

Implementation of the MQTT Protocol on GPS-NEO6M as a Motorcycle Monitoring Design

[Implementasi Protokol MQTT pada GPS-NEO6M sebagai Rancang Bangun Monitoring Sepeda Motor]

Aldi Viantara¹⁾, Mochamad Alfian Rosid^{*2)}

1)Program Studi Informatika, Universitas Muhammadiyah Sidoarjo, Indonesia

2)Program Studi Informatika, Universitas Muhammadiyah Sidoarjo, Indonesia

*Email Penulis Korespondensi: alfanrosid@umsida.ac.id

Abstract. Vehicles have become an essential need that facilitates human mobility. However, security threats such as motor vehicle theft are a significant problem. IoT security technologies, such as emergency features that turn off vehicles and GPS tracking using the HTTPS protocol, are becoming relevant solutions. Previous research shows the superiority of the MQTT protocol in IoT based on the lower latency difference. This research aims to utilize the MQTT protocol in building a motorbike monitoring system. This implementation involves using node js to display the location coordinates of the vehicle, allowing the vehicle owner to track the vehicle's position in the event of a theft. The test analysis results show that the hardware connected to WiFi and the MQTT broker was successful, but the GPS signal acquisition time varied between 9.07 seconds, 4.73 seconds and 11.03 seconds. The MQTT protocol using the EMQX broker on the VPS successfully sends data from the Neo6M GPS module to the web client with Node.js to display vehicle coordinates. Testing three duration and distance scenarios highlighted the performance differences between MQTT and HTTP: Average latency of MQTT (1.58 ms - 1,645 ms) and HTTP (0.455 ms - 0.46 ms), as well as average throughput of MQTT (621.48 bps - 676.385 bps) and HTTP (2378.355 bps - 2409.175 bps). Even though HTTP has lower latency, MQTT consistently shows higher throughput. MQTT is able to operate with a smaller bandwidth while still providing superior throughput performance, albeit with a slight increase in latency.

Keywords - IoT Security; MQTT Protocol; Vehicle Security; Throughput and Latency Comparison; GPS Tracking; Node.js.

Abstrak. Kendaraan sudah menjadi kebutuhan esensial yang memfasilitasi mobilitas manusia. Namun, ancaman keamanan seperti pencurian kendaraan bermotor menjadi masalah yang signifikan. Teknologi keamanan IoT, seperti fitur darurat yang mematikan kendaraan dan pelacakan GPS menggunakan protokol HTTPS, menjadi solusi yang relevan. Penelitian sebelumnya menunjukkan keunggulan protokol MQTT dalam IoT berdasarkan selisih latency yang lebih rendah. Penelitian ini bertujuan memanfaatkan protokol MQTT dalam membangun sistem monitoring sepeda motor. Implementasi ini melibatkan penggunaan node js untuk menampilkan lokasi koordinat kendaraan, memungkinkan pemilik kendaraan melacak posisi kendaraan saat terjadi pencurian. Hasil analisis pengujian menunjukkan bahwa perangkat keras terkoneksi dengan WiFi dan broker MQTT berhasil, namun waktu akuisisi sinyal GPS bervariasi antara 9.07 detik, 4.73 detik, dan 11.03 detik. Protokol MQTT menggunakan broker EMQX pada VPS berhasil mengirimkan data dari modul GPS Neo6M ke web client dengan Node.js untuk menampilkan koordinat kendaraan. Pengujian tiga skenario durasi dan jarak menyoroti perbedaan kinerja antara MQTT dan HTTP: Latency rata-rata MQTT (1.58 ms - 1.645 ms) dan HTTP (0.455 ms - 0.46 ms), serta throughput rata-rata MQTT (621.48 bps - 676.385 bps) dan HTTP (2378.355 bps - 2409.175 bps). Meskipun HTTP memiliki latency yang lebih rendah, MQTT secara konsisten menunjukkan throughput yang lebih tinggi. MQTT mampu beroperasi dengan bandwidth yang lebih kecil namun tetap memberikan kinerja throughput yang unggul, meskipun dengan sedikit peningkatan latency.

Kata Kunci - IoT Security; MQTT Protocol; Vehicle Security; Throughput and Latency Comparison; GPS Tracking; Node.js.

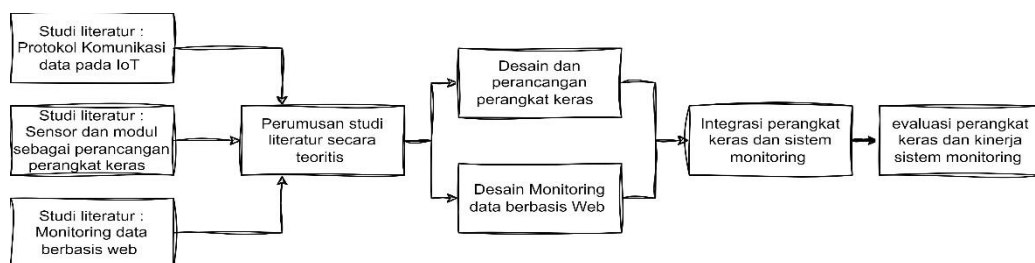
I. PENDAHULUAN

Kendaraan telah menjadi keharusan dalam kehidupan sehari-hari yang penuh aktivitas. Kendaraan memainkan peran penting dalam memfasilitasi mobilitas manusia, baik untuk perjalanan jauh maupun dekat. Kendaraan memungkinkan kita menjalani kehidupan dengan lebih efisien dan fleksibel. Yang memungkinkan kita untuk menghubungkan tempat-tempat penting tanpa terhalang oleh batasan geografis [1]. Tantangan yang perlu diatasi, bagaimanapun, terkait dengan kelanjutan penggunaan kendaraan. Salah satunya adalah masalah kejahatan seperti pencurian dan begal kendaraan bermotor menjadi masalah yang sangat menghantui masyarakat. Tingkat kejahatan yang tinggi ini tidak hanya mengancam harta benda, tetapi juga mengganggu rasa aman masyarakat [2]. Oleh karena itu, fitur keamanan pada sepeda motor menjadi alat yang dapat diandalkan untuk mencegah perilaku kriminal seperti begal atau perampasan kendaraan bermotor. Dalam sektor IoT, contohnya, teknologi keamanan membuat pengguna

kendaraan bermotor merasa aman saat terjadi keadaan darurat, dengan hanya menekan tombol darurat dan motor akan mati dalam waktu 10 detik[3]. Selain itu, salah satu solusi untuk masalah ini adalah menggunakan protokol HTTPS atau disebut juga Hypertext Transfer Protocol secure yang bekerja dengan cara request dan respon secara berkala untuk melacak posisi kendaraan dengan GPS. Data koordinat lokasi dapat dikirim ke web server dan ditampilkan dalam bentuk peta menggunakan Google Maps [4]. Berbicara tentang Internet of Things, atau Internet of Things, tidak bisa lepas dari kata "internet" protokol yang memungkinkan perangkat berkomunikasi satu sama lain. Beberapa protokol yang paling umum untuk komunikasi dan pemantauan Internet of Things adalah NFC, Bluetooth Low Energy, MQTT, HTTP, dan HTTPS[5]. Penelitian terkait IoT dan penerapan protokol Message Queuing Telemetry Transport (MQTT) yang bersifat publish dan subscribe serta HTTP dalam berbagai aplikasi telah dilakukan oleh beberapa peneliti, mulai dari aspek keamanan pada sepeda motor dengan SIM GSM [6], perbandingan performa latency antara protokol komunikasi HTTP dan MQTT pada Internet of Things, hingga implementasi MQTT dalam aplikasi perkebunan pintar dan greenhouse[7]. Meskipun penelitian ini menunjukkan kelebihan protokol MQTT, seperti keunggulan dalam keamanan dan performa latency[8], setiap studi juga mencerminkan keterbatasan, seperti biaya tambahan dalam penggunaan GSM/SMS, keterbatasan uji coba yang tidak memperhitungkan jarak antara server MQTT dan perangkat IoT, serta kebutuhan untuk lebih mengevaluasi efisiensi latency dan bandwidth saat menggunakan MQTT. Secara keseluruhan, penelitian-penelitian ini menyuguhkan wawasan yang mendalam mengenai potensi dan tantangan dalam mengadopsi protokol MQTT dalam konteks aplikasi Internet of Things[9]. Berdasarkan latar belakang diatas, penelitian ini bertujuan untuk memanfaatkan protokol MQTT dalam membangun sistem monitoring sepeda motor. Penelitian ini juga menerapkan nodejs[10] sebagai Bahasa pemrograman pengembangan sistem monitoring sepeda motor sebagai media untuk menampilkan lokasi koordinat kendaraan sehingga ketika terjadi pencurian bisa mengetahui lokasi kendaraan yang dicuri. Dengan demikian diharapkan penelitian ini dapat membantu pemilik kendaraan mengamankan dari tindakan pencurian.

II. METODE

Alur penelitian pada gambar 1 yang dirancang dari tahap awal hingga selesai. Setiap titik dalam alur tersebut menjelaskan proses perancangan sistem dan alat yang bertujuan membentuk prototipe. Penjelasan pada setiap tahap dalam alur penelitian menggunakan metode dibawah ini :



Gambar 1 Alur penelitian

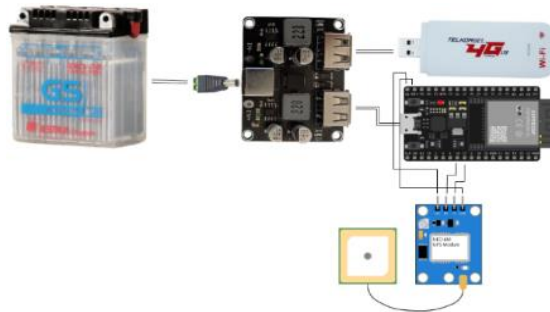
A. Perumusan metode studi literatur secara teoritis

Dalam tahap awal penelitian, fokusnya adalah menganalisis protokol komunikasi umum dalam Internet of Things (IoT), seperti MQTT, CoAP, dan HTTP. melibatkan pengumpulan informasi mendalam tentang berbagai protokol tersebut, menilai keunggulan, kelemahan, serta aplikasi khusus dalam konteks sistem pemantauan sepeda motor. Kemudian, dalam studi literatur mengenai pemilihan modul dan sensor, diputuskan untuk menggunakan modul GPS-Neo6M yang akan terkoneksi dengan ESP32. Modul ini akan mengirimkan data koordinat longitude dan latitude dalam format JSON. Dengan kombinasi ini, diharapkan mendapatkan informasi lokasi yang akurat dari sepeda motor yang sedang dimonitor. Langkah berikutnya adalah pengembangan antarmuka web menggunakan Node.js untuk responsifitas pembuatan halaman website, MySQL sebagai basis data penyimpanan lokasi dari GPS, dan library Leaflet maps untuk visualisasi data lokasi. Harapannya, antarmuka ini memberikan pengalaman yang intuitif bagi pengguna

B. Desain dan perancangan perangkat keras

. Beberapa komponen yang diperlukan untuk penelitian akan seperti pada gambar 2 yang meliputi ESP32, Modul GPS-Neo6M, Modul USB Fast Charger 5V 3.4A QC3.0 24W, dan Modem Stik. Pada Gambar 2, ESP32 berperan sebagai Mikrokontroler yang akan terkoneksi dengan Modul GPS Neo-6M untuk menangkap titik koordinat lokasi.

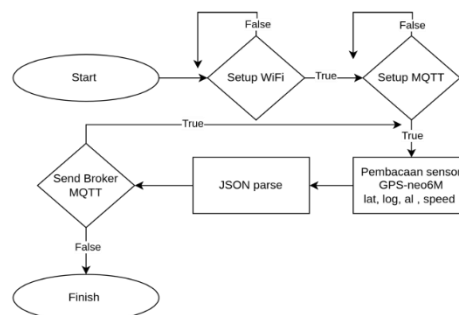
Mikrokontroler ESP32 akan bertindak sebagai pengirim data ke server menggunakan koneksi WiFi yang diakses melalui modem stik sebagai sumber internet. Kedua perangkat, ESP32 dan modem stik, akan ditenagai oleh Modul USB Fast Charger 5V 3.4A QC3.0 24W Charging. Modul ini akan memperoleh daya listrik dari aki motor. Pada table 1 adalah rangkain kable jumper dari esp32 ke modul GPS-Neo6M



Gambar 2. Rangkaian perangkat keras

C. Inisialisai Program ESP32

Pada gambar 3 Program ESP32 bertanggung jawab untuk mengatur koneksi WiFi, mengelola klien MQTT, dan menginisialisasi komunikasi serial untuk modul GPS. Dalam prosesnya, program terus memantau ketersediaan koneksi WiFi, koneksi broker MQTT, dan data sinyal GPS. Program akan mengambil informasi lintang, bujur, ketinggian, dan kecepatan jika data tersedia, lalu mengirimnya ke broker MQTT dalam format JSON. Selain itu, program juga terus memantau koneksi ke broker MQTT mencoba untuk terhubung kembali jika terputus, sambil memberi informasi tentang status koneksi. Semua proses ini berlangsung dalam loop utama program hingga program selesai atau dihentikan.



Gambar 3. Alur program pada ESP32

D. Pembuatan Prototype Web

Data output dari MQTT broker akan disubscribe oleh web menggunakan library dari Node.js, yaitu MQTT js, dan Leaflet.js sebagai maps. Desain web yang interaktif akan meningkatkan pengalaman pengguna secara lebih mendalam. Antarmuka yang dirancang merupakan representasi dari hasil pengumpulan data yang akan diimplementasikan ke dalam suatu sistem. Desain antarmuka diperlukan agar mempermudah interaksi antara pengguna dan sistem.

E. Integrasi Perangkat Keras dengan Sistem Monitoring

Integrasi perangkat keras akan seperti gambar 4 dengan sistem monitoring melibatkan tiga komponen utama. Pertama, NodeMCU ESP32 dan GPS Neo-6M berperan dalam pengambilan data dari modul GPS dan koneksi ke internet. Data seperti longitude, latitude, dan kecepatan akan dikirim ke Broker MQTT dengan pesan terjadwal setiap 5 detik dalam format JSON. Kedua, Broker MQTT menggunakan EMQX di VPS untuk menerima dan menyimpan data tersebut. Terakhir, klien web yang telah direncanakan akan berlangganan (subscribe) ke topik GPS/malih/data di MQTT Broker untuk menampilkan informasi tersebut dalam antarmuka web..



Gambar 4. Alur Integrasi Perangkat Keras dengan Sistem Monitoring

III. HASIL DAN PEMBAHASAN

A. HASIL DAN PEMBAHASAN

Bab ini merangkum hasil dari serangkaian eksperimen dan pengujian yang telah dilakukan dalam konteks implementasi dan evaluasi sistem yang telah dirancang. Hasil-hasil yang diuraikan dalam bab ini mencakup:

1. Pengujian perangkat Keras

Proses pertama melibatkan unggah sketch pada ESP32 melalui Arduino IDE. Rangkaian alat yang telah dirancang, pada gambar 5 dan gambar 6 merupakan hasil dari pengujian perangkat keras. Tahap pertama melibatkan kompilasi sketch dan menunggu hingga modul GPS mendapatkan sinyal yang terbukti dengan lampu indikator pada modul yang akan berkedip. Selanjutnya data akan ditampilkan,



Gambar 5. Pengujian perangkat keras

```

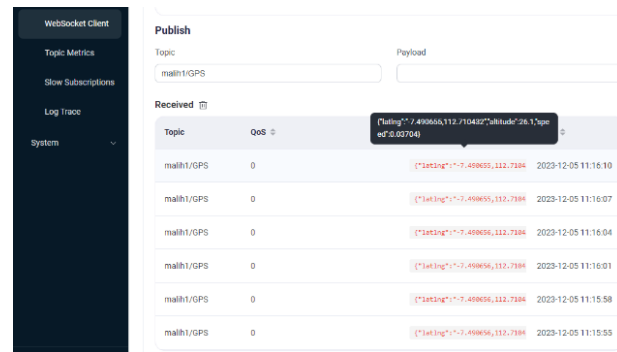
[ESP32] (POWERON_RESET) 160DC:0X13 (SPI_FLASH_BOOT)
configip: 0, SPIWP:0xee
clk_drv:0x00,q_drv:0x00,d_drv:0x00,cs0_drv:0x00,hd_drv:0x00,wp_drv:0x00
mode:DIO, clock div:1
load:0x3fff0030,len:1344
load:0x40078000,len:13964
load:0x40080400,len:3600
entry 0x400805f0
Connecting to Sasayang
.....
WiFi connected
IP address:
192.168.137.187
Attempting MQTT connection...Connected
data GPS: [ {"latlng":"-7.490716,112.710438","altitude":54.5,"speed":0.6482}]
Message sent to MQTT broker
data GPS: [ {"latlng":"-7.490675,112.710396","altitude":50,"speed":1.2038}]
Message sent to MQTT broker
data GPS: [ {"latlng":"-7.490653,112.710399","altitude":48,"speed":0.20372}]

```

Gambar 6. Pengujian serial monitor

2. Implementasi protokol MQTT

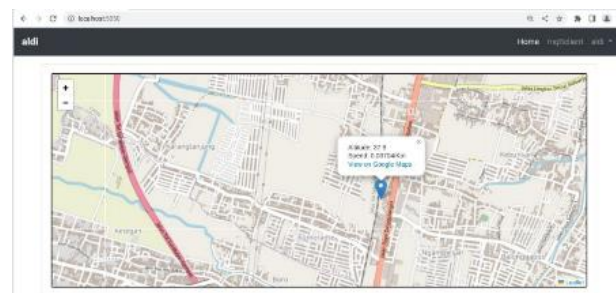
Langkah pertama adalah menginstal broker pada VPS. Langkah ini dimulai dengan mengunduh installer dari broker.emqx.io, kemudian menjalankan proses instalasi. Jika instalasi berhasil, akseslah web dashboard yang telah disediakan melalui web browser menggunakan alamat <http://alamatip:18083>. Setelah itu, periksa bagian klien pada dashboard dan subscribe topik yang sesuai untuk memastikan bahwa data dari modul perangkat keras, telah berhasil mengirimkan data berupa longitude, latitude, dan kecepatan. Hasil pengujian dapat dilihat pada gambar 7



Gambar 7. Hasil dari dashboard EMQX

3. Pengujian Web Dashboard

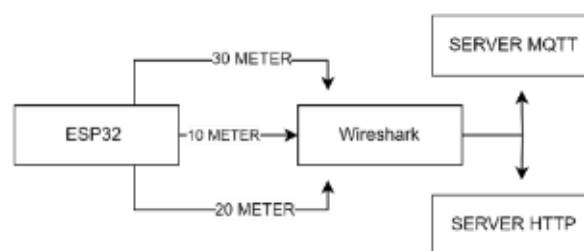
Pengujian ini dilakukan untuk melihat apakah data sensor yang dikirim oleh ESP32 pada server MQTT bisa ditampilkan dan data modul GPS-Neo6M bisa ditampilkan pada web yang telah dirancang . Hasil pengujian dapat dilihat pada gambar 8



Gambar 8 Hasil dari web dashboard

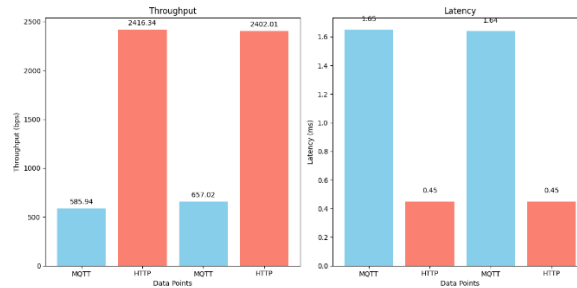
4. Pengujian Perbandingan Protokol MQTT dan HTTP

Mekanisme pengujian akan seperti pada gambar 9 , menggunakan MQTT dan HTTP sebagai protokol pengiriman data. Wireshark akan mengumpulkan informasi tentang throughput (bps), jitter (ms), dan latency (ms) dari data yang dikirim. Pengujian dilakukan dari jarak 20m, 50m, dan 70m dari WiFi ke ESP32, skenario perbedaan jarak pada pengujian dengan asumsi tertentu untuk mencapai peningkatan latency yang lebih besar . Ini memungkinkan evaluasi menyeluruh kinerja kedua protokol pada jarak yang berbeda, yang berdampak pada konektivitas skenario pengujian.



Gambar 9. Skenario

a. Pengujian throughput(bps), Latency(ms) selama 10 menit 20 meter , hasil dari pengujian dapat dilihat pada gambar 10 dan table 1

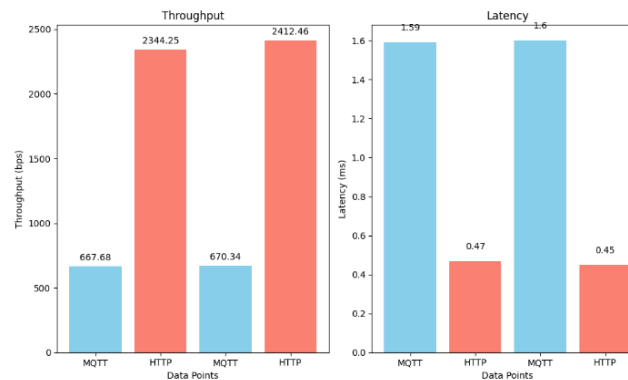


Gambar 10. Hasil pengujian HTTP dan MQTT

Tabel 1. Hasil pengujian HTTP dan MQTT

No	Scenario	Protokol	Throughput (bps)	Latency (ms)
1	10d-20M	MQTT	585.94	1.65
2		HTTP	2416.34	0,45
3		MQTT	657.02	1,64
4		HTTP	2402.01	0,45

b. Pengujian throughput(bps), Latency(ms) selama 15 menit 50 meter, hasil dari pengujian dapat dilihat pada gambar 11 dan table 2



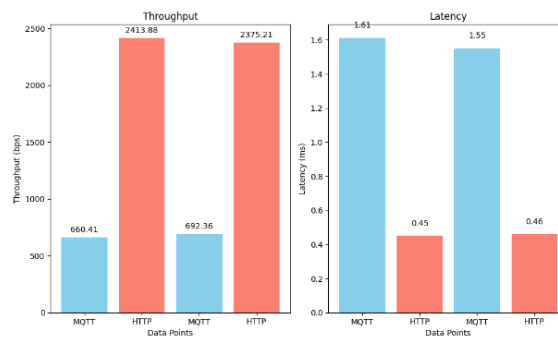
Gambar 11. Hasil pengujian HTTP dan MQTT

Tabel 2. Hasil pengujian HTTP dan MQTT

No	Scenario	Protokol	Throughput (bps)	Latency (ms)
1	15d-50M	MQTT	667.68	1,59
2		HTTP	2344.25	0,47

3	MQTT	670.34	1,60
4	HTTP	2412.46	0,45

c. Pengujian throughput(bps), Latency(ms) selama 10 menit 20 meter , hasil dari pengujian dapat dilihat pada gambar 12 dan table 3



Gambar 12. Hasil pengujian HTTP dan MQTT

Tabel 3. Hasil pengujian HTTP dan MQTT

No	Scenario	Protokol	Throughput (bps)	Latency (ms)
1	20d-50M	MQTT	660.41	1.61
2		HTTP	2413.88	0.45
3		MQTT	692.36	1.55
4		HTTP	2375.21	0.46

5. Analisa hasil Pengujian

Dari paparan dan penjelasan mendapatkan hasil yang terbaik pada implemtasi protokol MQTT pada rancang bangun monitroing sepeda motor ada bebrapa analisa diantaranya sebagai berikut :

- Perangkat keras berhasil berfungsi dengan baik mampu terkoneksi dengan WiFi dan terhubung ke broker MQTT. Dalam pengujian akuisisi sinyal GPS dengan tiga kali percobaan, waktu yang dihasilkan adalah 9.07 detik, 4.73 detik, dan 11.03 detik. Terdapat beberapa perbedaan dalam proses akuisisi sinyal yang bisa dipengaruhi oleh faktor-faktor seperti lokasi berada di dalam ruangan, kondisi cuaca, dan lain sebagainya. Semua hasil pengujian dimulai dari koneksi ke WiFi, koneksi ke broker MQTT, dan pengiriman data koordinat lokasi ke broker. Informasi ini dapat dilihat melalui serial monitor pada Arduino IDE
- Protokol MQTT menggunakan broker EMQX yang diinstal pada VPS berhasil berfungsi dengan baik untuk menerima dan mengirimkan data dari modul GPS Neo6M kepada web client
- Web client yang menggunakan Node.js berhasil berfungsi dengan baik, mampu menampilkan titik koordinat yang diterima dari broker MQTT

- d. Pada hasil pengujian tiga skenario yang berbeda, dengan durasi dan jarak yang beragam, menyoroti perbedaan kinerja antara protokol MQTT dan HTTP. Dalam pengukuran throughput (bps) dan latency (ms), protokol HTTP menunjukkan performa latency yang lebih baik dibandingkan MQTT, dengan nilai latency rata-rata 0.455 ms - 0.46 ms dibandingkan dengan 1.58 ms - 1.645 ms pada MQTT. Namun, nilai throughput rata-rata pada MQTT secara konsisten lebih tinggi daripada HTTP, dengan nilai rata-rata throughput MQTT antara 621.48 bps - 676.385 bps dan HTTP antara 2378.355 bps - 2409.175 bps. Dari hasil ini, terlihat bahwa MQTT mampu beroperasi dengan menggunakan bandwidth jaringan yang lebih kecil dibandingkan HTTP, sementara tetap memberikan kinerja throughput yang unggul, meskipun dengan nilai latency yang sedikit lebih tinggi

VII. SIMPULAN

Analisis dari hasil pengujian menunjukkan bahwa protokol MQTT sangat cocok untuk penggunaan dalam rancang bangun monitoring sepeda motor ini. Meskipun menunjukkan throughput yang lebih rendah dibandingkan HTTP, MQTT memberikan keunggulan dalam penggunaan bandwidth yang lebih efisien, menghasilkan penghematan dalam paket data yang dikirimkan melalui jaringan. Meskipun latency-nya sedikit lebih tinggi, kelebihan dalam efisiensi penggunaan bandwidth dapat menjadi faktor penting dalam mempertimbangkan penggunaan protokol ini untuk aplikasi monitoring sepeda motor

UCAPAN TERIMA KASIH

Peneliti mengucapkan terima kasih sebesar-besarnya kepada orang tua dan keluarga yang selalu memberikan dukungan penuh dalam penelitian ini, Tak lupa juga terima kasih kepada pihak kampus telah memberikan sarana dan fasilitas. Semua kontribusi dan dukungan untuk peneliti sangat berarti bagi kesuksesan penelitian ini. Terima kasih atas segala doa dan support yang telah diberikan.

REFERENSI

References

- [1] Patel and R. Goyena, "UPAYA KEPOLISIAN DALAM MENANGGULANGI TINDAK PIDANA PEMBEGALAN DI WILAYAH KOTA PASURUAN," *J. Chem. Inf. Model.*, vol. 15, no. 2, pp. 9–25, 2019.
- [2] M. Arifin, D. P. D. selengkapnya <https://www.detik.com/jatim/berita/-6030059/3-begal-sadis-di-pasuruan-diringkus-dua-pelaku-ditembak>. Baca artikel detikjatim, "3 Begal Sadis di Pasuruan Diringkus, and D. A. D. S. <https://apps.detik.com/detik/>, "3 Begal Sadis di Pasuruan Diringkus, Dua Pelaku Ditembak." <https://www.detik.com/jatim/berita/d-6030059/3-begal-sadis-di-pasuruan-diringkus-dua-pelaku-ditembak> (accessed Aug. 28, 2023).
- [3] S. Samsugi and W. Wajiran, "IoT: EMERGENCY BUTTON SEBAGAI PENGAMAN UNTUK MENGHINDARI PERAMPASAN SEPEDA MOTOR," *J. Teknoinfo*, vol. 14, no. 2, p. 99, 2020, doi: 10.33365/jti.v14i2.653.
- [4] Y. Pratama, D. N. Ramadan, S. Pd, and T. N. Damayanti, "Perancangan GPS Tracking Untuk Penyewaan Kendaraan Bermotor Design of GPS Tracking on Lending Motor Vehicle," *e-Proceeding Appl. Sci.*, vol. 6, no. 2, pp. 1–15, 2020.
- [5] S. B. Bhaskoro, H. Supriyanto, B. B. Aji, and B. Pamungkas, "Perbandingan Performansi Latency Protokol Komunikasi Http Dan Mqtt Pada Internet of Things," *JTT (Jurnal Teknol. Ter.*, vol. 8, no. 2, p. 82, 2022, doi: 10.31884/jtt.v8i2.309.
- [6] A. Surahman, A. T. Prastowo, and L. A. Aziz, "Rancang Alat Keamanan Sepeda Motor Honda Beat Berbasis Sim Gsm Menggunakan Metode Rancang Bangun," *J. Teknol. dan Sist. Tertanam*, vol. 3, no. 1, 2022, doi: 10.33365/jtst.v3i1.1918.
- [7] M. Saiqul Umam, S. Adi Wibowo, and Y. Agus Pranoto, "Implementasi Protokol Mqtt Pada Aplikasi Smart Garden Berbasis Iot (Internet of Things)," *JATI (Jurnal Mhs. Tek. Inform.*, vol. 7, no. 1, pp. 899–906, 2023, doi: 10.36040/jati.v7i1.6131.
- [8] 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.
- [9] Uray Ristian, "Perancangan Arsitektur Node Nirkabel dalam Efisiensi Bandwidth Smart Greenhouse Berbasis," vol. 9, no. 2, pp. 218–225, 2023.
- [10] I. Kurniawan, Humaira, and F. Rozi, "REST API Menggunakan NodeJS pada Aplikasi Transaksi Jasa Elektronik Berbasis Android," *JITSI J. Ilm. Teknol. Sist. Inf.*, vol. 1, no. 4, pp. 127–132, 2020, doi: 10.30630/jitsi.1.4.18.

Conflict of Interest Statement:

The author declares that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.