

Implementasi *Internet Of Things* Pada Protokol MQTT Dan HTTP Dalam Sistem Pendeteksi Banjir

Al Hanif¹, Rahyul Amri²

^{1,2}Universitas Riau, Simpang Baru, Kec. Tampan, Kota Pekanbaru, Riau, Indonesia

Email: *al.hanif1401@student.unri.ac.id, Rahyulamri@lecturer.unri.ac.id*²

Abstract – Current information technology is rapidly advancing, with users utilizing the internet for daily needs, such as the Internet of Things (IoT). IoT plays a crucial role in early flood detection, functioning like smart alarms where connected sensors in flood-prone areas provide rapid warnings as water levels rise. In terms of natural disaster mitigation, IoT acts as an intelligent assistant, aiding in planning measures to reduce risks and losses. With real-time data provided by IoT sensors, authorities can respond more promptly, guide evacuations, and assist communities in facing disasters more effectively. However, challenges arise due to limited response time and early detection processes. Responsive and efficient systems are key to addressing these challenges. In the effort to enhance responsiveness to flood threats, the use of optimal network protocols, namely Message Queue Telemetry Transport (MQTT) and Hyper Text Transport Protocol (HTTP), is crucial in IoT systems. Therefore, a comparison of these network protocols becomes necessary. Results indicate that the MQTT protocol is more suitable for responsive and efficient flood notification data transmission due to its low delay value of 185.749403 ms, while the highest delay is observed in the HTTP protocol at 361.800781 ms. This research provides insights into the superior implementation of MQTT protocols in early flood detection systems.

Keywords – *HTTP, MQTT, QoS, IoT, Sistem Pendeteksi Banjir*

Intisari - Teknologi informasi saat ini sangat berkembang pesat, pengguna memanfaatkan internet untuk kebutuhan sehari-hari seperti Internet of Things . IoT sangat penting dalam deteksi dini banjir karena seperti alarm pintar, sensor-sensor terhubung di area rawan banjir memberikan peringatan cepat saat air mulai naik. Dalam hal mitigasi bencana alam, IoT berfungsi seperti asisten cerdas yang membantu merencanakan langkah-langkah untuk mengurangi risiko dan kerugian. Dengan data real-time yang diberikan oleh sensor IoT, pihak berwenang dapat merespons lebih cepat, memandu evakuasi, dan membantu masyarakat menghadapi bencana dengan lebih baik. Namun, permasalahan yang muncul adalah terbatasnya waktu respons dan proses pendeteksian dini. Sistem yang responsif dan efisien menjadi kunci dalam menghadapi tantangan ini. Dalam upaya meningkatkan respons terhadap ancaman banjir, penggunaan protokol jaringan terbaik antara Message Queue Telemetry Transport (MQTT) dan Hyper Text Transport Protocol (HTTP) sangat penting Dalam sistemn IoT. Oleh sebab itu, perlunya perbandingan protokol jaringan keduanya. Hasilnya menunjukkan bahwa protokol MQTT lebih cocok untuk responsive dan efisien mengirim data pemeritahuan banjir karena nilai Delay yang rendah sebesar 185,749403 ms, Sedangkan delay tertinggi dari protokol HTTP sebesar 361,800781 ms. Penelitian ini memberikan wawasan mengenai penerapan protokol MQTT lebih baik dalam Sistem deteksi dini banjir.

Kata Kunci - *HTTP, MQTT, QoS, IoT, Sistem Pendeteksi Banjir*

I. PENDAHULUAN

Pada era Teknologi 4.0, di mana konektivitas global semakin erat terjalin melalui internet, Indonesia memegang peranan penting sebagai salah satu negara yang mengalami perkembangan pesat dalam sektor teknologi. Data survei Asosiasi Penyelenggara Jasa Internet Indonesia (APJII) tahun 2018 mencatat bahwa sekitar 64,8% dari total penduduk Indonesia, atau sekitar 171 juta jiwa, telah terhubung dengan internet[1]. Fenomena ini bukan hanya

mencerminkan adopsi teknologi yang cepat, tetapi juga membuka peluang yang besar bagi implementasi teknologi inovatif seperti Internet of Things (IoT), sebuah konsep yang memungkinkan perangkat terhubung dan bertukar informasi melalui jaringan internet[2].

Pentingnya IoT terlihat jelas dalam sektor pengelolaan bencana, khususnya dalam konteks banjir. Sistem deteksi banjir yang menggunakan sensor ultrasonik dan mikrokontroler NodeMCu ESP8266, untuk memberikan kontribusi dalam pemahaman dan penanganan dini terhadap risiko banjir, dengan fokus pada integrasi respons pemadaman listrik oleh pihak PLN meningkatkan keamanan dan efisiensi energi [3]. Meskipun melalui penyebaran sensor pintar yang luas, IoT memberikan kemampuan untuk pemantauan real-time terhadap parameter lingkungan kritis seperti ketinggian air, namun, muncul permasalahan terkait dengan waktu respons yang terbatas dalam proses pendeteksian dini. Oleh karena itu, efisiensi dan responsivitas sistem menjadi kunci utama untuk mengatasi tantangan ini, dengan penekanan khusus pada pemilihan protokol jaringan yang sesuai dan efektif.

Dalam ranah protokol jaringan, IoT menggunakan beberapa metode pertukaran informasi, di antaranya adalah Message Queue Telemetry Transport (MQTT) dan Hyper Text Transport Protocol (HTTP). Protokol MQTT yang mengusung konsep publish dan subscribe, beroperasi dalam model klien-server di mana server pusat, atau broker, berfungsi sebagai penerima pesan dari klien yang dapat berupa publish topik atau subscribe, memungkinkan komunikasi many-to-many (banyak-ke-banyak). Sementara itu, protokol HTTP pengiriman pesan berbasis web yang mendukung request/response, menggunakan TCP sebagai protokol transport default dan TLS/SSL untuk keamanan.

Berdasarkan permasalahan yang telah dijelaskan diatas, penelitian ini bertujuan untuk membandingkan Quality of Service (QoS) antara protokol komunikasi MQTT dan HTTP, dengan fokus pada peningkatan waktu respons dalam Sistem pendeteksi dini banjir. Melalui evaluasi ini, penelitian berusaha untuk memberikan pemahaman yang lebih mendalam mengenai bagaimana kedua protokol tersebut berkinerja dalam mengirimkan data deteksi banjir, dengan harapan dapat memberikan kontribusi terhadap pengembangan sistem yang lebih responsif dan efisien dalam mengatasi ancaman banjir.

II. SIGNIFIKANSI STUDI

A. Tinjauan Pustaka

Penelitian pertama telah dilakukan oleh Chrisyantar Hasiholan , Rakhmadhany Primananda , Kasyful Amron pada 2018 yang berjudul “Implementasi Konsep Internet of Things pada Sistem Monitoring Banjir menggunakan Protokol MQTT” Permasalahan yang diangkat pada penelitian ini adalah Bagaimana kinerja protokol MQTT pada sistem Monitoring banjir. Publisher yang digunakan adalah Raspberry Pi 3, sensor ultrasonik HC-SR04. Broker yang digunakan adalah Mosquitto 1.4.7 dan laman web yang berperan sebagai subscriber. Hasil pengujian didapatkan kesimpulan Tingkat akurasi yang di dapat dari pengujian sistem adalah 97,801% dengan standar deviasi yang diperoleh sebesar ± 0.0309 cm dan pada pengujian integritas data, protokol MQTT memperoleh kesamaan data sebesar 100% di setiap percobaan dengan interval 10ms, 100ms dan 1000ms.[2].

Penelitian kedua telah dilakukan oleh Susetyo Bagas Bhaskoro dkk pada tahun 2022 yang berjudul “Perbandingan Performansi Latency Protokol Komunikasi HTTP Dan MQTT Pada Internet Of Things”. Permasalahan yang diangkat penelitian ini adalah perbandingan performansi latency protokol komunikasi HTTP dan MQTT. Hasil pengujian didapatkan protokol MQTT lebih unggul dibandingkan dengan HTTP. Nilai selisih latency keduanya sebanyak tiga kali pengujian sebagai berikut 46,6ms, 52,3ms, dan 50,2ms yang membuktikan bahwa protokol MQTT lebih unggul. [4].

Penelitian ketiga telah dilakukan oleh Fitra Ilham dkk pada tahun 2019 yang berjudul “Analisis performansi QoS MQTT pada sistem monitoring sungai” Permasalahan yang

diangkat adalah Bagaimana Performasi MQTT pada Sistem monitoring sungai. Hasil Pengujian pengujian *QoS* pada satu kondisi, hasilnya waktu *Delay* rata-rata pengiriman data masing-masing *QoS* MQTT adalah *QoS0* 0.943921s, *QoS1* 0.98576s, *QoS2* 1.11717s. Sedangkan *Packet Loss* rata-rata pengiriman data masing-masing *QoS* adalah *QoS0* 0.194530%, *QoS1* 0.24329%, *QoS2* 0.23604%. [5]

Penelitian keempat telah dilakukan oleh Shania Putri Windiastik, Elsha Novia Ardhana, dan Joko Triono pada tahun 2019 berjudul “Perancangan Sistem Pendeteksi Banjir Berbasis Iot (Internet Of Thing)”. Permasalahan yang diangkat pada penelitian ini adalah Bagaimana upaya untuk meminimalisasi jatuhnya korban jiwa dan kerugian yang terjadi akibat bencana banjir. Sistem deteksi banjir ini menggunakan water level sensor untuk mendeteksi ketinggian air, NodeMCU ESP8266 untuk membaca data dan mengirimkannya melalui internet dan buzzer yang berfungsi untuk mengeluarkan suara ketika sensor mendeteksi air dan berpotensi banjir. Hasil telah dibuat web Sistem Deteksi Banjir pada smartphone yang tersambung dengan WiFi. Sistem ini nantinya akan diimplementasikan kepada masyarakat di berbagai titik daerah rawan banjir.[6].

Penelitian kelima telah dilakukan oleh Ketty Siti Salamah, dan Samsul Anwar pada tahun 2021 berjudul “Rancang Bangun Sistem Pendeteksi Banjir Otomatis Berbasis Internet of Things”. Permasalahan yang diangkat adalah Keterlambatan pemadaman listrik wilayah yang terdampak banjir agar tidak menimbulkan korban jiwa akibat aliran listrik yang dimana proses pendeteksi banjir masih dilakukan secara manual. Pada penelitian ini menggunakan Wemos, HC-SR04 dan blynk merupakan suatu teknologi yang dapat dimanfaatkan untuk mendeteksi, mengukur ketinggian level banjir dan memberikan informasi yang cepat terkait keberadaan dan ketinggian level banjir. Hasil Penelitian ini adalah Rancang Bangun Miniatur Sistem Pendeteksi Banjir berbasis Internet Of Things yang berfungsi untuk mendeteksi, mengukur ketinggian level banjir dan memberikan informasi yang cepat terkait keberadaan dan ketinggian level banjir menggunakan aplikasi Blynk. [3].

B. Landasan Teori

1. Monitoring

Monitoring merupakan sebuah kegiatan untuk menjamin akan tercapainya semua tujuan organisasi dan manajemen. Monitoring juga didefinisikan sebagai langkah untuk mengkaji apakah kegiatan yang dilaksanakan telah sesuai dengan rencana, mengidentifikasi masalah yang timbul agar langsung dapat diatasi, melakukan penilaian apakah pola kerja dan manajemen yang digunakan sudah tepat untuk mencapai tujuan, mengetahui kaitan antara kegiatan dengan tujuan untuk memperoleh ukuran kemajuan [7]

2. Internet of Things (IoT)

Internet of Things (IoT) merupakan suatu jaringan yang menghubungkan berbagai objek yang memiliki identitas pengenalan serta alamat IP, sehingga dapat saling berkomunikasi dan bertukar informasi mengenai dirinya maupun lingkungan yang diinderanya. Objek-objek dalam IoT dapat menggunakan maupun menghasilkan layanan-layanan dan saling bekerjasama untuk mencapai suatu tujuan bersama.[8].

3. *QoS*

Quality Of Service (QoS) merupakan metode pengukuran tentang seberapa baik jaringan dan merupakan suatu usaha untuk mendefinisikan karakteristik dan sifat dari satu servis. *QoS* digunakan untuk mengukur sekumpulan atribut kinerja yang telah dispesifikasikan dan diasosiasikan dengan suatu servis. *QoS* didesain untuk membantu end user menjadi lebih produktif dengan memastikan bahwa user mendapatkan kinerja yang handal dari aplikasi-aplikasi berbasis jaringan. *QoS* mengacu pada kemampuan jaringan untuk menyediakan layanan

yang lebih baik pada trafik jaringan tertentu melalui teknologi yang berbeda-beda. *QoS* menawarkan kemampuan untuk mendefinisikan atribut-atribut layanan jaringan yang disediakan, baik secara kualitatif maupun kuantitatif (Ferguson & Huston, 1998). Ada beberapa parameter dalam Quality of Service, seperti Throughput, Packet Loss, Delay (Latency), Jitter dan hasilnya nanti akan distandarisasikan menggunakan standar dari TIPHON[9],[10].

TABEL I
STANDAR TIPHON

Kategori	Throughput (bps)	Delay(ms)	Packet Loss
Sangat baik	100	<150 ms	0%
Baik	75	150-300 ms	3%
Sedang	50	300-450 ms	15%
Jelek	<25	>450 ms	25%

4. *NodeMCU ESP8266*

NodeMCU merupakan sebuah open source platform IoT dan pengembangan kit yang menggunakan bahasa pemrograman Lua untuk membantu dalam membuat prototype produk IoT atau bisa dengan memakai sketch dengan arduino IDE. Pengembangan kit ini didasarkan pada modul ESP8266, yang mengintegrasikan GPIO, PWM (Pulse Width Modulation), IIC, 1-Wire dan ADC (Analog to Digital Converter) semua dalam satu board. GPIO NodeMCU ESP8266.

5. *Sensor Ultrasonik HCSR04*

Sensor ultrasonik tipe HCSR04 merupakan perangkat yang digunakan untuk mengukur jarak dari suatu objek. Dalam hal ini digunakan untuk mengukur ketinggian permukaan air dari Sensor. Kisaran jarak yang dapat diukur sekitar 2-450 cm. Perangkat ini menggunakan dua pin digital untuk mengkomunikasikan jarak yang terbaca. Prinsip kerja sensor ultrasonik ini bekerja dengan mengirimkan pulsa ultrasonik sekitar 40 KHz, kemudian dapat memantulkan pulsa echo kembali, dan menghitung waktu yang diambil dalam mikrodetik.

6. *Buzzer*

Buzzer merupakan sebuah modul komponen elektronika kategori transduser, yang bekerja dengan cara mengubah sinyal elektrik menjadi sebuah gelombang suara. Buzzer biasa difungsikan sebagai alarm sinyal. Biasa di implementasikan pada project penelitian sebagai sebuah indicator terhadap suatu kondisi.

7. *Protokol MQTT*

Protokol MQTT (Message Queue Telemetry Transport) adalah protokol pesan ringan (lightweight) berbasis publish-subscribe digunakan di atas protokol TCP/IP. Protokol ini mempunyai ukuran paket data low overhead. MQTT bersifat terbuka, simpel dan didesain agar mudah untuk diimplementasikan, yang mampu menangani ribuan client jarak jauh dengan hanya satu server [5].

8. *Protokol HTTP*

HTTP atau Hypertext Transfer Protocol, pada awalnya dikembangkan untuk mempublikasikan dan mengunduh halaman HTML. Saat ini, HTTP tidak hanya digunakan sebagai protokol untuk menangani halaman web, tetapi juga dimanfaatkan untuk mentransfer data antara klien dan server. HTTP menentukan protokol dalam melakukan permintaan (request) dan tanggapan (response) di antara klien dan server.

Dengan menggunakan HTTP, terdapat tiga jenis pesan yang dapat dipertukarkan antara klien dan server, yaitu GET, POST, dan PUT. Metode GET digunakan oleh klien untuk meminta data dari server. Sementara itu, metode POST dan PUT digunakan untuk mengunggah (upload) data ke server [11].

9. *WireShark*

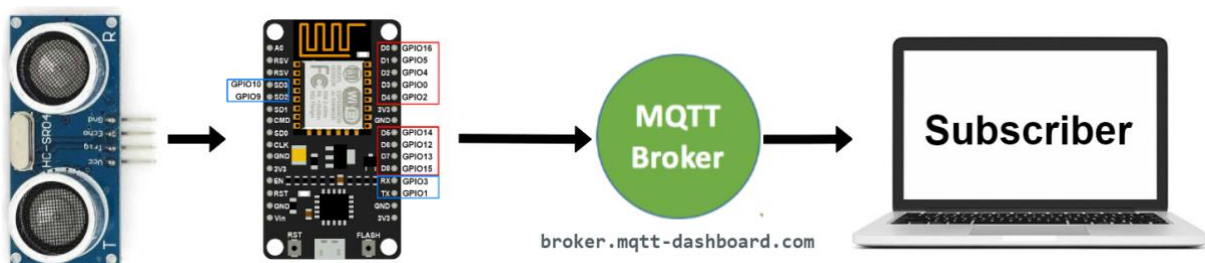
Wireshark adalah penganalisa protokol jaringan terdepan di dunia. Ini memungkinkan Anda melihat apa yang terjadi di jaringan Anda pada tingkat mikroskopis. Ini adalah standar de facto (dan seringkali de jure) di banyak industri dan institusi pendidikan. Pengembangan Wireshark berkembang pesat berkat kontribusi para pakar jaringan di seluruh dunia [12].

10. *Arduino IDE*

Arduino IDE merupakan software pengembangan terintegrasi yang hadir untuk perangkat Arduino yang digunakan untuk membantu memberikan kode mikrokontroler Arduino Integrated Development Environment (IDE) Arduino memiliki peran penting dalam proses pemrograman, kompilasi biner, dan unduhan memori mikrokontroler. Platform ini tidak hanya menyediakan berbagai modul pendukung seperti sensor, monitor, dan pembaca, tetapi juga telah menjadi pilihan utama bagi banyak profesional. Daya tarik utama Arduino bagi banyak pengguna adalah sifatnya yang open source, mencakup baik sisi perangkat keras (hardware) maupun perangkat lunak (software) [13].

C. *Metode Penelitian*

Metode penelitian yang dilakukan peneliti ini menggunakan metode penelitian eksperimental dengan metode pengembangan prototipe rancang bangun alat simulasi. Metode eksperimen adalah untuk mengetahui akibat dari perlakuan yang diberikan terhadap suatu hal yang sedang diteliti. Pengukur parameter QoS menggunakan Wireshark, program sniffer freeware. Wireshark adalah penganalisis paket jaringan gratis dan open-source yang dapat digunakan untuk menganalisis masalah jaringan atau bisa disebut juga *packet analyzer* mengembangkan perangkat lunak dan protokol komunikasi, serta untuk pendidikan. Wireshark bersifat lintas platform dan menggunakan pcap untuk menangkap paket jaringan. Wireshark dapat berjalan di hampir semua sistem operasi yang tersedia [10]. Protokol MQTT menggunakan konsep publish dan subscribe. Dalam skenario publish dan subscribe, ESP8266 mengirimkan data/topik ke broker MQTT. Broker MQTT kemudian menyimpan data/topik tersebut. Jika ada klien yang berlangganan topik, broker MQTT akan meneruskan data/topik tersebut ke klien tersebut untuk lebih jelasnya dapat dilihat gambar 2 berikut.



Gambar 1. Topologi Protokol MQTT

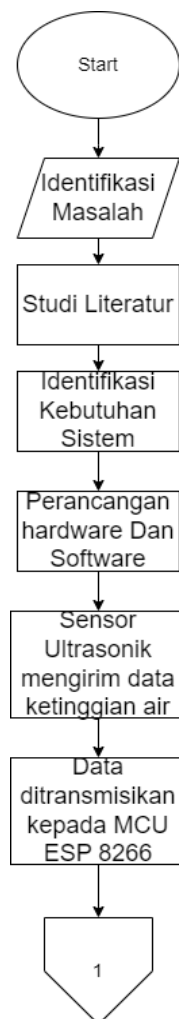
Web server adalah perangkat yang menawarkan layanan penyimpanan untuk halaman web sehingga browser dapat mengaksesnya. Server web dapat diakses di browser protokol HTTP itu sendiri. Dengan server web klien, halaman web tersedia untuk diakses. Server web dapat menerima permintaan HTTP dari pengguna dan membalas dengan menyediakan kode HTML.

Server web menggunakan protokol HTTP (Hypertext Transfer Protocol) untuk berkomunikasi dengan browser. HTTP adalah protokol yang mendefinisikan bagaimana permintaan dan respons web ditangani, untuk lebih jelasnya dapat dilihat Gambar 3 berikut

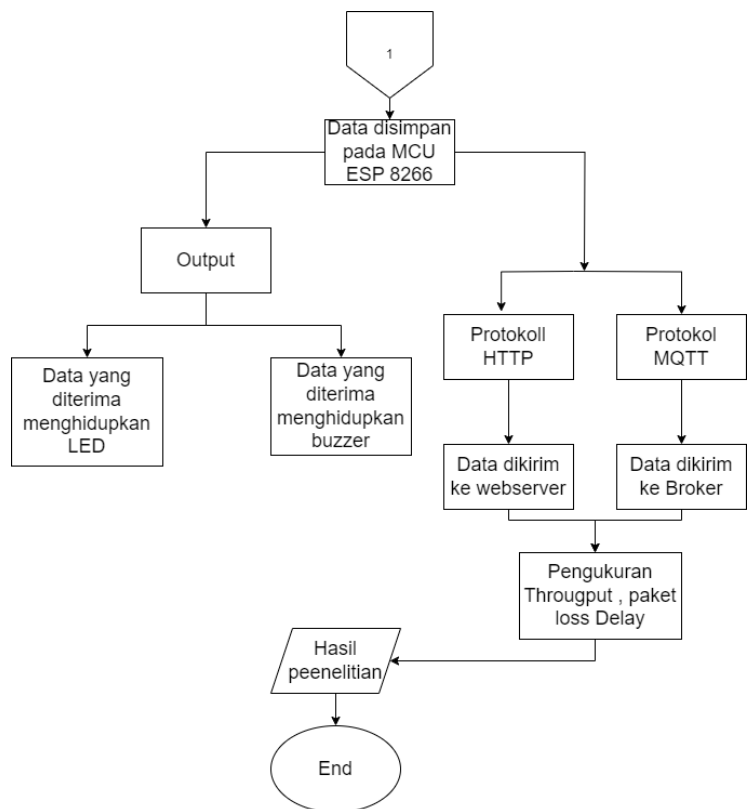


Gambar 2. Topologi Protokol HTTP

Komunikasi HTTP sebagian besar dimulai oleh pengguna dan yang mengirimkan permintaan untuk dikirimkan keserver, konsep request dan respon. Adapun langkah-langkah dalam penelitian ini yaitu Studi Literatur, identifikasi masalah Identifikasi Kebutuhan Sistem, Perancangan Hardware, Perancangan Software, Pengujian Sistem, Hasil Penelitian Untuk Diagram alir penelitian dapat dilihat gambar 4 dan gambar 5 berikut.



Gambar 3. Diagram Alir Penelitian

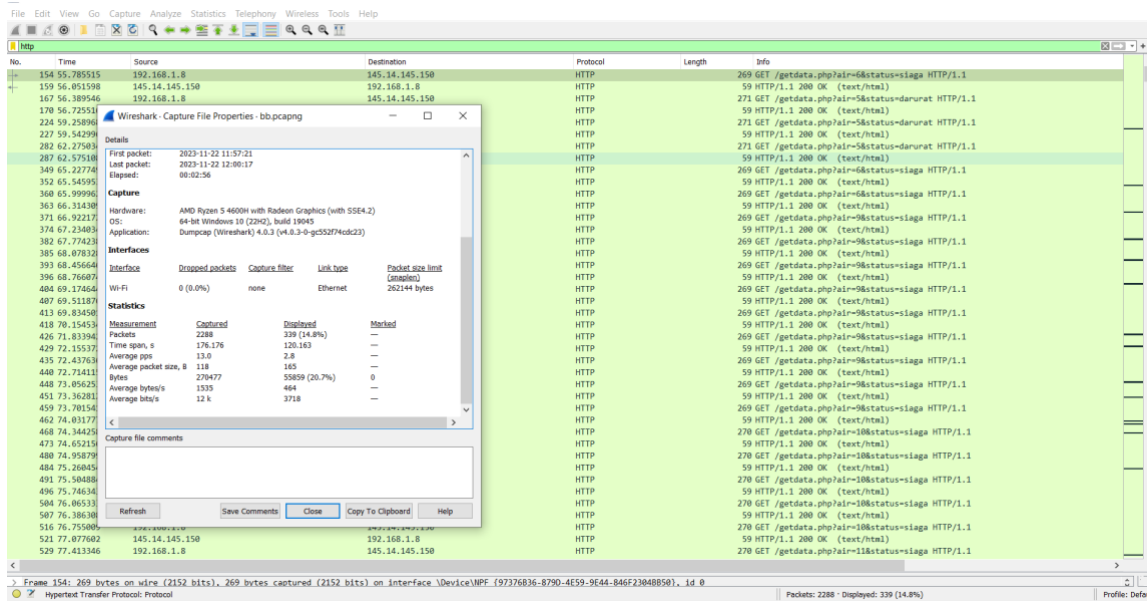


Gambar 4. Lanjutan Diagram Alir Penelitian

NodeMCU ESP8266 dan Sensor Ultrasonik HCSR04 bekerja bersama dalam konteks IoT untuk membentuk sistem pendeteksi banjir yang responsif dan efisien. NodeMCU bertindak sebagai pusat pengendalian yang mengelola koneksi internet dan menerima data dari sensor ultrasonik. Sensor Ultrasonik HC-SR04 berfungsi sebagai perangkat pengukur ketinggian permukaan air. Mekanisme menguji yang dilakukan yaitu perangkat keras dengan menguji koneksi Internet, Internet sudah bisa terhubung ke mikrokontroler NodeMCU ESP8266. Kemudian uji coba sensor HC-SR04 apakah sensor sudah bisa mengukur ketinggian permukaan air dan dapat mengirimkannya ke NodeMCU ESP8266 yang terkoneksi dengan internet. Penerapan protokol MQTT dan HTTP untuk transmisi pesan akan diuji Quality of Service (QoS) dengan menggunakan berbagai metrik seperti Throughput, Packet Loss, Delay. Data Packet ditangkap menggunakan Wireshark yang akan menangkap lalu lintas jaringan. Pengujian dilakukan Selama 2 menit dengan 3 kali pengujian. Data yang diambil yaitu Throughput, Packet Loss Delay. Tujuan dari penelitian ini adalah untuk membandingkan QoS protokol MQTT dan HTTP dan menentukan mana yang lebih cocok untuk studi kasus ini.

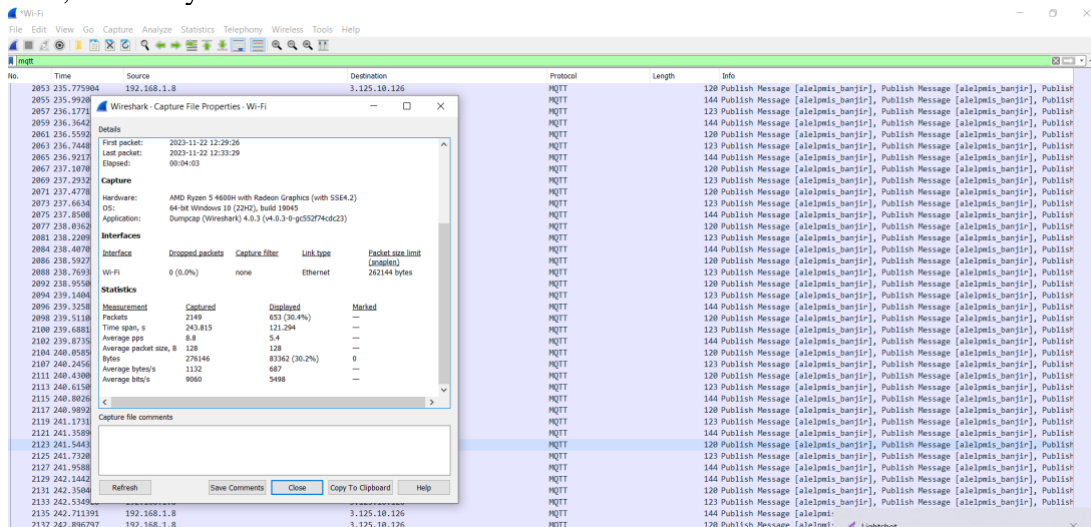
III. HASIL DAN PEMBAHASAN

Dalam pembahasan ini ingin mengukur Quality of service (QoS). Pengujian terbagi menjadi 3 percobaan, yaitu pengujian pada protokol HTTP dilakukan sebanyak 3 kali dan protokol MQTT dilakukan sebanyak 3 kali, masing – masing pengujian menguji coba kualitas layanan pada tiga kategori parameter dari delay, throughput dan packet loss. Pengujian dilakukan dengan prototype dimana di uji ketinggian permukaan air wadah. Masing – masing pengujian dilakukan selama 2 menit lamanya untuk mengetahui hasil dari kualitas layanan. Berikut pada gambar 6 dan 7 pengujian protokol MQTT dan Protokol HTTP menggunakan wireshark.



Gambar 5. Pengujian Protokol HTTP dengan Wireshark

Gambar 6, proses pemantauan pengujian protokol HTTP melalui aplikasi Wireshark. Gambar ini menampilkan tampilan layar aplikasi Wireshark yang mencapture dan menganalisis paket data yang dikirim dan diterima selama uji coba protokol HTTP. Informasi tentang throughput, packet loss, dan delay.



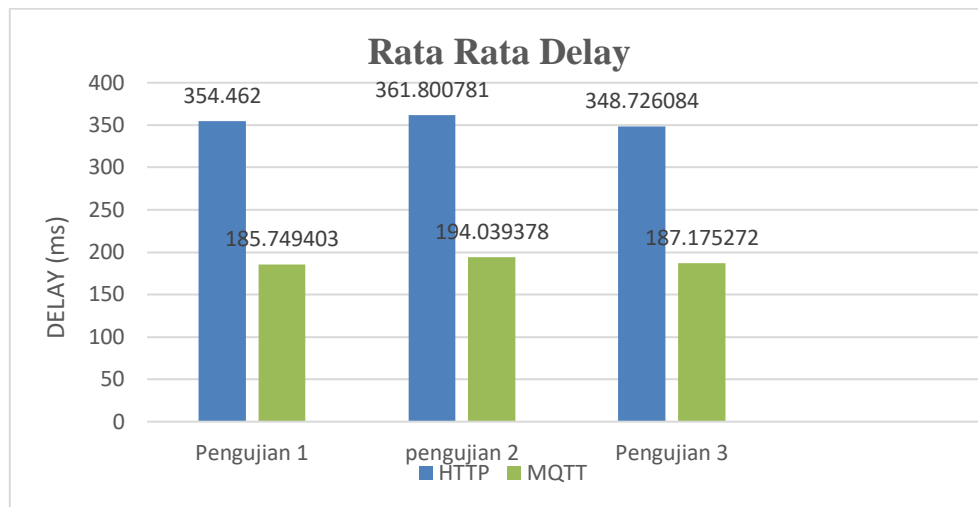
Gambar 6. Pengujian Protokol MQTT dengan WireShark

Dalam Gambar 7, tampak dengan rinci proses pemantauan pengujian protokol MQTT menggunakan aplikasi Wireshark. Gambar ini menunjukkan layar aplikasi Wireshark yang secara aktif memantau dan menganalisis paket data yang dikirim dan diterima selama uji coba protokol MQTT. Melalui visualisasi yang terperinci pada antarmuka aplikasi, dapat teridentifikasi informasi tentang throughput, packet loss, dan delay.

A. Perbandingan Delay pada HTTP dan MQTT

Delay (Latency) merupakan durasi yang dibutuhkan packet/data untuk menempuh perjalanan dari source ke destination. Faktor-faktor yang dapat memengaruhi Delay yaitu jarak, kongesti, media fisik, dan waktu proses yang diperlukan [5]. Rumus untuk menghitung Delay dapat dilihat dari persamaan 1 sebagai berikut.

$$Rata - Rata Delay = \frac{Total Delay}{Total Paket Diterima} \tag{1}$$



Gambar 7. Perbandingan Delay Protokol MQTT dan Protokol HTTP

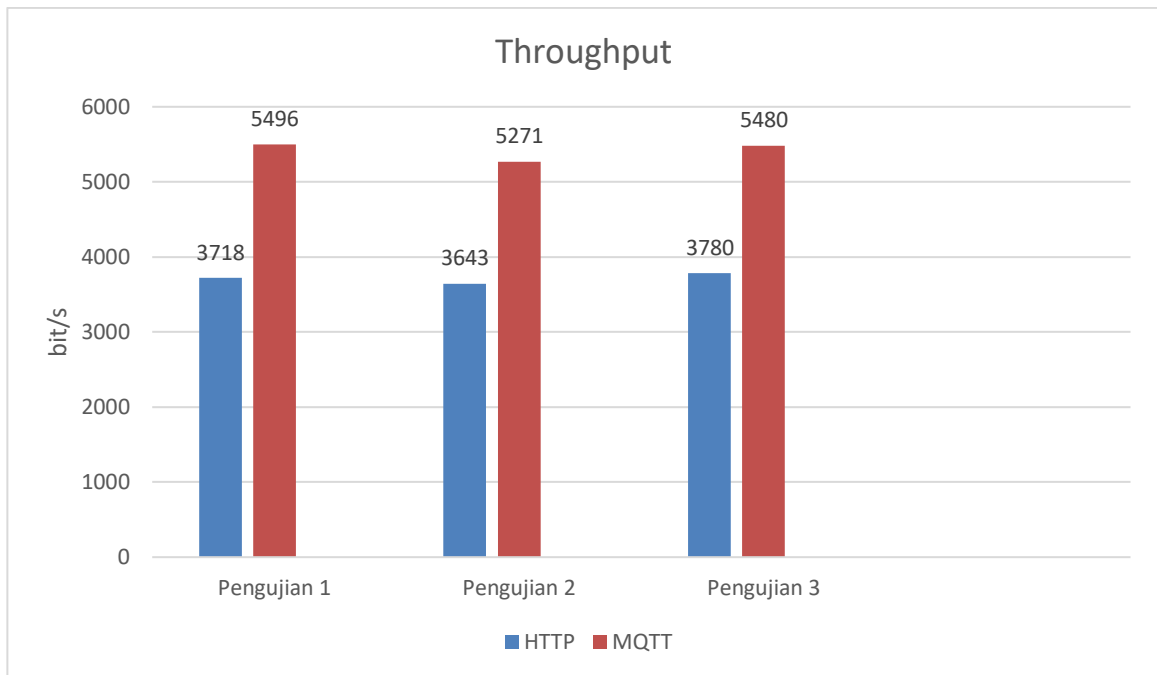
Dari Gambar 8 pengujian Delay antara Protokol MQTT dan Protokol HTTP pengujian terhadap delay dilakukan dengan menghitung rata-rata waktu jeda antara pengiriman satu paket dengan paket berikutnya yang dilakukan oleh mikrokontroler ke webserver atau broker. Seperti yang terlihat dalam gambar 5, delay yang tercatat pada protokol HTTP. Pengujian delay dilakukan untuk setiap protokol, dan terlihat bahwa delay terkecil terjadi saat menggunakan protokol MQTT. Pada pengujian pertama dengan protokol MQTT, waktu delay mencapai 185,749403 milisecond dengan total paket sebanyak 687. Sedangkan untuk delay tertinggi didapatkan dari protokol http Pada pengujian kedua, waktu delay mencapai 361,800781 milisecond detik dengan total paket yang diproses sebanyak 293.

Hasil rata-rata delay yang diperoleh melalui penggunaan protokol HTTP dan MQTT dengan standar THIPON pada table 1. Rata-rata delay Protokol MQTT tercatat kecil dari 300 milisecond, sesuai dengan standar THIPON, menunjukkan bahwa kualitas layanan yang baik, sedangkan protokol HTTP Rata-rata delay tercatat kecil dari 450 milisecond ,menunjukkan bahwa kualitas layanan yang sedang. Kedua protokol memiliki nilai rata-rata delay yang cukup jauh perbedaannya, perbandingan antara protokol HTTP dan MQTT menunjukkan bahwa protokol MQTT memiliki performa yang lebih baik dengan nilai delay yang lebih rendah. Oleh karena itu, dapat disimpulkan bahwa dari segi parameter delay, protokol MQTT unggul dibandingkan dengan HTTP.

B. Perbandingan Throughput pada Protokol HTTP dan Protokol MQTT

Throughput adalah laju transfer data efektif yang diukur dalam bps (bit per second). Throughput mencakup total kedatangan paket yang berhasil diamati di tujuan selama suatu interval waktu, dibagi oleh durasi interval waktu tersebut. Rumus perhitungan throughput dapat ditemukan pada Persamaan 2 berikut ini.

$$Throughput = \frac{(Paket Data Diterima)}{Lama Pengamatan} \tag{2}$$



Gambar 8. Perbandingan Throughput Protokol HTTP dan Protokol MQTT

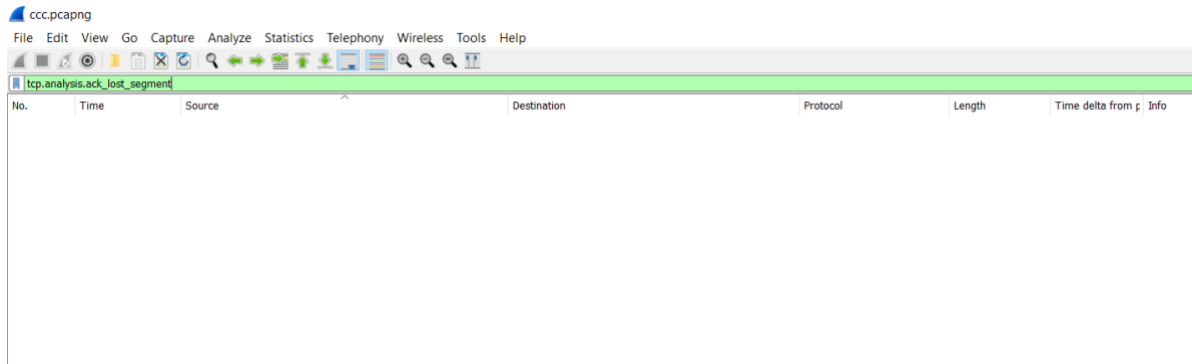
Gambar 9., dapat disimpulkan bahwa throughput yang dihasilkan oleh MQTT menunjukkan performa yang lebih baik jika dibandingkan dengan throughput yang dihasilkan oleh HTTP, berdasarkan pengaruh waktu penanganan sistem. Contohnya, pada pengujian 1 MQTT pada sistem pemantauan udara, throughput mencapai 5496 bit/s, pada pengujian 2 mencapai 5271 bit/s, dan pada pengujian 3 mencapai 5480 bit/s. Hal ini mencerminkan bahwa kecepatan pengiriman data MQTT lebih tinggi daripada HTTP.

C. Pengujian Packet Loss Protokol HTTP dan Protokol MQTT

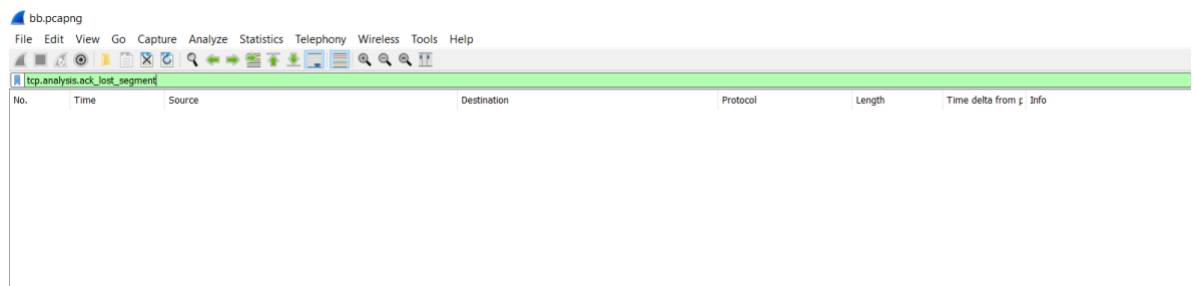
Paket loss adalah sebuah parameter yang mengindikasikan suatu situasi di mana jumlah paket yang tidak berhasil dikirim bisa terjadi akibat tabrakan dan kepadatan pada jaringan[14] Rumus perhitungan hilangnya paket dapat ditemukan pada Persamaan 3 di bawah ini.

$$Packet\ Loss = \frac{(Paket\ data\ dikirim - Paket\ data\ diterima) \times 100\%}{Paket\ data\ dikirim} \tag{3}$$

kehilangan paket biasanya terjadi saat pengiriman data dengan ukuran yang besar. Hal ini hanya berdampak pada pengurangan ukuran paket dan bukan pada kehilangan paket itu sendiri. Padahal Delay pada pengujian protokol HTTP kurang baik, mengingat pada uji throughput, delay sebelumnya Namun, pada setiap uji tersebut tidak terjadi kehilangan paket atau pengiriman paket yang tidak berhasil. Dapat dilihat bahwa tidak ada paket yang mengalami kehilangan dapat dilihat pada gambar 10 dan 11 berikut.

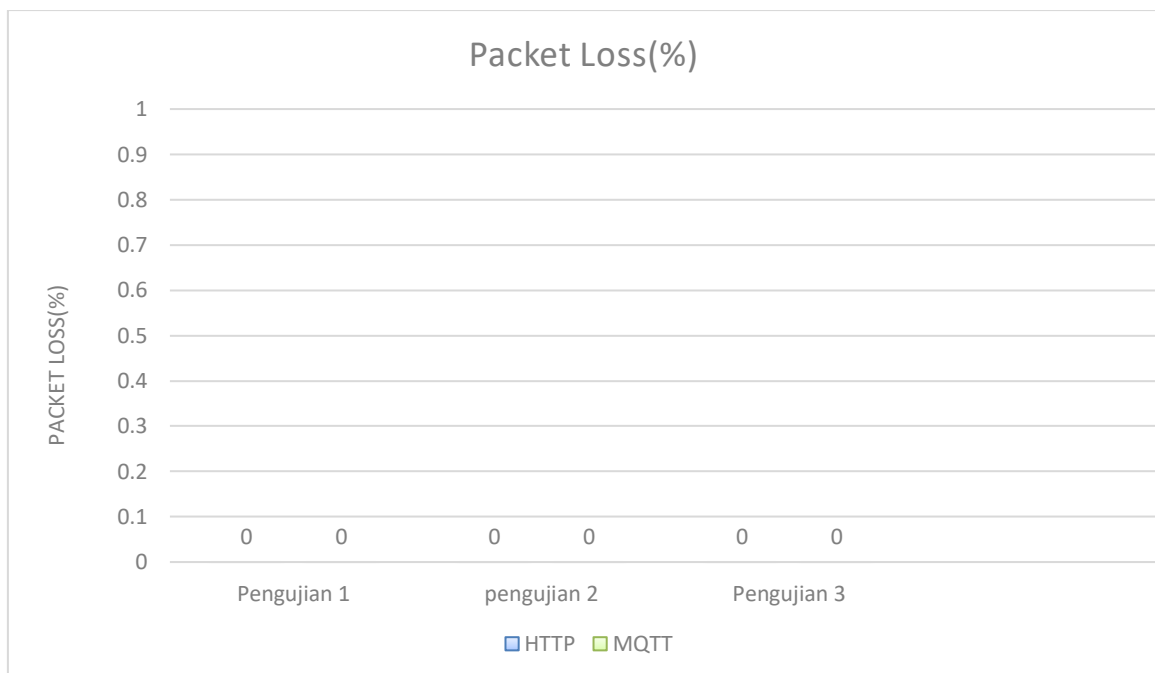


Gambar 9. Tidak ada Paket Yang Hilang Protokol HTTP



Gambar 10. Tidak ada Paket Yang Hilang Protokol MQTT

Secara keseluruhan, dalam proses pengiriman data, tidak ada gangguan yang menyebabkan hilangnya paket yang dikirim, sehingga nilai packet lossnya adalah 0%. Dari grafik yang diperoleh, dapat diamati bahwa kedua protokol memiliki tingkat packet loss yang sama, yaitu 0%. Oleh karena itu, berdasarkan standar THIPON, keduanya dapat dikategorikan sebagai memiliki indikasi "Sangat Baik" dalam hal packet loss. Dari segi kategori packet loss, dapat disimpulkan bahwa performa kualitas layanan dari kedua protokol tersebut setara. Grafik hasil packet loss dari protokol HTTP dan MQTT dapat dilihat pada ilustrasi berikut.



Gambar 11. Packet Loss

IV. KESIMPULAN

Melalui rangkaian tahapan penelitian ini, dapat disimpulkan bahwa integrasi konsep Internet of Things dengan penggunaan protokol MQTT memberikan hasil yang optimal dalam pengembangan sistem pendeteksi banjir. Analisis parameter antara protokol MQTT dan HTTP menunjukkan bahwa nilai delay rata-rata tertinggi pada HTTP mencapai 361,800781, sedangkan pada MQTT hanya sebesar 194,039378, menunjukkan bahwa MQTT mengungguli HTTP dalam hal respons waktu dan Pada pengujian throughput, nilai throughput tertinggi dicapai oleh MQTT, yakni mencapai 5496 bit/s, sedangkan HTTP hanya mencapai nilai throughput sebesar 3780 bit/s, tanpa kehilangan paket pada parameter packet loss. Hasil ini memberikan dasar kuat untuk mempertimbangkan protokol MQTT sebagai pilihan utama dalam membangun sistem pendeteksi yang efisien dan handal, khususnya dalam konteks mitigasi risiko banjir. Kesimpulan ini menegaskan bahwa penerapan konsep Internet of Things dengan memanfaatkan protokol MQTT merupakan langkah yang lebih optimal dalam meningkatkan respons sistem pendeteksi banjir. Namun, perlu dicatat bahwa batasan masalah penelitian ini mencakup hanya pengembangan prototype dan penghitungan Quality of Service (QoS) yang terfokus pada parameter delay, throughput, dan packet loss. Melibatkan aspek-aspek lain dari QoS atau melibatkan fase implementasi penuh sistem mungkin memerlukan penelitian lanjutan untuk memperluas cakupan evaluasi dan validasi hasil.

REFERENSI

- [1] L. D. Yadnya Prandini and N. L. W. Sayang Telagawathi, "Faktor - Faktor Yang Menentukan Keputusan Pembelian Produk Virtual Dalam Online Games Mobile Legends," *Bisma J. Manaj.*, vol. 7, no. 2, p. 294, 2021, doi: 10.23887/bjm.v7i2.32500.
- [2] A. K. Hasiholan Chrisyantar, Primananda Rakhmadhany, "Implementasi Konsep Internet of Things pada Sistem Monitoring Banjir menggunakan Protokol MQTT," *Pengemb. Teknol. Inf. dan Ilmu Komput.*, vol. 2, no. 12, pp. 6128–6135, 2018, [Online]. Available: <https://j-ptiik.ub.ac.id/index.php/j-ptiik/article/download/3529/1389/>
- [3] K. S. Salamah and S. Anwar, "Rancang Bangun Sistem Pendeteksi Banjir Otomatis Berbasis Internet Of Things," *J. Teknol. Elektro*, vol. 12, no. 1, p. 40, 2021, doi: 10.22441/jte.2021.v12i1.008.
- [4] S. Bagas Bhaskoro, H. Supriyanto, B. B. Aji, and B. Pamungkas, "Perbandingan Performansi Latency Protokol Komunikasi HTTP Dan MQTT Pada Internet of Things," *J. Teknol. Terap.*, vol. 8, no. 2, pp. 82–89, 2022.
- [5] I. F., "Analisis performansi QoS MQTT pada sistem monitoring sungai Tugas Akhir diajukan untuk memenuhi salah satu syarat memperoleh gelar sarjana Program Studi Sarjana S1 Teknik Informatika Fakultas Informatika Universitas Telkom Bandung," vol. 6, no. 1, pp. 2006–2020, 2019.
- [6] S. P. Windiastik, E. N. Ardhana, and J. Triono, "Perancangan Sistem Pendeteksi Banjir Berbasis Iot (Internet of Thing)," *Semin. Nas. Sist. Inf.*, vol. 3, no. September, pp. 1925–1931, 2019, [Online]. Available: <https://jurnalfti.unmer.ac.id/index.php/senasif/article/view/256>
- [7] D. A. Megawaty and M. E. Putra, "Aplikasi Monitoring Aktivitas Akademik Mahasiswa Program Studi Informatika Universitas Xyz Berbasis Android," *J. Inform. dan Rekayasa Perangkat Lunak*, vol. 1, no. 1, pp. 65–74, 2020, doi: 10.33365/jatika.v1i1.177.
- [8] F. Adani and S. Salsabil, "Internet of Things: Sejarah Teknologi Dan Penerapannya," *Isu Teknol. Stt Mandala*, vol. 14, no. 2, pp. 92–99, 2019.
- [9] M. Y. Simargolang and A. Widarma, "Quality of Service (QoS) for Network Performance Analysis Wireless Area Network (WLAN)," *CESS (Journal Comput. Eng.*

- Syst. Sci.*, vol. 7, no. 1, p. 162, 2022, doi: 10.24114/cess.v7i1.29758.
- [10] Satria Turangga, Martanto, and Yudhistira Arie Wijaya, “Analisis Internet Menggunakan Paramater Quality of Service Pada Alfamart Tuparev 70,” *JATI (Jurnal Mhs. Tek. Inform.*, vol. 6, no. 1, pp. 392–398, 2022, doi: 10.36040/jati.v6i1.4693.
- [11] M. Katoningati *et al.*, “Analisis Layer Aplikasi (Protokol HTTP) menggunakan Wireshark,” *J. Elektro Smart*, vol. 1, no. 1, pp. 13–15, 2021, [Online]. Available: <https://ilmukomputer.org/2013/06/02/monitoring-layer-aplikasi->
- [12] R. Hanipah and H. Dhika, “Analisa Pencegahan Aktivitas Ilegal Didalam Jaringan Dengan Wireshark,” *DoubleClick J. Comput. Inf. Technol.*, vol. 4, no. 1, p. 11, 2020, doi: 10.25273/doubleclick.v4i1.5668.
- [13] Slamet Purwo Santoso and Fajar Wijayanto “RANCANG BANGUN AKSES PINTU DENGAN SENSOR SUHU DAN HANDSANITIZER OTOMATIS BERBASIS ARDUINO” Vol 10 No 1 Januari 2022.
- [14] L. Nurfiqin, “Analisis Quality Of Service (QoS) Protokol MQTT dan HTTP Pada Sistem Smart Metering Arus Listrik,” *J. Repos.*, vol. 3, no. 1, pp. 121–130, 2020, doi: 10.22219/repositor.v3i1.1084.

UCAPAN TERIMA KASIH

Terimakasih kepada pihak-pihak yang telah mendukung terlaksananya penelitian ini, khususnya kepada Tim *Jurnal Informatika Polbeng* yang telah memfasilitasi penerbitan jurnal ini.