BAB II
TINJAUAN PUSTAKA

2.1. Studi Literatur

Dalam melakukan penelitian ini, terdapat penelitian terdahulu untuk mendukung teori dan metode pemecahan masalah pada penelitian ini. Beberapa penelitian terdahulu yang dijadikan referensi peneliti disajikan pada Tabel 1.

Tabel 1. Studi Literatur

No	Penulis	Judul	Tahun	Hasil Penelitian
1.	Hamoud H. Alshammari	*The Internet of Things Healthcare Monitoring System Based on MQTT Protocol*	2023	Penelitian tentang sistem pemantauan kesehatan berbasis IoT dengan protokol MQTT menunjukkan bahwa sistem ini dapat memantau tanda-tanda vital pasien secara *real-time* dengan *low latency*, yang sangat bermanfaat bagi pasien di daerah terpencil. Evaluasi menunjukkan keandalan dan efisiensi MQTT, dengan RTT yang lebih singkat pada tingkat QoS MQTT 1 dan 2 dibandingkan protokol lain, menjadikannya cocok untuk transmisi data kesehatan secara *real-time*.
2.	Victor Seoane,	*Performance Evaluation of*	2021	Protokol MQTT memerlukan lebih banyak

No	Penulis	Judul	Tahun	Hasil Penelitian
	Carlos Garcia-Rubio, Florina Almenares, Celeste Campo	*CoAP And MQTT with Security Support For IoT Environments*		*bandwidth* karena *overhead* TCP, namun menawarkan berbagai tingkat QoS dan keandalan yang lebih tinggi. Sebaliknya, CoAP menawarkan QoS dan keandalan yang lebih sederhana. Pengamanan komunikasi melalui DTLS untuk CoAP atau TLS untuk MQTT juga meningkatkan penggunaan *bandwidth*.
3.	Vincent de Paul Niyigena Kwizera, Zhanming Li, Victus Elikplim Lumorvie, Febronie Nambajemariya, Xiaowei Niu	*IoT Based Greenhouse Real-Time Data Acquisition and Visualization through Message Queuing Telemetry Transfer (MQTT) Protocol*	2021	Penggunaan protokol MQTT memungkinkan transmisi data secara efisien dan *real-time* ke *cloud platform*. Pengujian fungsionalitas memastikan data berhasil dipublikasikan dan diterima, menjamin komunikasi yang andal antara sensor dan *cloud*. Sistem ini memungkinkan pemantauan pengoperasian pengontrol dalam rumah kaca dari jarak jauh, meningkatkan kepraktisan dan ketepatan manajemen pertanian serta mendukung

No	Penulis	Judul	Tahun	Hasil Penelitian
				pengambilan keputusan yang lebih analitis dalam praktik pertanian.
4.	Hoe Tung Yew; Ming Fung Ng; Soh Zhi Ping; Seng Kheau Chung; Ali Chekima; Jamal A. Dargham	*IoT Based Real-Time Remote Patient Monitoring System*	2020	Sistem ini memungkinkan dokter untuk memantau data ECG pasien secara *real-time* melalui server web atau aplikasi seluler. Kemampuan ini meningkatkan aksesibilitas data pasien, sehingga memungkinkan intervensi medis yang tepat waktu. Penerapan antarmuka pengguna grafis (GUI) memungkinkan pengguna untuk mengamati dan menganalisis data yang dikumpulkan dengan mudah. Pendekatan yang mudah digunakan ini penting bagi penyedia layanan kesehatan dan pasien.
5.	Biswajeeban Mishra; Attila Kertesz	*The Use of MQTT in M2M and IoT Systems: A Survey*	2020	Penelitian ini menyoroti evolusi berbagai protokol komunikasi M2M selama 20 tahun terakhir, menekankan bahwa MQTT telah menjadi

No	Penulis	Judul	Tahun	Hasil Penelitian
				protokol yang paling banyak digunakan dalam sistem M2M/IoT. Hal ini didukung oleh analisis literatur yang komprehensif, yang menunjukkan fitur utama, kelebihan, dan keterbatasan MQTT dibandingkan dengan protokol lain. Makalah ini mengidentifikasi berbagai bidang aplikasi untuk MQTT, yang menunjukkan fleksibilitasnya dalam berbagai domain seperti rumah pintar, perawatan kesehatan, dan otomatisasi industri. Hal ini dicapai melalui tinjauan menyeluruh terhadap berbagai studi yang relevan.
6.	Cheng-Yi Chen, Sheng-Han Wu, Bo-Wun Huang, Chao-Hung Huang, Cheng-Fu Yang	*Web-based Internet of Things on environmental and lighting control and monitoring system using node-RED, MQTT and*	2024	Penelitian ini menyajikan integrasi yang berhasil antara perangkat lunak dan perangkat keras eksperimen, dengan mengonversi data register Modbus ke dalam komunikasi MQTT untuk proses *publishing* dan

No	Penulis	Judul	Tahun	Hasil Penelitian
		Modbus communications within embedded Linux platform		*subscribing*, yang dicapai melalui program Python tertentu. Hasil eksperimen memvalidasi efektivitas biaya dalam mentransformasi sistem kendali tradisional menjadi sistem kendali supervisi berbasis web, yang meningkatkan fungsionalitas peralatan yang telah ada. Antarmuka Mesin-Manusia (*Human-Machine Interface*/HMI) untuk pengendalian pencahayaan, tirai, dan pemantauan lingkungan telah dikembangkan, menunjukkan kemampuan kendali dan pemantauan yang efektif. Sistem ini memungkinkan pengendalian pencahayaan secara jarak jauh, yang berkontribusi pada tujuan penghematan energi dan pengurangan emisi karbon.

2.2. *Internet of Things* (IoT)

Istilah *Internet of Things* (IoT) pertama kali dicetuskan oleh Kevin Ashton pada tahun 1999. Ashton, seorang pelopor dalam teknologi dan inovasi,

menciptakan istilah ini saat bekerja di *Procter & Gamble* (P&G). Dia menggambarkan bagaimana konektivitas internet dapat diterapkan pada dunia fisik melalui sensor dan perangkat yang saling terhubung. Konsep ini pada awalnya diterapkan untuk meningkatkan efisiensi rantai pasokan di P&G. Sejak saat itu, IoT telah berkembang menjadi salah satu bidang teknologi yang paling cepat berkembang, mencakup berbagai aplikasi mulai dari rumah pintar hingga kota pintar dan industri manufaktur.

Setelah mempresentasikan penggunaan RFID yang terhubung dengan internet dalam rantai pasokan di P&G, Ashton melihat masa depan di mana komputer bisa mengumpulkan data dan mengolahnya menjadi informasi yang berguna tanpa intervensi manusia [5]. Hal ini dimungkinkan melalui teknologi seperti sensor dan RFID, yang memungkinkan komputer untuk mengamati, menafsirkan, dan memahami lingkungan sekitarnya.

Dalam konteks *Internet of Things* (IoT), istilah "*things*" merujuk pada entitas fisik apa pun yang dilengkapi dengan sensor dan modul komunikasi yang memungkinkan perangkat tersebut untuk secara otomatis mengumpulkan, memproses, dan mengirimkan data melalui jaringan tanpa keterlibatan manusia secara langsung. Integrasi teknologi ini memungkinkan perangkat untuk memantau kondisi internal maupun faktor lingkungan eksternal secara berkelanjutan, sehingga mendukung sistem dalam melakukan analisis situasi dan pengambilan keputusan secara otonom dan berbasis data [6].

IoT merupakan suatu paradigma teknologi yang mengintegrasikan entitas fisik ke dalam jaringan digital, sehingga setiap objek memiliki identitas, kemampuan komunikasi, dan eksistensi virtual dalam internet. Tujuan utama dari IoT adalah menyediakan solusi inovatif melalui layanan dan aplikasi yang menghubungkan dunia nyata dengan sistem digital secara sinergis. Dalam kerangka ini, komunikasi antar mesin (*Machine-to-Machine/M2M*) berperan sebagai fondasi penting yang memungkinkan pertukaran data secara otomatis antara perangkat fisik dan aplikasi berbasis *cloud computing*.

IoT merupakan jaringan yang terdiri dari berbagai perangkat cerdas yang mampu melakukan komunikasi dan pertukaran data secara otomatis. Perangkat-

perangkat ini dapat menggunakan konektivitas kabel maupun nirkabel, dan umumnya terhubung ke jaringan global melalui tumpukan protokol internet (IP). Meskipun pendekatan berbasis IP memungkinkan keterhubungan antar sistem yang luas, implementasinya sering kali memerlukan konsumsi daya dan sumber daya memori yang relatif tinggi, yang menjadi tantangan bagi perangkat dengan spesifikasi terbatas. Sebagai alternatif, sejumlah perangkat IoT dapat memanfaatkan jaringan lokal non-IP yang lebih efisien dalam penggunaan energi. Melalui bantuan *smart gateway*, jaringan lokal tersebut tetap dapat terhubung ke internet, sehingga menjaga efisiensi operasional tanpa mengorbankan konektivitas global.

Gelombang baru perkembangan teknologi yang ditandai dengan adopsi IoT diprediksi akan menjadi fondasi utama dalam kemajuan teknologi global. Fenomena ini ditandai dengan pertumbuhan eksponensial jumlah perangkat pintar yang saling terhubung dan secara aktif memanfaatkan data dari berbagai aspek kehidupan manusia. Didukung oleh jaringan nirkabel generasi terbaru, sensor dengan presisi tinggi, serta kemampuan komputasi yang semakin canggih, aplikasi berbasis IoT menjanjikan peningkatan efisiensi, kenyamanan, dan nilai tambah yang signifikan bagi pengguna.

Perkembangan IoT telah mengubah cara manusia hidup, bekerja, bepergian, dan berbisnis. Hal ini menandai transformasi industri baru yang dikenal sebagai industri 4.0, dan menjadi kunci dalam transformasi digital masyarakat [7]. Seiring kemajuan teknologi, istilah IoT semakin diperluas dengan adanya sensor dan penggerak yang memungkinkan interaksi dengan kondisi internal maupun eksternal. Berbagai perangkat fisik seperti mobil, jam tangan pintar, AC, dan bahkan alat industri, akan mampu mengumpulkan data dan membagikannya dengan perangkat lain, memungkinkan pengukuran dan respons yang lebih akurat [8].

2.3. ESP32

ESP32 adalah sebuah *microcontroller* berbasis *chip* yang dikembangkan oleh Espressif Systems. ESP32 memiliki kemampuan Wi-Fi dan Bluetooth yang terintegrasi, serta prosesor *dual-core* Xtensa LX6 yang dapat beroperasi pada kecepatan 160 hingga 240 MHz. ESP32 juga mendukung berbagai mode daya

rendah seperti *deep sleep* untuk menghemat daya [9]. ESP32 sangat populer dalam aplikasi IoT dan otomatisasi rumah karena kemampuannya untuk menghubungkan dengan jaringan internet dan berkomunikasi dengan perangkat lain.

ESP32 memiliki sejumlah besar pin GPIO (*General-Purpose Input/Output*). Setiap pin berperan penting dalam memfasilitasi koneksi dan kontrol perangkat eksternal serta sensor, memungkinkan komunikasi yang kompleks dan keinteraktifan yang efisien. Pin GPIO pada ESP32 mendukung berbagai antarmuka komunikasi, termasuk SPI (*Serial Peripheral Interface*), I2C (*Inter-Integrated Circuit*), UART (*Universal Asynchronous Receiver-Transmitter*), dan PWM (*Pulse Width Modulation*) [9].

Desain ESP32 yang hemat daya sangat krusial dalam pengembangan aplikasi IoT yang efisien dalam konsumsi energi. *Microcontroller* ini memungkinkan perangkat untuk menjalankan operasi selama periode waktu yang lebih lama tanpa membutuhkan penggantian atau pengisian ulang daya secara sering, sehingga ideal untuk aplikasi IoT yang mengutamakan efisiensi energi.

2.4. *Message Queuing Telemetry Transport* (MQTT)

Message Queuing Telemetry Transport (MQTT) merupakan protokol komunikasi yang dikembangkan oleh IBM pada tahun 1999 dan didasarkan pada model *publish-subscribe*. Protokol ini beroperasi di atas TCP/IP dan dirancang untuk memungkinkan klien, baik yang berperan sebagai penerbit (*publisher*) maupun pelanggan (*subscriber*), untuk berkomunikasi dengan server yang bertindak sebagai perantara pesan (*message broker*). Perantara pesan ini bertugas untuk mendistribusikan pemberitahuan kepada klien yang relevan. MQTT sangat ideal untuk digunakan di lokasi terpencil yang membutuhkan efisiensi dalam penggunaan *bandwidth* yang rendah [10].

Seperti protokol *publish-subscribe* lainnya, MQTT menggunakan arsitektur berbasis topik di mana data yang dipertukarkan diorganisir menurut topik yang disusun secara hierarkis. Setiap pesan yang diterbitkan oleh klien dikaitkan dengan topik tertentu. Klien yang melakukan *publish* pesan mengirimkan informasi ke topik tersebut, sementara klien lain yang berlangganan (*subscribe*) topik tersebut akan menerima pesan yang relevan. Hal ini memungkinkan pengiriman pesan yang

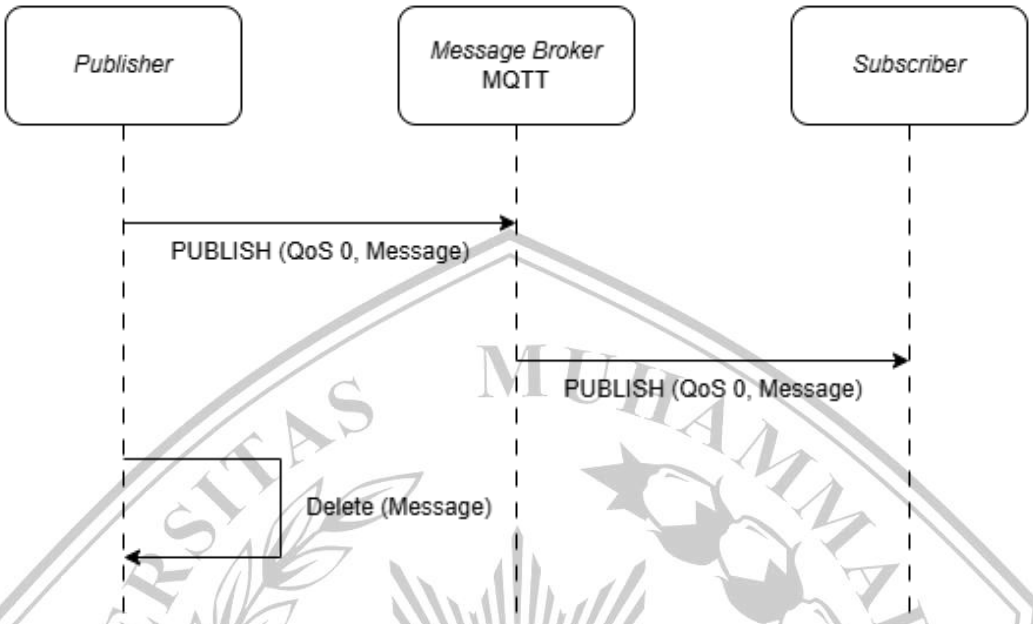
selektif, di mana pesan hanya diarahkan kepada klien yang memiliki ketertarikan atau minat pada topik yang sesuai [11]. Protokol *publish-subscribe* berbasis topik, seperti MQTT, biasanya menyajikan arsitektur fisik yang terdiri dari tiga jenis *node*: *publisher*, *subscriber*, dan *broker* [12].

- *Publisher* berperan sebagai klien yang menerbitkan pesan yang terkait dengan berbagai topik, sering kali dari sensor atau perangkat lain yang mengumpulkan data. *Publisher* bertindak sebagai produsen data dalam sistem ini. Namun, peran penerbit tidak bersifat eksklusif; klien dapat berfungsi sebagai *publisher* sekaligus *subscriber* secara bersamaan, baik untuk topik yang berbeda atau bahkan topik yang sama.
- *Subscriber* adalah klien yang mengajukan permintaan untuk berlangganan topik tertentu, dengan tujuan untuk menerima informasi yang relevan. Dalam hal ini, klien berperan sebagai konsumen data. Mirip dengan *publisher*, peran *subscriber* juga tidak bersifat eksklusif.
- *Broker* adalah entitas yang berfungsi sebagai server pusat untuk pengelolaan informasi. *Broker* bertanggung jawab untuk menerima, mengelola, dan memelihara minat topik dari *subscriber*. Selain itu, *broker* menerima pesan yang diterbitkan oleh *publisher* dan mengarahkan pesan tersebut kepada klien yang telah berlangganan topik yang bersangkutan. *Broker* bertindak sebagai filter perantara bagi klien, melakukan seleksi dan penyaringan berdasarkan topik untuk memastikan hanya informasi yang relevan yang dikirimkan kepada setiap klien.

Quality of Service (QoS) dalam MQTT merupakan mekanisme yang menetapkan tingkat keandalan dalam pengiriman pesan antara klien dan broker. MQTT memiliki tiga tingkat QoS [13], yaitu:

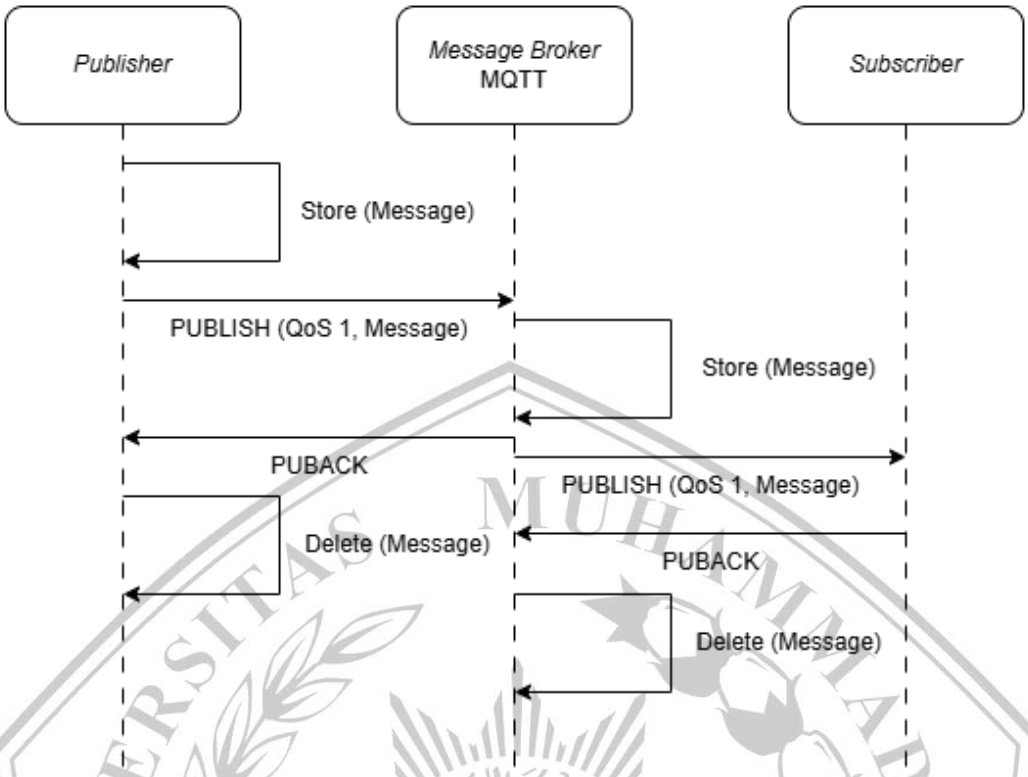
- QoS 0 (*at most once*) pada Gambar 1, pesan dikirim oleh *publisher* hanya satu kali tanpa adanya jaminan bahwa pesan akan benar-benar diterima oleh broker atau *subscriber*. MQTT tidak mengirim ulang pesan jika terjadi kegagalan, dan tidak menggunakan mekanisme *acknowledgement* (*ACK*). Artinya, jika terjadi gangguan dalam jaringan atau perangkat penerima

sedang tidak tersedia, maka pesan dapat hilang dan tidak akan pernah diterima.



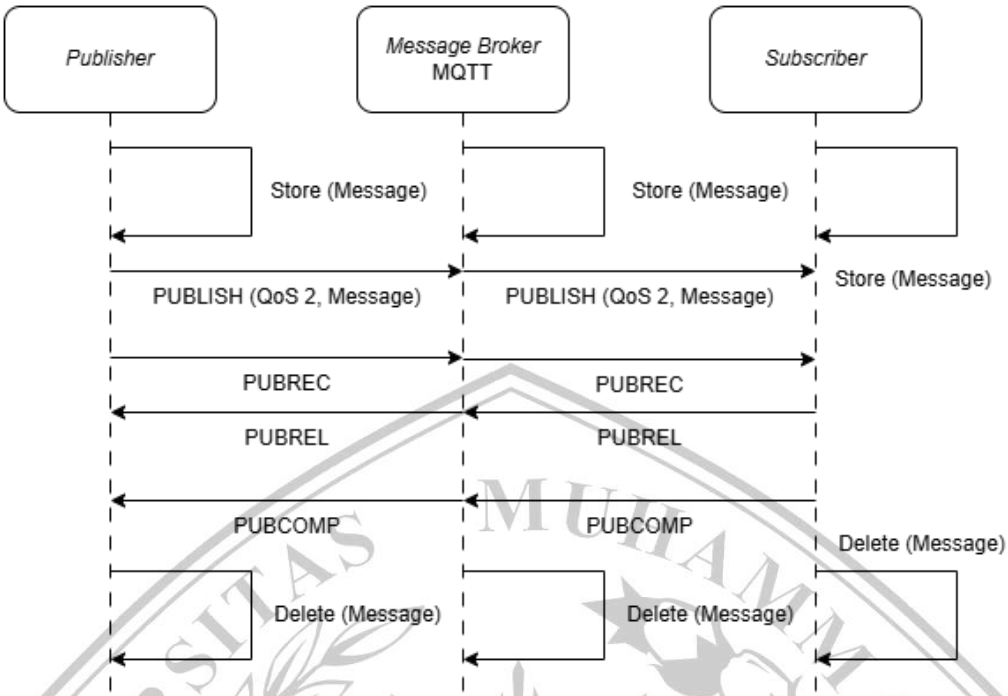
Gambar 1. Quality of Service 0 (QoS 0)

- QoS 1 (*at least once*) pada Gambar 2, menjamin bahwa pesan akan sampai ke tujuan minimal satu kali. Untuk mencapai itu, setiap pesan *PUBLISH* yang dikirim oleh *publisher* harus mendapat konfirmasi penerimaan berupa *PUBACK* dari broker. Jika konfirmasi ini tidak diterima dalam waktu tertentu, *publisher* akan mengirim ulang pesan tersebut. Akibatnya, kemungkinan pesan diterima lebih dari satu kali oleh broker maupun *subscriber* cukup besar, sehingga implementasi pada *subscriber* harus mampu menangani duplikasi pesan. QoS 1 memberikan keseimbangan antara keandalan dan performa, yang memastikan pesan dikirim minimal satu kali dengan kemungkinan duplikasi akibat pengiriman ulang jika tidak ada konfirmasi penerimaan.



Gambar 2. Quality of Service 1 (QoS 1)

- QoS 2 (*exactly once*) pada Gambar 3, adalah level paling andal dalam MQTT karena menjamin bahwa setiap pesan hanya dikirim dan diterima satu kali, tanpa adanya duplikasi. Untuk mencapai tingkat keandalan ini, MQTT menggunakan mekanisme empat langkah: *PUBLISH*, *PUBREC*, *PUBREL*, dan *PUBCOMP*. Setelah *publisher* mengirim PUBLISH, broker mengirim *PUBREC* sebagai tanda terima awal. *Publisher* kemudian mengirim *PUBREL*, lalu broker mengakhiri proses dengan PUBCOMP. Mekanisme ini menghindari duplikasi walau terjadi transmisi ulang akibat gangguan jaringan. Meskipun memberikan jaminan paling tinggi, QoS 2 juga memiliki *overhead* paling besar dan digunakan hanya ketika keandalan sangat diperlukan.



Gambar 3. Quality of Service 2 (QoS 2)

Pemilihan tingkat QoS yang sesuai bergantung pada kebutuhan aplikasi, di mana QoS yang lebih tinggi menawarkan tingkat keandalan lebih baik namun dengan peningkatan latensi serta beban komunikasi yang lebih besar [14].

2.5. Tcpdump

Tcpdump adalah alat berbasis baris perintah yang memungkinkan pengguna untuk menangkap dan memantau lalu lintas jaringan secara *real-time* [15]. Alat ini menyajikan data dalam format teks mentah, menjadikannya pilihan efisien untuk pengumpulan data cepat. Hasil dari Tcpdump biasanya disimpan dalam format ekstensi *.pcap*, yang kemudian dapat dianalisis lebih lanjut menggunakan alat lain seperti Wireshark.

Tcpdump bekerja dengan menggunakan *libpcap*, sebuah pustaka yang memungkinkan pengambilan paket dari berbagai antarmuka jaringan [16]. Dengan Tcpdump, pengguna dapat melihat detail paket yang dikirim dan diterima dalam jaringan, termasuk alamat IP sumber dan tujuan, protokol yang digunakan, serta isi paket. Tcpdump memiliki berbagai fungsi yang sangat berguna dalam analisis jaringan, di antaranya:

- Menangkap paket jaringan. Memungkinkan pengguna untuk melihat lalu lintas jaringan secara langsung.
- Menganalisis protokol. Dapat digunakan untuk memeriksa paket dari berbagai protokol seperti TCP, UDP, ICMP, HTTP, dan MQTT.
- Memantau keamanan jaringan. Membantu dalam mendeteksi aktivitas mencurigakan atau serangan siber seperti *port scanning* dan *sniffing*.
- Menguji koneksi dan latensi. Berguna dalam mengukur latensi jaringan dan mengidentifikasi masalah koneksi.
- Menyimpan data untuk Analisis lebih lanjut. Paket yang ditangkap dapat disimpan dalam format *.pcap* untuk dianalisis menggunakan alat lain seperti Wireshark.

2.6. Wireshark

Wireshark adalah perangkat lunak *open-source* yang berfungsi sebagai *network packet analyzer*. Alat ini memungkinkan pengguna untuk menangkap dan menganalisis lalu lintas jaringan secara mendalam, memberikan wawasan tentang bagaimana data dikirim dan diterima dalam suatu jaringan [17]. Wireshark memungkinkan pengguna untuk memvisualisasikan, memfilter, dan memahami pola komunikasi jaringan dengan lebih efisien melalui antarmuka yang intuitif [18]. Alat ini juga mendukung rekonstruksi sesi jaringan seperti HTTP atau VoIP, sehingga sangat berguna untuk investigasi lanjutan [19].

Wireshark sering digunakan oleh administrator jaringan, profesional keamanan, dan pengembang perangkat lunak untuk memantau, menganalisis, dan memecahkan masalah jaringan. Dengan antarmuka grafis yang intuitif, Wireshark mempermudah pengguna dalam memahami struktur paket dan protokol yang digunakan dalam komunikasi jaringan. Wireshark memiliki berbagai fungsi utama yang menjadikannya alat yang sangat berguna dalam analisis jaringan:

- Menangkap paket jaringan, Memungkinkan pengguna untuk melihat lalu lintas jaringan secara langsung.
- Menganalisis protokol, Dapat digunakan untuk memeriksa paket dari berbagai protokol seperti TCP, UDP, ICMP, HTTP, dan MQTT.

- Memantau keamanan jaringan. Membantu dalam mendeteksi aktivitas mencurigakan atau serangan siber seperti *port scanning* dan *sniffing*.
- Menguji koneksi dan latensi. Berguna dalam mengukur latensi jaringan dan mengidentifikasi masalah koneksi.
- Menyimpan data untuk analisis lebih lanjut. Paket yang ditangkap dapat disimpan dalam format *.pcap* untuk dianalisis lebih lanjut.
- Visualisasi data. Menampilkan paket dalam format yang lebih mudah dibaca dibandingkan alat berbasis *command-line* seperti Tcpdump.

Penggunaan kombinasi antara Tcpdump dan Wireshark sering diimplementasikan, dengan Tcpdump digunakan untuk menangkap data jaringan di server, sementara Wireshark dipakai untuk analisis data yang lebih detail di komputer lain.