol MQTT pada Sistem IoT Monitoring Tanaman

Performance Analysis of MQTT Protocol in IoT Plant Monitoring System Based

Elisa Soleha^{a,1}, Linna Oktaviana Sari^{b,2}

^{a,b}Program Studi Teknik Informatika, Jurusan Teknik Elektro, Fakultas Teknik, Universitas Riau
Jl. HR. Subrantas KM. 12.5, Pekanbaru, 28293, Indonesia
elisa.soleha1387@student.unri.ac.id¹, linnaosari@lecturer.unri.ac.id²

ABSTRAK

Penelitian ini menganalisis pengaruh variasi tingkat Quality of Service (QoS) 0, 1, dan 2 pada protokol MQTT terhadap parameter kinerja jaringan pada sistem IoT monitoring tanaman cabai rawit. Penelitian ini menggunakan metode eksperimental dengan pendekatan kuantitatif dalam lingkungan jaringan WiFi lokal yang terkontrol. Sistem IoT dibangun menggunakan mikrokontroler ESP32 yang dilengkapi dengan sensor DHT22 (suhu dan kelembapan udara), sensor kelembapan tanah (capacitive soil moisture), dan sensor intensitas cahaya GY-2561, yang dihubungkan ke broker MQTT Mosquitto melalui protokol MQTT. Parameter kinerja jaringan yang diukur meliputi delay, throughput, packet loss, bandwidth consumption, dan jitter menggunakan perangkat lunak Wireshark. Setiap skenario QoS dijalankan selama 30 menit dengan interval pengiriman data setiap 5 detik, menghasilkan minimal 360 sampel paket per tingkat QoS. Hasil pengujian menunjukkan bahwa QoS 0 memiliki delay rata-rata tertinggi (2.609,70 ms), QoS 1 dengan delay 2.488,51 ms, sedangkan QoS 2 menghasilkan delay terendah (1.246,76 ms). Nilai jitter menunjukkan QoS 2 paling stabil (130,07 ms), QoS 0 (190,04 ms), dan QoS 1 tertinggi (276,50 ms). Packet loss pada semua QoS berada di bawah 1%, dengan QoS 2 paling rendah. Bandwidth consumption meningkat seiring level QoS: QoS 0 (0,31 Kbps), QoS 1 (0,32 Kbps), dan QoS 2 (0,51 Kbps). Throughput konsisten di semua level (0,23 Kbps). Penelitian ini memberikan rekomendasi konfigurasi QoS berdasarkan kebutuhan sistem: QoS 0 untuk efisiensi bandwidth, QoS 1 untuk keseimbangan keandalan-efisiensi, dan QoS 2 untuk keandalan maksimal. Kebaruan penelitian ini terletak pada analisis pengaruh variasi QoS 0, 1, dan 2 secara komprehensif terhadap lima parameter kinerja jaringan sekaligus pada sistem IoT monitoring tanaman berbasis jaringan WiFi lokal, melengkapi penelitian sebelumnya yang umumnya hanya mengkaji sebagian parameter atau hanya mengimplementasikan sistem IoT monitoring tanaman saja.

Kata Kunci: MQTT, QoS, Kinerja Jaringan, IoT Monitoring Tanaman, Delay, Throughput, Packet Loss

ABSTRACT

This study analyzes the effect of varying Quality of Service (QoS) levels 0, 1, and 2 in the MQTT protocol on network performance parameters in an IoT monitoring system for chili plants. The research uses an experimental method with a quantitative approach in a controlled local WiFi network environment. The IoT system is built using an ESP32 microcontroller equipped with a DHT22 sensor (temperature and air humidity), a capacitive soil moisture sensor, and a GY-2561 light intensity sensor, connected to a Mosquitto MQTT broker. Network performance parameters measured include delay, throughput, packet loss, bandwidth consumption, and jitter using Wireshark software. Each QoS scenario was run for 30 minutes with data transmission intervals of 5 seconds, generating a minimum of 360 packet samples per QoS level. Test results show that QoS 0 has the highest average delay (2,609.70 ms), QoS 1 with 2,488.51 ms, while QoS 2 produces the lowest delay (1,246.76 ms). Jitter values indicate QoS 2 is most stable (130.07 ms), QoS 0 (190.04 ms), and QoS 1 highest (276.50 ms). Packet loss across all QoS levels is below 1%, with QoS 2 being lowest. Bandwidth consumption increases with QoS level: QoS 0 (0.31 Kbps), QoS 1 (0.32 Kbps), and QoS 2 (0.51 Kbps). Throughput is consistent across all levels (0.23 Kbps). This study provides QoS configuration recommendations based on system requirements: QoS 0 for bandwidth efficiency, QoS 1 for reliability-efficiency balance, and QoS 2 for maximum reliability. The novelty of this research lies in the comprehensive analysis of the effect of QoS 0, 1, and 2 variations on five network performance parameters simultaneously in an IoT plant monitoring system based on a local WiFi

network, complementing previous studies that generally only examined partial parameters or merely implemented an IoT plant monitoring system without in-depth network performance analysis.

Keywords: MQTT, QoS, Network Performance, IoT Plant Monitoring, Delay, Throughput, Packet Loss

Disubmit:

Info Artikel :
Direview:

Diterima :

Copyright © 2026 – CSRID Journal. All rights reserved.

1. PENDAHULUAN

Internet of Things (IoT) telah menjadi teknologi utama dalam otomasi dan pengawasan sistem pertanian modern. Salah satu penerapan penting IoT adalah sistem monitoring tanaman yang dapat mengawasi kondisi lingkungan seperti suhu, kelembapan tanah, dan intensitas cahaya secara real-time. Dengan sistem seperti ini, keputusan terkait penyiraman, pemupukan, dan perawatan tanaman dapat dilakukan secara lebih efisien dan berbasis data. Namun, kunci keberhasilan sistem IoT bukan hanya pada sensor yang digunakan, tetapi juga pada kinerja protokol komunikasi data yang menjadi penghubung antarperangkat.

Berbagai protokol IoT seperti CoAP, AMQP, dan HTTP telah dikembangkan, namun MQTT dipilih sebagai fokus penelitian ini karena sifatnya yang ringan, hemat bandwidth, dan mendukung komunikasi real-time melalui konsep publish-subscribe yang dirancang khusus untuk perangkat dengan sumber daya terbatas seperti ESP32. MQTT juga memiliki fleksibilitas dalam QoS yang memungkinkannya menyesuaikan trade-off antara kecepatan dan keandalan data pada jaringan lokal pertanian yang sering kali tidak stabil. Hal ini didukung oleh penelitian yang membuktikan bahwa protokol MQTT memiliki performa pengiriman data yang lebih baik dibandingkan HTTP pada sistem smart metering, khususnya dalam efisiensi kecepatan dan keandalan transmisi data [16]. Keunggulan inilah yang membuatnya lebih cocok untuk sistem monitoring dengan perangkat bersumber daya rendah seperti ESP32 dan jaringan WiFi lokal [1].

Tanaman cabai rawit memerlukan pemantauan yang ketat terhadap kondisi lingkungan untuk menghindari stres lingkungan baik itu kekurangan air maupun kelebihan air, fluktuasi suhu ekstrim, dan ketidaksesuaian intensitas cahaya untuk fotosintesis. Cabai rawit membutuhkan kelembapan tanah pada kisaran 50–70% agar tidak mengalami kondisi kekeringan maupun kelebihan air, dengan pH tanah optimal antara 6–7 serta suhu lingkungan berkisar 24–28 °C. Selain kelembapan tanah dan suhu lingkungan, intensitas cahaya merupakan faktor lingkungan penting yang sangat memengaruhi proses fotosintesis dan pertumbuhan tanaman cabai rawit, dengan kebutuhan intensitas cahaya antara 12.000–30.000 Lux untuk mendukung proses fotosintesis yang optimal [2]. Oleh karena itu, pemantauan intensitas cahaya menjadi aspek yang tidak terpisahkan dalam sistem monitoring budidaya tanaman cabai rawit di dalam greenhouse [15].

Beberapa penelitian terdahulu telah menganalisis kinerja protokol MQTT pada berbagai konteks IoT. Berbagai penelitian terdahulu yang melakukan perbandingan performa MQTT dan CoAP pada sistem hidroponik tanaman stroberi dan menemukan bahwa MQTT unggul dalam packet loss dan keandalan pesan, namun tidak membahas pengaruh variasi QoS terhadap performa komunikasi secara mendalam[3]. Ada juga yang melakukan evaluasi performa MQTT konfigurasi QoS 0, 1, dan 2, tetapi hanya menganalisis parameter konsumsi daya menggunakan INA219, latency, dan packet loss [4]. Penelitian lain menganalisis performansi protokol MQTT pada sistem pemantauan kualitas udara ruangan berbasis IoT dengan menguji ketiga level QoS, namun tidak mengkaji parameter bandwidth consumption dan jitter serta tidak diterapkan pada konteks monitoring tanaman [17]. Dan juga yang meneliti pengaruh jumlah publisher terhadap throughput dan delay[5], tetapi tidak menganalisis bagaimana variasi QoS memengaruhi performa komunikasi terhadap lima parameter kinerja jaringan

yaitu delay, throughput, packet loss, bandwidth consumption, dan jitter. Ada pula yang menggunakan platform cloud berbasis internet publik sehingga tidak mencerminkan kinerja murni protokol MQTT pada jaringan lokal [13]. Selain itu, terdapat penelitian yang mengimplementasikan dan membandingkan protokol MQTT dan HTTP pada sistem pendeteksi banjir, namun hanya mengukur tiga parameter QoS yaitu throughput, packet loss, dan delay tanpa menyertakan bandwidth consumption dan jitter [18]. Penelitian lain berfokus pada perancangan fungsionalitas sistem monitoring aeroponik dan efisiensi daya tanpa menganalisis pengaruh variasi QoS terhadap parameter kinerja jaringan [14]. Selain itu, ada yang hanya membahas implementasi sistem monitoring intensitas cahaya pada tanaman tanpa melakukan analisis kinerja jaringan terhadap variasi level QoS [15].

Berdasarkan kajian literatur tersebut, dapat disimpulkan bahwa sebagian besar penelitian terdahulu masih berfokus pada aspek perancangan sistem atau perbandingan antar protokol yang hanya mengkaji beberapa parameter QoS terhadap performa komunikasi MQTT dalam konteks pertanian. Celah penelitian ini (research gap) penting karena konfigurasi QoS berpengaruh langsung terhadap efisiensi transmisi, keandalan komunikasi, serta konsistensi data sensor dalam kondisi jaringan yang tidak selalu stabil di lingkungan pertanian. Oleh karena itu, penelitian ini berupaya mengisi kekosongan tersebut dengan melakukan analisis kinerja jaringan secara empiris terhadap pengaruh variasi QoS 0, 1, dan 2 terhadap parameter kinerja jaringan yang lebih komprehensif meliputi delay, throughput, packet loss, bandwidth consumption, dan jitter yang diterapkan langsung pada sistem IoT monitoring tanaman cabai rawit.

Tujuan penelitian ini adalah menganalisis pengaruh variasi tingkat QoS 0, 1, dan 2 pada protokol MQTT terhadap parameter kinerja jaringan yang meliputi delay, throughput, packet loss, bandwidth consumption, dan jitter pada sistem IoT monitoring tanaman cabai rawit. Penelitian ini diharapkan dapat memberikan rekomendasi konfigurasi QoS yang mampu mengoptimalkan kinerja jaringan sistem IoT monitoring tanaman dengan mempertimbangkan keseimbangan antara keandalan data dan efisiensi penggunaan bandwidth.

A. IoT (Internet of Things)

Internet of Things (IoT) merupakan paradigma yang menghubungkan perangkat fisik ke internet untuk saling bertukar data dan informasi. Dalam konteks pertanian, IoT berperan penting dalam smart farming melalui pemantauan otomatis terhadap kondisi lingkungan tanaman. IoT adalah konsep dimana berbagai perangkat fisik seperti sensor, aktuator, alat elektronik, kendaraan, hingga peralatan rumah tangga terhubung ke internet dan saling bertukar data tanpa campur tangan manusia secara langsung. Perangkat-perangkat tersebut dilengkapi dengan sensor, software, dan konektivitas jaringan sehingga mampu mengumpulkan data dari lingkungan sekitarnya, mengirimkan data tersebut ke sistem lain melalui internet, dan menerima instruksi atau perintah dari sistem secara otomatis [6].

Arsitektur Internet of Things (IoT) pada umumnya terdiri dari tiga lapisan utama, yaitu perception layer, network layer, dan application layer. Perception layer berfungsi sebagai lapisan penginderaan yang bertugas mengumpulkan data lingkungan menggunakan berbagai sensor, seperti sensor kelembapan tanah, suhu, dan kelembapan udara, yang banyak diterapkan pada sistem monitoring tanaman dan pertanian pintar. Network layer bertanggung jawab menghubungkan perangkat IoT dengan sistem pemrosesan data melalui jaringan komunikasi. Pada lapisan ini, protokol MQTT sering digunakan karena ringan dan mendukung mekanisme publish-subscribe, serta menyediakan pengaturan Quality of Service (QoS) untuk menjamin keandalan pengiriman data. Application layer berperan dalam menyajikan data hasil pemantauan kepada pengguna dalam bentuk antarmuka aplikasi, dashboard web, atau notifikasi real-time, sehingga pengguna dapat memantau kondisi tanaman secara efektif [7].

B. Protokol MQTT

MQTT (*Message Queuing Telemetry Transport*) adalah protokol komunikasi ringan berbasis TCP/IP yang menggunakan arsitektur publish-subscribe. MQTT adalah protokol pesan berbasis standar yang digunakan untuk komunikasi mesin-ke-mesin. Sensor pintar, perangkat yang dapat dikenakan, dan perangkat IoT lainnya biasanya harus mengirim dan menerima data melalui jaringan dengan sumber daya dan bandwidth terbatas. Protokol ini dirancang untuk komunikasi mesin-ke-mesin dengan kebutuhan bandwidth terbatas dan cocok digunakan pada perangkat IoT dengan keterbatasan sumber daya [8].

C. *Quality of Service (QoS) pada MQTT*

Protokol MQTT menyediakan tiga level Quality of Service (QoS) yang menentukan mekanisme pengiriman pesan dan tingkat keandalan komunikasi data. Setiap level QoS memiliki karakteristik dan trade-off yang berbeda antara kecepatan, keandalan, dan penggunaan bandwidth.

QoS 0 (*At most once*) adalah QoS yang datanya dikirim tanpa konfirmasi. Merupakan *level* tercepat namun tidak ada jaminan pesan sampai ke tujuan. Pesan dikirimkan oleh publisher tanpa menunggu konfirmasi dari broker. Metode ini menawarkan pengiriman tercepat dan beban jaringan terendah, namun memiliki risiko kehilangan data terbesar jika koneksi terputus. Pada QoS 0, publisher mengirim paket PUBLISH ke broker tanpa mekanisme acknowledgment, sehingga jika terjadi gangguan jaringan sesaat, paket dapat hilang tanpa ada pengiriman ulang [9].

QoS 1 (*At least once*) adalah *level* QoS pada protokol MQTT menjamin pesan sampai ke broker setidaknya satu kali. Publisher akan menunggu pesan konfirmasi (PUBACK) dari broker. Jika konfirmasi tidak diterima dalam waktu tertentu, pesan akan dikirim ulang. Hal ini menjamin data sampai, namun berpotensi menyebabkan duplikasi pesan di sisi penerima jika PUBACK terlambat atau hilang dalam perjalanan. Mekanisme ini melibatkan pertukaran dua paket: PUBLISH dari publisher dan PUBACK dari broker, yang menambah overhead komunikasi namun memberikan jaminan keandalan yang lebih baik [4].

QoS 2 (*Exactly once*) adalah *level* QoS dengan tingkat tertinggi yang menjamin pesan diterima tepat satu kali tanpa duplikasi. Proses ini melibatkan mekanisme four-way handshake yang terdiri dari empat tahap: publisher mengirim PUBLISH, broker merespons dengan PUBREC, publisher mengirim PUBREL, dan broker mengonfirmasi dengan PUBCOMP. Meskipun memberikan keandalan tertinggi dan menghilangkan kemungkinan duplikasi, QoS 2 menghasilkan overhead komunikasi terbesar karena memerlukan empat paket untuk setiap pengiriman pesan, sehingga mengonsumsi bandwidth lebih besar dan menghasilkan latensi lebih tinggi dibandingkan QoS 0 dan QoS 1 [4].

D. *Parameter Kinerja Jaringan*

Delay (Latensi) adalah waktu yang dibutuhkan paket data untuk melakukan perjalanan dari perangkat pengirim ke penerima melalui broker MQTT. *Delay* merupakan salah satu parameter kritis dalam sistem *monitoring real-time* karena berpengaruh langsung terhadap kecepatan respon sistem terhadap perubahan kondisi lingkungan. Nilai *delay* dihitung berdasarkan selisih waktu antara timestamp pengiriman paket dan timestamp penerimaan paket. Standar TIPHON mengkategorikan *delay* sebagai: Sangat Bagus (< 150 ms), Bagus (150–300 ms), Sedang (300–450 ms), dan Jelek (> 450 ms). Dalam konteks sistem IoT *monitoring* tanaman, *delay* yang rendah memastikan data sensor dapat ditampilkan secara real-time dan keputusan dapat diambil dengan cepat [10].

Jitter adalah ketidakstabilan atau variasi waktu kedatangan paket data yang diterima oleh perangkat tujuan. *Jitter* yang tinggi dapat menyebabkan tampilan data tidak sinkron pada sistem monitoring real-time dan mengganggu konsistensi visualisasi data. *Jitter* dihitung sebagai rata-rata selisih absolut antara *delay* paket berturut-turut. Standar TIPHON mengkategorikan *jitter* sebagai: Sangat Bagus (0 ms), Bagus (0–75 ms), Sedang (75–125 ms), dan Jelek (125–225 ms). *Jitter* yang rendah sangat penting untuk

memastikan data sensor ditampilkan secara konsisten dan dapat dipercaya untuk pengambilan keputusan [10].

Packet Loss adalah persentase paket data yang hilang selama transmisi dan tidak pernah sampai ke tujuan. Packet loss yang tinggi menyebabkan data monitoring tidak lengkap dan dapat mengakibatkan kesalahan dalam pengambilan keputusan. Packet loss dihitung dengan membandingkan jumlah paket yang hilang terhadap total paket yang dikirimkan. Standar TIPHON mengkategorikan packet loss sebagai: Sangat Bagus (0%), Bagus ($\approx 3\%$), Sedang ($\approx 15\%$), dan Jelek ($> 25\%$). Dalam sistem monitoring tanaman, packet loss yang tinggi dapat menyebabkan hilangnya informasi penting tentang kondisi lingkungan yang dapat berdampak pada kesehatan tanaman [10].

Throughput adalah tingkat transfer data efektif yang berhasil dikirimkan melalui jaringan dalam satuan waktu tertentu, umumnya dinyatakan dalam bit per detik (bps) atau Kilobit per detik (Kbps). *Throughput* menunjukkan seberapa banyak data payload aktual yang dapat ditangani oleh sistem secara efisien, tidak termasuk overhead protokol. Throughput yang tinggi menandakan sistem mampu mengirimkan data sensor dalam jumlah besar secara efisien. Standar TIPHON mengkategorikan throughput sebagai: Sangat Bagus (≥ 100 bps), Bagus (≥ 75 bps), Sedang (≥ 50 bps), dan Buruk (< 25 bps). Dalam konteks IoT monitoring tanaman, throughput yang memadai memastikan semua data sensor dapat ditransmisikan tanpa bottleneck [10].

Bandwidth Consumption adalah Jumlah total *bandwidth* yang digunakan untuk mengirimkan data dalam satuan waktu tertentu, termasuk overhead protokol (header, paket kontrol, acknowledgment). Bandwidth consumption berbeda dengan throughput karena mencakup semua paket yang dikirimkan, bukan hanya payload data. Parameter ini penting untuk mengevaluasi efisiensi penggunaan jaringan, terutama pada sistem IoT yang beroperasi pada jaringan dengan bandwidth terbatas. *Bandwidth consumption* yang tinggi dapat mengindikasikan overhead protokol yang besar atau pengiriman paket kontrol yang berlebihan. Dalam sistem monitoring tanaman yang menggunakan jaringan WiFi lokal, efisiensi bandwidth sangat penting untuk memastikan stabilitas koneksi dan menghindari kongesti jaringan [11].

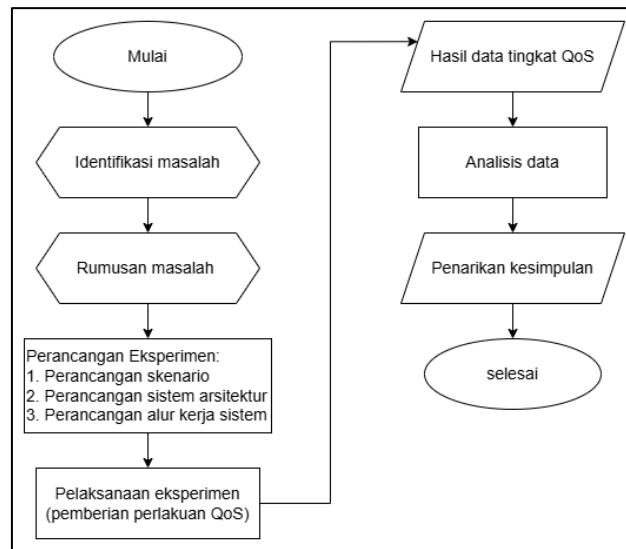
E. Sistem IoT Monitoring Tanaman

Sistem IoT *monitoring* tanaman merupakan penerapan teknologi Internet of Things untuk mengamati kondisi lingkungan dan pertumbuhan tanaman secara real-time. Sistem ini dirancang untuk melakukan pemantauan berkelanjutan terhadap parameter lingkungan seperti suhu, kelembapan udara, kelembapan tanah, dan intensitas cahaya. Data sensor diproses oleh mikrokontroler ESP32 dan dikirimkan ke platform monitoring menggunakan protokol MQTT melalui broker ke Node-RED sebagai dashboard visualisasi.

Karakteristik sistem IoT monitoring tanaman yang relevan dengan kinerja jaringan meliputi: (1) kebutuhan pengukuran data secara berkelanjutan (continuous monitoring); (2) transmisi data real-time yang membutuhkan latensi rendah dan jitter minimal; (3) berbasis perangkat rendah daya (low-power devices) dengan keterbatasan sumber daya; dan (4) ketergantungan pada jaringan nirkabel yang sensitif terhadap interferensi. Kondisi-kondisi ini menjadikan pemilihan konfigurasi QoS yang tepat sangat penting untuk memastikan sistem dapat beroperasi secara optimal [12].

2. METODE

Penelitian ini menggunakan metode eksperimental dengan pendekatan kuantitatif untuk menganalisis pengaruh variasi tingkat QoS (0, 1, dan 2) pada protokol MQTT terhadap parameter kinerja jaringan sistem IoT monitoring tanaman. Alur tahapan metode eksperimental penelitian ini divisualisasikan sebagai pada Gambar 1 berikut.

Gambar 1. *Flowchart Alur Penelitian.*

Penelitian ini dirancang menggunakan skema one-factor experiment, di mana hanya satu faktor yang dimanipulasi yaitu tingkat QoS, sementara variabel lain dikontrol secara ketat. Setiap tingkat QoS diuji secara terpisah dengan menggunakan konfigurasi sistem yang sama untuk memastikan hasil yang dapat dibandingkan secara objektif. pengujian dilakukan dalam lingkungan laboratorium yang terkontrol untuk meminimalkan pengaruh faktor eksternal yang tidak terprediksi. Semua pengujian dilakukan pada jaringan WiFi lokal yang sama dengan konfigurasi perangkat keras dan perangkat lunak yang identik. Durasi pengujian untuk setiap skenario QoS dibuat sama, yaitu 30 menit dengan interval pengiriman data setiap 5 detik. Setiap skenario diulang sebanyak 3 kali pada waktu yang berbeda (pagi, siang, malam) untuk mendapatkan data yang representatif dan meminimalkan bias akibat variasi kondisi jaringan pada waktu yang berbeda.

A. Identifikasi Variabel

Variabel Bebas pada penelitian ini adalah Tingkat Quality of Service pada protokol MQTT, yang terdiri dari Quality of Service 0, 1, dan 2. Variabel ini dimanipulasi secara sistematis untuk mengamati pengaruhnya terhadap kinerja jaringan.

Variabel Terikat pada penelitian ini adalah parameter kinerja jaringan yang meliputi delay (latensi), throughput, packet loss, bandwidth consumption, dan jitter. Parameter-parameter ini diukur untuk setiap level QoS menggunakan perangkat lunak Wireshark.

Variabel pada penelitian ini juga di kontrol untuk memastikan bahwa perubahan variabel terikat hanya dipengaruhi oleh variasi QoS, beberapa variabel lain dikontrol sebagai berikut: (1) interval pengiriman data sensor (publish interval) dibuat tetap setiap 5 detik; (2) ukuran payload MQTT dibuat sama untuk setiap skenario; (3) jumlah publisher dan subscriber tidak berubah; (4) topik MQTT yang digunakan sama untuk semua pengujian; (5) broker MQTT (*Mosquitto*) dan perangkat keras yang digunakan tetap sama; (6) pengujian dilakukan pada jaringan WiFi lokal yang sama; dan (7) durasi pengujian untuk setiap skenario QoS dibuat sama.

B. Perangkat yang Digunakan.

Perangkat yang digunakan pada penelitian ini dapat dilihat pada Tabel 1 sebagai berikut.

Tabel 1. Perangkat Keras yang Digunakan.

No	Perangkat	Fungsi
1	ESP32	Mikrokontroler Publisher IoT
2	Sensor DHT22	Sensor suhu dan kelembapan udara
3	Capacitive Soil Moisture Sensor	Sensor kelembapan tanah
4	GY-2561	Sensor intensitas cahaya
5	Komputer Lokal (Server Broker)	Server MQTT Broker Mosquitto

Perangkat lunak yang digunakan dalam penelitian ini dapat dilihat pada Tabel 2 sebagai berikut.

Tabel 2. Perangkat Lunak yang Digunakan.

No	Perangkat Lunak	Fungsi
1	Mosquitto MQTT Broker	Broker MQTT pada server lokal
2	Wireshark	Capture dan analisis paket jaringan
3	Arduino IDE	Pemrograman mikrokontroler ESP32
4	Node-RED	Dashboard visualisasi data monitoring

C. Perancangan Skenario Pengujian

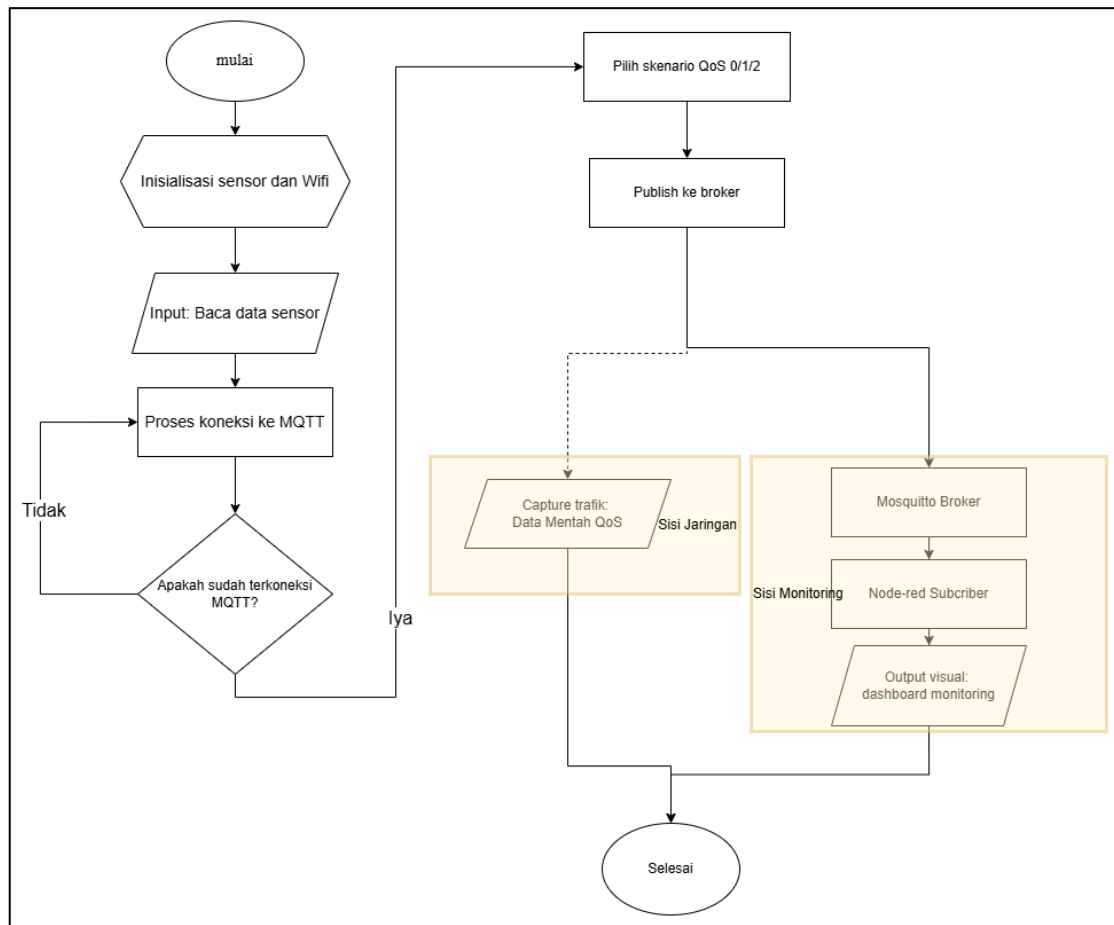
Skenario pengujian difokuskan pada observasi lalu lintas data di jalur transmisi antara Publisher (perangkat IoT) dan Broker. Sistem diatur pada topologi jaringan lokal dengan satu broker dan satu node publisher. Data yang ditransmisikan merupakan representasi kondisi lingkungan tanaman cabai rawit yang meliputi nilai analog dari sensor kelembapan tanah serta nilai digital suhu dan kelembapan udara. Data dikemas dalam format payload JSON dengan ukuran yang seragam untuk memastikan beban trafik yang adil antar skenario.

Pengambilan data menggunakan teknik passive sniffing melalui perangkat lunak Wireshark yang dipasang pada sisi broker. Metode ini memungkinkan peneliti menangkap setiap paket MQTT (CONNECT, PUBLISH, PUBACK, PUBREC, PUBREL, PUBCOMP) secara mendetail hingga ke tingkat byte. Setiap skenario QoS dijalankan selama 30 menit dengan interval pengiriman data setiap 5 detik, sehingga menghasilkan minimal 360 sampel paket per tingkat QoS. Prosedur pengujian dilakukan sebanyak 3 kali pengulangan untuk setiap level QoS pada waktu yang berbeda (pagi, siang, malam), dan nilai akhir setiap parameter diperoleh dari rata-rata ketiga sesi tersebut untuk menjamin konsistensi dan reliabilitas data. Waktu pengambilan data dapat dilihat lebih ringkas pada Tabel 3 sebagai berikut.

Tabel 3. Waktu Pengujian Kinerja Jaringan pada setiap *level Quality of Service* MQTT.

Skenario	Level QoS	Durasi	Interval	Jumlah Sampel
1	QoS 0	30 menit	5 detik	≥ 360 paket
2	QoS 1	30 menit	5 detik	≥ 360 paket
3	QoS 2	30 menit	5 detik	≥ 360 paket

D. Perancangan Alur Kerja Sistem Pengujian pada Penelitian



Gambar 2. Alur Kerja Sistem Pengujian pada Penelitian.

E. Metode Pengukuran dan Analisis Data

Setelah seluruh data dari ketiga skenario QoS terkumpul, dilakukan proses analisis untuk memperoleh nilai parameter kinerja jaringan menggunakan metode perhitungan yang telah ditetapkan. Setiap parameter diukur berdasarkan data hasil capture Wireshark yang diekspor ke format CSV untuk diolah menggunakan Microsoft Excel.

1) Pengukuran Delay

Delay merupakan waktu yang dibutuhkan paket data untuk melakukan perjalanan dari perangkat pengirim ke penerima melalui broker MQTT. Secara teoritis, delay setiap paket dihitung menggunakan persamaan 1.

$$Delay_i = Waktu_{diterima,i} - Waktu_{dikirim,i} \quad (1)$$

Untuk mendapatkan nilai rata-rata delay keseluruhan, digunakan persamaan 2.

Dalam implementasi praktis menggunakan Wireshark, pengukuran delay dilakukan dengan menghitung rata-rata nilai kolom Time dari seluruh paket yang tertangkap dengan filter 'tcp & mqtt'. Nilai Time merepresentasikan waktu relatif setiap paket sejak koneksi MQTT pertama kali dibangun, sehingga rata-rata nilai Time mencerminkan distribusi waktu pengiriman paket selama sesi pengujian berlangsung.

2) Pengukuran Jitter

Jitter adalah ketidakstabilan waktu kedatangan paket data yang menunjukkan seberapa stabil selang waktu antar paket yang diterima. Jitter dihitung menggunakan persamaan 3.

$$\text{Jitter} = \frac{\sum_{i=1}^{n-1} |\text{Delay}_i - \text{Delay}_{i+1}|}{n - 1} \quad (2)$$

Dalam implementasi menggunakan Wireshark, pengukuran jitter dilakukan dengan menghitung rata-rata nilai kolom Delta Time yang merepresentasikan selisih waktu antara paket ke-i dengan paket sebelumnya. Rata-rata Delta Time mencerminkan konsistensi interval kedatangan antar paket selama sesi pengujian.

3) Pengukuran Packet Loss

Packet loss adalah persentase paket data yang hilang selama transmisi dan tidak pernah sampai ke tujuan. Packet loss dihitung menggunakan persamaan 4 dan bisa menggunakan persamaan 5.

$$\text{Packet Loss} = \frac{(\text{Jumlah Paket Terkirim} - \text{Jumlah Paket Diterima})}{\text{Jumlah Paket Terkirim}} \times 100\% \quad (3)$$

Atau bisa menggunakan persamaan 5.

$$\text{Packet Loss} = \frac{(\text{lost segment})}{\text{total tcp}} \times 100\% \quad (5)$$

Pengukuran packet loss dilakukan menggunakan filter 'tcp' dan 'tcp.analysis.lost_segment' pada Wireshark untuk mendeteksi segmen TCP yang hilang, kemudian dihitung persentasenya terhadap total paket yang tertangkap selama sesi pengujian.

4) Pengukuran Throughput

Throughput adalah tingkat transfer data efektif yang berhasil dikirimkan melalui jaringan dalam satuan waktu tertentu. Throughput dihitung menggunakan persamaan 6.

$$\text{Throughput} = \frac{\text{Total Byte Data Diterima}}{\text{Total Waktu Pengujian (detik)}} \quad (6)$$

Pengukuran throughput dilakukan dengan membagi total byte data payload yang berhasil diterima dengan durasi aktual pengamatan (time span) dari hasil capture Wireshark, kemudian dikonversi ke satuan Kilobit per second (Kbps).

5) Pengukuran Bandwidth Consumption

Bandwidth consumption adalah jumlah bandwidth yang digunakan untuk mengirimkan data dalam satuan waktu tertentu. Bandwidth consumption dihitung menggunakan persamaan:

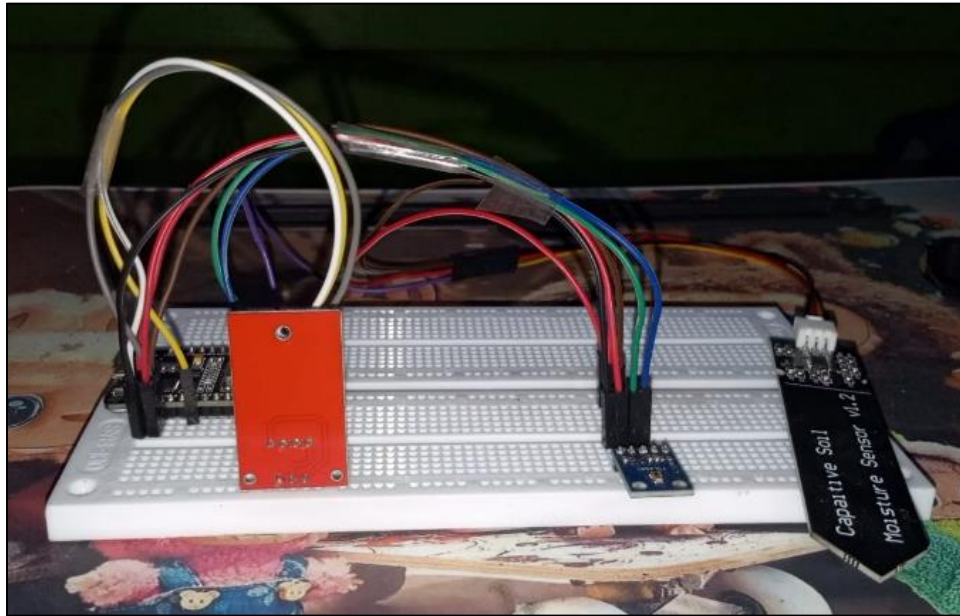
$$\text{Bandwidth Consumption} = \frac{\sum \text{Ukuran Paket (bit)}}{\text{Total Waktu Pengujian (detik)}} \quad (7)$$

Pengukuran bandwidth consumption dilakukan dengan membagi total byte seluruh paket (termasuk header dan paket kontrol) yang tertangkap Wireshark dengan durasi pengamatan, kemudian dikonversi ke satuan Kilobit per second (Kbps). Parameter ini berbeda dengan throughput karena mencakup overhead protokol, bukan hanya payload data.

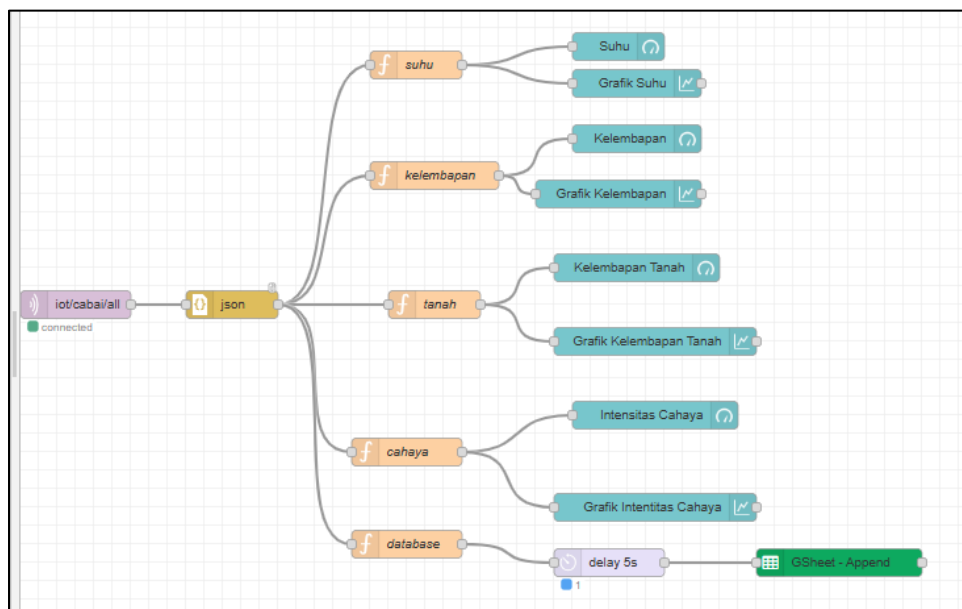
3. HASIL DAN PEMBAHASAN

A. Hasil Rangkaian Alat dan Visualisasi Data pada Pengujian

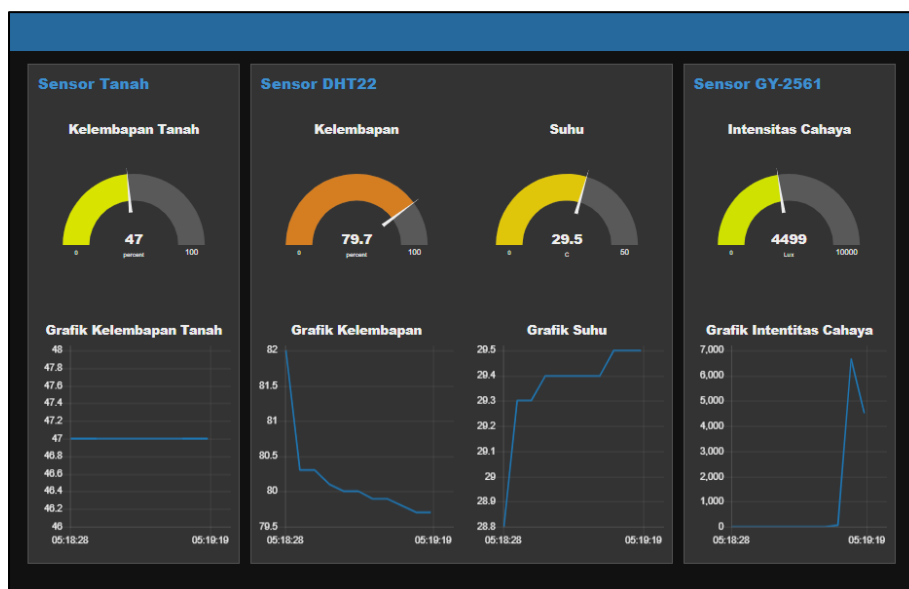
Hasil rangkaian alat yang dibuat dan rangkaian node-red serta tampilan *dashboard* dapat di lihat pada Gambar 3, 4 dan 5.



Gambar 3. Hasil rangkaian alat untuk pengujian.



Gambar 4. Hasil Rangkaian untuk Tampilan di Node-Red.



Gambar 5. Hasil Dashboard pada Node-Red.

B. Hasil Pengujian

1) Hasil Delay Jaringan

Parameter delay diukur sebagai waktu rata-rata yang dibutuhkan paket data dari publisher (ESP32) untuk sampai ke subscriber melalui broker Mosquitto. Pengukuran dilakukan pada ketiga level QoS dengan hasil yang ditampilkan pada Tabel 4.

Tabel 4. Hasil Pengujian Delay setiap Level QoS.

Sesi Pengujian	QoS	Delay (ms)
Pagi	0	2.612,52
Siang	0	2.607,97
Malam	0	2.608,60
Rata-rata	0	2.609,70
Pagi	1	2.488,56
Siang	1	2.488,56
Malam	1	2.488,41
Rata-rata	1	2.488,51
Pagi	2	1.246,83
Siang	2	1.246,55
Malam	2	1.246,89
Rata-rata	2	1.246,76

Hasil pengukuran delay menunjukkan perbedaan yang signifikan antar level QoS. Pada QoS 0, nilai delay berkisar antara 2.607,97 ms hingga 2.612,52 ms dengan rata-rata sebesar 2.609,70 ms. Pada QoS 1, delay berkisar antara 2.488,41 ms hingga 2.488,56 ms dengan rata-rata 2.488,51 ms. Sedangkan QoS 2 menunjukkan delay paling rendah, berkisar antara 1.246,55 ms hingga 1.246,89 ms dengan rata-rata 1.246,76 ms.

Temuan ini menunjukkan bahwa QoS 2 memiliki nilai delay rata-rata paling rendah (1.246,76 ms), sekitar setengah dari delay QoS 0 (2.609,70 ms) dan QoS 1 (2.488,51 ms). Hal ini

disebabkan oleh lebih banyaknya jumlah paket yang tertangkap pada QoS 2 akibat overhead handshake empat tahap (PUBLISH, PUBREC, PUBREL, PUBCOMP), sehingga distribusi waktu paket lebih padat di bagian awal sesi capture dan menurunkan nilai rata-rata Time secara keseluruhan. QoS 0 memiliki delay tertinggi karena jumlah paket yang lebih sedikit (tanpa paket konfirmasi) membuat distribusi waktu paket lebih menyebar sepanjang durasi capture, sehingga rata-rata Time-nya lebih besar. QoS 1 berada di antara keduanya dengan satu paket PUBACK per pesan yang menambah kerapatan distribusi. Nilai delay yang diukur merupakan rata-rata waktu relatif paket sejak koneksi MQTT pertama dibangun, sehingga tidak merepresentasikan round-trip time sesungguhnya antar perangkat. Oleh karena itu, analisis delay pada penelitian ini difokuskan pada perbandingan antar level QoS, di mana QoS 2 secara konsisten menghasilkan nilai terendah dibanding QoS 0 dan QoS 1 di seluruh sesi pengujian.

2) Hasil Pengujian Jitter

Pada bagian ini hasil jitter untuk menunjukkan variasi ketidakstabilan kedatangan antar paket data. Nilai jitter yang rendah mengindikasikan konsistensi pengiriman data yang baik. Hasil pengukuran jitter ditampilkan pada Tabel 5.

Tabel 5. Hasil Pengujian Jitter setiap Level QoS.

Sesi Pengujian	QoS	Jitter (ms)
Pagi	0	172,77
Siang	0	202,39
Malam	0	194,97
Rata-rata	0	190,04
Pagi	1	305,16
Siang	1	250,00
Malam	1	274,33
Rata-rata	1	276,50
Pagi	2	136,29
Siang	2	128,27
Malam	2	125,65
Rata-rata	2	130,07

Hasil pengukuran jitter menunjukkan variasi yang cukup signifikan antar level QoS. QoS 1 memiliki nilai jitter tertinggi dengan rata-rata 276,50 ms, diikuti QoS 0 dengan rata-rata 190,04 ms, dan QoS 2 dengan rata-rata terendah 130,07 ms. Pada QoS 0, nilai jitter berkisar antara 172,77 ms (pagi) hingga 202,39 ms (siang). Pada QoS 1, jitter berkisar antara 250,00 ms (siang) hingga 305,16 ms (pagi). Sedangkan QoS 2 menunjukkan jitter paling konsisten, berkisar antara 125,65 ms (malam) hingga 136,29 ms (pagi).

QoS 2 memiliki nilai jitter terendah (rata-rata 130,07 ms) karena mekanisme handshake empat tahap menghasilkan paket-paket konfirmasi yang dikirim dalam interval sangat singkat secara berurutan, sehingga rata-rata selisih waktu antar paket menjadi lebih kecil. QoS 1 memiliki jitter tertinggi (rata-rata 276,50 ms) karena mekanisme at-least-once dapat memicu pengiriman ulang paket secara tidak teratur apabila PUBACK tidak diterima tepat waktu, menghasilkan variasi interval Delta Time yang lebih besar. QoS 0 berada di antara keduanya (190,04 ms) karena meskipun tidak ada overhead konfirmasi, interval pengiriman sensor yang tetap setiap 5 detik menjadi satu-satunya penentu jarak antar paket tanpa ada paket tambahan yang memperkecil rata-rata Delta Time. Nilai jitter yang tinggi pada seluruh level QoS dipengaruhi oleh metode pengukuran Delta Time Wireshark yang mencakup semua paket

termasuk paket kontrol handshake. Dalam konteks monitoring tanaman dengan interval data 5 detik, variasi ini tidak berdampak signifikan terhadap konsistensi data di dashboard, sehingga analisis jitter lebih relevan dilihat dari perbandingan antar QoS daripada terhadap standar telekomunikasi umum.

3) Hasil Pengujian Packet Loss

Packet loss merupakan persentase paket data yang gagal sampai ke tujuan selama proses transmisi berlangsung. Nilai packet loss yang rendah mengindikasikan keandalan jaringan yang baik. Standar TIPHON menetapkan bahwa packet loss di bawah 1% dikategorikan sebagai kondisi jaringan yang sangat baik. Hasil pengukuran ditampilkan pada Tabel 6.

Tabel 6. Hasil Pengujian Packet Loss setiap *Level* QoS.

Sesi	QoS	Lost Segment	Total Packet	Packet Loss (%)
Pagi	0	47	13.135	0,358
Siang	0	25	12.152	0,206
Malam	0	29	12.113	0,239
Pagi	1	70	12.830	0,546
Siang	1	55	13.549	0,406
Malam	1	15	11.649	0,129
Pagi	2	9	11.804	0,076
Siang	2	10	13.506	0,074
Malam	2	39	13.175	0,296

Hasil pengukuran packet loss menunjukkan bahwa seluruh skenario pengujian berada di bawah ambang batas 1%, yang mengindikasikan kualitas jaringan yang sangat baik menurut standar TIPHON. Pada QoS 0, nilai packet loss berkisar antara 0,206% hingga 0,358% dengan nilai tertinggi pada sesi pagi (0,358%) dan terendah pada sesi siang (0,206%), dengan rata-rata 0,268%. Pada QoS 1, packet loss berkisar antara 0,129% hingga 0,546% dengan nilai tertinggi pada sesi pagi (0,546%) dan terendah pada sesi malam (0,129%), dengan rata-rata 0,360%. Sementara QoS 2 mencatat packet loss paling rendah secara keseluruhan, berkisar antara 0,074% hingga 0,296%, dengan nilai terendah pada sesi siang (0,074%) dan rata-rata 0,149%. Berdasarkan standar TIPHON, seluruh nilai packet loss pada ketiga level QoS berada dalam kategori 'Sangat Bagus' (<3%). Nilai packet loss QoS 0 yang lebih rendah dari QoS 1 dapat dijelaskan karena mekanisme pengiriman ulang pada QoS 1 justru menambah beban trafik yang berpotensi meningkatkan hilangnya segmen TCP pada kondisi tertentu. Ketiga level QoS dinilai memenuhi syarat keandalan minimal untuk aplikasi monitoring pertanian berskala kecil.

4) Hasil Pengujian *Throughput*

Throughput dihitung berdasarkan total *bytes* seluruh paket yang tertangkap selama durasi capture dibagi dengan durasi aktual pengamatan. Nilai *throughput* menunjukkan tingkat transfer data efektif yang berhasil dikirimkan melalui jaringan. Hasil pengukuran ditampilkan pada Tabel 7.

Tabel 7. Hasil Pengujian *Throughput* setiap *Level* QoS.

Sesi	QoS	Total Byte	Time Span (s)	Throughput (Kbps)
Pagi	0	50.210	1.779,998	0,23

Sesi	QoS	Total Byte	Time Span (s)	Throughput (Kbps)
Siang	0	50.619	1.790,002	0,23
Malam	0	50.337	1.779,999	0,23
Pagi	1	51.194	1.784,998	0,23
Siang	1	51.193	1.785,000	0,23
Malam	1	51.194	1.784,998	0,23
Pagi	2	51.194	1.785,000	0,23
Siang	2	51.231	1.790,001	0,23
Malam	2	51.194	1.785,000	0,23

Hasil pengukuran throughput menunjukkan nilai yang seragam pada seluruh skenario pengujian, yaitu sebesar 0,23 Kbps untuk semua level QoS 0, 1, dan 2 di ketiga sesi pengujian. Hal ini mengindikasikan bahwa jumlah data sensor (payload) yang berhasil dikirimkan relatif konstan dan tidak dipengaruhi secara signifikan oleh perbedaan level QoS maupun waktu pengujian.

Konsistensi throughput di semua level QoS menunjukkan bahwa payload data aktual yang dikirimkan oleh sensor adalah sama, dan perbedaan QoS hanya mempengaruhi mekanisme pengiriman dan konfirmasi, bukan jumlah data payload itu sendiri. Hal ini sesuai dengan desain penelitian di mana interval pengiriman data dan ukuran payload dibuat tetap untuk semua skenario. Nilai throughput 0,23 Kbps juga menunjukkan bahwa sistem IoT monitoring tanaman ini menggunakan bandwidth yang sangat kecil, sesuai dengan karakteristik aplikasi monitoring yang mengirimkan data sensor dalam ukuran kecil secara berkala. Berdasarkan standar TIPHON, nilai throughput 0,23 Kbps (setara 230 bps) masuk kategori 'Sangat Bagus' (≥ 100 bps). Konsistensi throughput di seluruh sesi juga memvalidasi desain eksperimen, membuktikan bahwa perbedaan hasil antar skenario murni berasal dari perbedaan mekanisme QoS, bukan dari faktor payload maupun kondisi jaringan.

5) Hasil Pengujian Bandwidth Consumption

Bandwidth consumption mengukur total penggunaan bandwidth termasuk overhead protokol (header, paket kontrol, acknowledgment). Parameter ini berbeda dengan throughput karena mencakup semua paket yang dikirimkan, bukan hanya payload data. Hasil pengukuran ditampilkan pada Tabel 8.

Tabel 8. Hasil pengujian Bandwidth Consumption setiap Level QoS.

Sesi	QoS	Total Byte	Time Span (s)	Bandwidth (Kbps)
Pagi	0	68.501	1.784,354	0,31
Siang	0	69.012	1.794,286	0,31
Malam	0	68.618	1.784,282	0,31
Rata-rata	0	-	-	0,31
Pagi	1	72.095	1.789,275	0,32
Siang	1	72.094	1.789,277	0,32
Malam	1	72.095	1.789,168	0,32
Rata-rata	1	-	-	0,32
Pagi	2	113.623	1.789,201	0,51
Siang	2	113.834	1.793,787	0,51
Malam	2	113.623	1.789,289	0,51

Sesi	QoS	Total Byte	Time Span (s)	Bandwidth (Kbps)
Rata-rata	2	-	-	0,51

Hasil pengukuran bandwidth consumption menunjukkan adanya perbedaan yang signifikan antar level QoS. QoS 0 mengonsumsi bandwidth sebesar 0,31 Kbps, QoS 1 sebesar 0,32 Kbps, dan QoS 2 sebesar 0,51 Kbps. Nilai bandwidth pada masing-masing level QoS konsisten di seluruh sesi pengujian, yang menunjukkan bahwa pola overhead protokol stabil dan dapat diprediksi.

QoS 2 mengonsumsi bandwidth hampir dua kali lipat dibandingkan QoS 0 (peningkatan sekitar 64,5%). Perbedaan ini disebabkan oleh overhead paket kontrol yang jauh lebih banyak pada QoS 2, di mana setiap pengiriman pesan memerlukan empat paket tambahan (PUBREC, PUBREL, PUBCOMP, dan ACK), sedangkan QoS 0 tidak memerlukan paket konfirmasi sama sekali. QoS 1 berada di antara keduanya dengan overhead satu paket PUBACK per pesan, menghasilkan bandwidth consumption sebesar 0,32 Kbps (peningkatan sekitar 3,2% dari QoS 0).

Perbandingan antara bandwidth consumption dan throughput sangat signifikan, throughput yang sama (0,23 Kbps) pada semua QoS membuktikan bahwa perbedaan bandwidth semata-mata berasal dari overhead protokol, bukan dari perbedaan payload data. Selisih antara bandwidth consumption dan throughput merepresentasikan overhead murni: QoS 0 memiliki overhead sekitar 0,08 Kbps (34,8% dari throughput), QoS 1 sekitar 0,09 Kbps (39,1%), dan QoS 2 sekitar 0,28 Kbps (121,7%). Hal ini menunjukkan bahwa mekanisme *exactly-once* pada QoS 2 memerlukan bandwidth tambahan yang lebih dari dua kali lipat payload data aktual untuk menjamin keandalan pengiriman. Dalam pemilihan konfigurasi QoS secara praktis, overhead QoS 2 yang mencapai 121,7% perlu dipertimbangkan ketika sistem dioperasikan pada jaringan dengan banyak perangkat IoT secara bersamaan karena berpotensi menyebabkan kongesti. QoS 0 dengan overhead 34,8% lebih direkomendasikan pada jaringan terbatas, sedangkan QoS 2 lebih sesuai pada jaringan stabil yang mengutamakan keandalan data.

4. KESIMPULAN

Berdasarkan hasil pengujian dan analisis yang dilakukan terhadap kinerja jaringan protokol MQTT pada sistem IoT *monitoring* tanaman cabai rawit dengan variasi tingkat *Quality of Service* (QoS), dapat ditarik kesimpulan bahwa variasi tingkat QoS pada protokol MQTT berpengaruh signifikan terhadap kinerja jaringan sistem IoT *monitoring* tanaman cabai rawit. QoS 2 menghasilkan *delay* terendah (1.246,76 ms) dan jitter paling stabil (130,07 ms), namun mengonsumsi *bandwidth* tertinggi (0,51 Kbps dengan overhead 121,7%). QoS 0 memiliki *delay* tertinggi (2.609,70 ms) namun *bandwidth consumption* paling efisien (0,31 Kbps dengan overhead 34,8%). QoS 1 berada di tengah dengan delay 2.488,51 ms dan bandwidth 0,32 Kbps (overhead 39,1%). Packet loss pada semua level QoS sangat baik (< 1%), dengan QoS 2 terendah (0,149%). Throughput konsisten di semua level (0,23 Kbps), menunjukkan bahwa perbedaan *bandwidth consumption* semata-mata berasal dari overhead protokol. Mekanisme *exactly-once* pada QoS 2 memerlukan *bandwidth* tambahan lebih dari dua kali lipat *payload* data aktual untuk menjamin keandalan pengiriman. Penelitian ini memiliki beberapa keterbatasan yaitu pengujian dilakukan pada jaringan WiFi lokal yang terkontrol sehingga belum mencerminkan kondisi jaringan pertanian di lapangan yang lebih kompleks, hanya menggunakan satu node publisher sehingga belum mengevaluasi skalabilitas sistem dengan banyak perangkat, dan pengukuran delay menggunakan pendekatan rata-rata waktu relatif Wireshark, bukan pengukuran round-trip time langsung yang lebih

presisi. Untuk penelitian selanjutnya, disarankan untuk menguji kinerja MQTT pada kondisi jaringan tidak stabil dengan variasi kekuatan sinyal WiFi guna mengevaluasi ketahanan setiap level QoS terhadap gangguan jaringan di lingkungan pertanian sesungguhnya, membandingkan performa MQTT dengan protokol CoAP atau AMQP pada skenario multi-node (>10 publisher), dan mengintegrasikan analisis konsumsi daya menggunakan sensor INA219 untuk mengevaluasi efisiensi energi di setiap level QoS.

REFERENSI

- [1] Suswanto, D., et al. (2025). Internet of Things: Konsep dan Implementasi. *Jurnal Teknologi Informasi*, 12(1), 1-15.
- [2] Budiani, R. E., Irawan, J. D., & Rudhistiar, D. (2024). Sistem Monitoring dan Penyiraman Otomatis pada Tanaman Cabai Berbasis Internet of Things (IoT). *Jurnal Mahasiswa Teknik Informatika*, 8(2), 45-58.
- [3] Safitri, N. A., & Seto Priambodo, A. (2023). MQTT and CoAP Communication Protocol Analysis in Internet of Things System for Strawberry Hydroponic Plants. *Journal of Robotics, Automation, and Electronics Engineering*, 1(1), 88-97.
- [4] Dewantara, D. (2025). Evaluasi Kinerja MQTT Dengan Berbagai Tingkat Quality Of Service (QoS). *Jurnal Teknik Elektro dan Informatika*, 4(1), 23-35.
- [5] Hermansyah, A., Ubaya, H., Aljali, A. A., Afifah, N., & Ningrum, S. K. (2025). Quality of Service (QoS) Analysis of MQTT Protocol for Smart Farming Monitoring System Optimization. *Jurnal Teknokes*, 18(1), 1-08.
- [6] Choudhary, A. (2024). Internet of Things: a comprehensive overview, architectures, applications, simulation tools, challenges and future directions. *Discover Internet of Things*, 4(1), 1-25.
- [7] Barus, E., Kunci, K., Tanaman, P., & Cabai, T. (2023). Sistem Monitoring Pertumbuhan Tanaman Berbasis Internet of Things. *Jurnal Ilmu Komputer Dan Sistem Informasi (JIKOMSI)*, 6(1), 1-8.
- [8] Saptadi, N. T. S., Iskandar, D., et al. (2025). Integrasi Internet of Things (IoT) dan Embedded System dalam Era Digital. *Jurnal Teknologi Digital*, 15(2), 89-102.
- [9] Mahfuza, S., Ristian, U., et al. (2024). Penerapan Protokol MQTT pada Sistem Otomatisasi Fotosintesis Tanaman Aquascape Berbasis Internet of Things. *Jurnal Ilmu Komputer dan Sistem Informasi*, 12(3), 134-145.
- [10] Nababan, A. (2024). Analisis Quality Of Service Dengan Standar Tiphon Tr 101 329 V1.2.5 Pada Sistem Monitoring Perangkat Hydrant. *Jurnal Penelitian Teknik Informatika*, 9(4), 201-215.
- [11] Safira, R., & W, Y. (2023). Analisis Kinerja Jaringan Komputer Dengan Menggunakan Metode Network Performance Analysis. *Jurnal Teknik Informatika dan Komputer*, 1(1), 45-56.
- [12] Islamy, I., & Wisudawati, L. M. (2023). Sistem Monitoring Smart Garden Tanaman Cabai Berbasis IoT Menggunakan Protokol MQTT, Node Red, dan Telegram Bot. *Jurnal Teknotan*, 17(3), 197-205.
- [13] Setyawan, D., Abdulloh, A., & Trisno, B. (2018). Sistem monitoring kelembaban tanah, kelembaban udara, dan suhu pada lahan pertanian menggunakan protokol MQTT. *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, 2(11), 5776-5784.
- [14] Harianto, S., Firmansyah, R., & Wibowo, A. (2021). Perancangan dan realisasi sistem monitoring penyemprotan air pada budidaya aeroponik menggunakan protokol MQTT berbasis Internet of Things. *eProceedings of Engineering*, 8(5), 6789-6802.
- [15] Yusa, M., Hasanah, U., & Prabowo, D. (2023). Kebutuhan intensitas cahaya pada pertumbuhan optimal tanaman hortikultura di greenhouse tertutup. *Jurnal Agronomi Indonesia*, 51(1), 89-98.

-
- [16] Nurfiqin, L., Sari, Z., & Sumadi, F. D. S. (2024). Analisis Quality of Service (QoS) Protokol MQTT dan HTTP pada Sistem Smart Metering Arus Listrik. *Jurnal Repositor*, 3(1).
- [17] Akbar, R. M. R., Arif, T. Y., & Irhamsyah, M. (2023). Analisis Performansi Protokol MQTT pada Sistem Pemantauan Kualitas Udara Ruangan Berbasis IoT. *KITEKTRO: Jurnal Komputer, Informasi Teknologi, dan Elektro*, Universitas Syiah Kuala.
- [18] Hanif, A., & Amri, R. (2023). Implementasi Internet of Things pada Protokol MQTT dan HTTP dalam Sistem Pendeteksi Banjir. *Jurnal INOVTEK Polbeng - Seri Informatika*, 8(2).