

Pengendali Raspberry Pi Jarak Jauh Menggunakan Protokol Mqtt Berbasis Web

1st Mitsal Fabian Nadiyah
Fakultas Ilmu Terapan
Universitas Telkom
Bandung, Indonesia

mitsalfabian@student.telkomuniversity.ac.id

2nd Prajna Deshanta Ibnugraha
Fakultas Ilmu Terapan
Universitas Telkom
Bandung, Indonesia

prajna@telkomuniversity.ac.id

3rd Aris Pujud Kurniawan
Fakultas Ilmu Terapan
Universitas Telkom
Bandung, Indonesia

arispujudku@telkomuniversity.ac.id

Abstrak—Fasilitas pelayanan dalam transportasi umum seperti CCTV dan GPS perlu diterapkan agar rasa aman dan nyaman dapat terpenuhi sehingga timbul kepercayaan masyarakat dalam menggunakan transportasi umum. Penerapan solusi teknologi ini memunculkan beberapa kendala seperti kesulitan dalam konfigurasi perangkat yang terpasang pada kendaraan, karena lokasi kendaraan yang selalu berpindah atau berada di lokasi yang jauh. Penelitian ini bertujuan untuk membuat sistem manajemen berbasis web yang dapat mengelola Raspberry Pi sebagai unit kontrol perangkat penunjang keamanan dalam kendaraan. Message Queue Telemetry Transport Protocol (MQTT) dapat diterapkan dalam sistem untuk mendukung komunikasi antara website dengan Raspberry Pi. Hasil yang di dapat adalah sebuah sistem berbasis website yang dapat melakukan konfigurasi data berupa nilai interval waktu, GPIO pin activation, dan manual capture. Pengujian juga dilakukan dengan mengukur tingkat kesesuaian dan latency dari setiap transmisi data. Hasil pengukuran kesesuaian menunjukkan bahwa data dapat diterima dengan benar dan sesuai oleh Raspberry Pi, dengan tingkat kesesuaian mencapai 100%. Sedangkan hasil pengukuran latency menunjukkan data dapat diterima oleh Raspberry Pi dengan rata-rata latency sebesar 443,93 ms.

Kata Kunci— mqtt, raspberry pi, sistem manajemen transportasi, website, latency.

I. PENDAHULUAN

Peran penting transportasi pada segala aspek kehidupan masyarakat Indonesia membuat kebutuhan akan transportasi semakin meningkat. Badan Pusat Statistik (BPS) mencatat perkembangan jumlah kendaraan menurut jenis, dimana penggunaan transportasi pribadi seperti motor dan mobil menjadi jenis transportasi yang paling mendominasi dan terus mengalami peningkatan sepanjang tahun 2018 sampai tahun 2020. Data menyebutkan ada lebih dari 130 juta unit kendaraan pribadi dari total 136 juta unit kendaraan yang ada [1]. Hal ini tentunya berdampak pada angka kemacetan yang semakin meningkat, salah satunya Kota Jakarta. Data dari Dinas Perhubungan Provinsi DKI Jakarta menyebutkan bahwa pada tahun 2021 Jakarta menduduki peringkat 46 dari 404 kota yang diukur di seluruh dunia dan memiliki indeks kemacetan mencapai angka 34% [2]. Pengalihan penggunaan transportasi pribadi ke transportasi umum merupakan salah satu upaya dari pemerintah dan masyarakat

dalam mengurangi tingkat kemacetan di Indonesia. Badan Pusat Statistik mencatat jumlah penumpang angkutan umum secara kumulatif meningkat 13,91% pada periode 24 Desember 2021 hingga 2 Januari 2022 [3].

Peningkatan penggunaan transportasi umum khususnya bus juga harus diimbangi dengan fasilitas pelayanan yang baik. Tersedianya beberapa fasilitas seperti layanan 2 call center, penyediaan smoking area di dalam bus, pemeliharaan terhadap AC dan kursi secara berkala dapat memberikan pengalaman perjalanan yang nyaman kepada para penumpang sekaligus dapat mempertahankan loyalitas masyarakat dalam menggunakan bus sebagai alat transportasi utama [4]. Tak hanya itu, keamanan juga merupakan salah satu faktor penting yang dapat menunjang kenyamanan dalam perjalanan. Salah satu permasalahan besar yang selalu dialami dan sulit dihindari di seluruh dunia, di negara maju maupun negara berkembang adalah masalah kriminalitas. Indonesia menjadi salah satu negara dengan tingkat kriminalitas yang tinggi di dunia dengan angka kriminalitas mencapai 38,45% pada pertengahan 2020 lalu [5].

Beberapa solusi teknologi yang telah diterapkan untuk mengurangi tindak kriminalitas pada transportasi umum khususnya bus adalah pemasangan CCTV (Closed Circuit Television). CCTV ini digunakan untuk memantau keadaan yang terjadi di dalam bus [6]. Solusi lain juga ditawarkan pemerintah berupa regulasi yang dikeluarkan untuk mendukung peningkatan keamanan dalam bus. Solusi tersebut ialah pemasangan perangkat Global Positioning System (GPS) Tracker pada armada bus untuk memantau posisi sekaligus meningkatkan efisiensi dan efektivitas dalam hal pelayanan [7]. Adapun kendala yang muncul dari solusi teknologi yang telah disebutkan sebelumnya, salah satu kendala yang kerap terjadi adalah kesulitan dalam konfigurasi perangkat yang telah terpasang pada kendaraan karena lokasi kendaraan yang selalu berpindah-pindah atau karena kendaraan berada di lokasi yang sangat jauh bahkan beda pulau.

Kesulitan dan kendala yang telah disebutkan sebelumnya dapat diatasi dengan menerapkan *Raspberry Pi* sebagai unit kontrol alat penunjang keamanan dalam kendaraan. *Raspberry Pi* dapat digunakan sebagai perangkat yang dapat mengontrol CCTV maupun GPS yang terpasang dalam

kendaraan. Selain itu, diperlukan juga suatu instrumen atau protokol sebagai perantara antara sistem pengendali yang dibangun dengan perangkat *Raspberry Pi*, agar *Raspberry Pi* dapat dikonfigurasi dan dikendalikan dari jarak jauh. Adapun protokol yang dimaksud adalah protokol MQTT. *Message Queue Telemetry Transport* atau MQTT merupakan protokol komunikasi *machine to machine* (M2M) yang sangat ringan dan efisien, karena protokol ini dapat mengirim pesan dengan header yang berukuran sangat kecil yaitu 2 bytes sehingga dapat mengoptimalkan penggunaan *bandwidth* jaringan [8]. Protokol MQTT ini digunakan sebagai perantara antara sistem pengendali berbasis web dengan *Raspberry Pi*. Penelitian ini bertujuan untuk membuat suatu sistem berbasis *website* yang dapat melakukan konfigurasi *Raspberry Pi* berupa nilai interval waktu, GPIO pin *activation*, dan *manual capture*. Selain itu, penelitian ini bertujuan untuk melakukan pengukuran akurasi kesesuaian data dan *latency* yang dibutuhkan pada pengiriman data dari *website* ke *Raspberry Pi* melalui protokol MQTT.

II. KAJIAN TEORI

A. Tinjauan Pustaka

Berdasarkan penelitian yang dilakukan kumbhar mengenai pemecahan dari permasalahan sistem transportasi yaitu dengan sistem pelacakan bus secara *real time*. Sistem ini digunakan untuk menciptakan manajemen transportasi secara efisien. Acuan dari sistem ini dilakukan berdasarkan teknologi *Global Positioning System* (GPS), Google peta dan GPRS. Hasil dari sistem ini berupa data berbasis web aplikasi yang mampu memberikan informasi lokasi bus secara *real time* di Google peta untuk pengguna umum. Teknologi ini memiliki kekurangan karena tidak adanya sensor gerak yang dapat menghitung kecepatan bus [9].

Pada jurnal penelitian selanjutnya menunjukkan bahwa MQTT menjadi salah satu pilihan utama dalam membangun sistem berbasis *Internet of Thing* (IoT) yang membutuhkan adanya distribusi data dari satu perangkat ke perangkat lain. Hasil penelitian sebelumnya menunjukkan bahwa protokol MQTT dapat bekerja lebih baik dibandingkan protokol transfer data lainnya. Protokol MQTT memiliki waktu *delay* dan *response time* yang kecil serta dapat bekerja pada sumber daya dan *bandwidth* yang minim, sehingga tepat dalam implementasi sistem berbasis IoT [10]. Suatu penelitian mengimplementasikan MQTT dalam mengembangkan aplikasi *smart home*. MQTT digunakan sebagai perantara data antara sensor dengan *Raspberry Pi*. Pengimplementasian MQTT pada penelitian ini hanya berfokus pada aplikasi *smart home* [11].

Penggunaan MQTT dalam bidang manajemen transportasi telah dilakukan pada penelitian yang berjudul "Pemodelan Sistem Pemantauan Posisi Kereta Api Berbasis RFID Menggunakan Protokol *Message Queue Telemetry Transport* (MQTT)". Dalam sistem ini sebuah RFID reader dipasang pada kereta api yang digunakan untuk membaca RFID tag yang telah terpasang pada jalur kereta sebagai representasi lokasi kereta api. Dalam penelitiannya penulis menjelaskan bahwa protokol MQTT dimanfaatkan sebagai transmisi data secara nirkabel. MQTT digunakan untuk 7 pendistribusian data dari RFID reader menuju *website*

pemantauan yang akan menampilkan informasi berupa lokasi suatu rangkaian kereta api secara *real time* [12].

Penelitian selanjutnya mengenai framework laravel telah dijabarkan oleh Mediana pada pengaplikasian rancang bangun aplikasi *helpdesk* berbasis *website*. Proses pengembangan sistem menggunakan *framework laravel* sehingga tampilan akan terlihat lebih rapi daripada menggunakan *PHP native*. Perlu terdapat penelitian dan terobosan terbaru dalam pengaplikasian *framework laravel* di bidang web yang menyajikan data *real time* pelacakan kendaraan [13].

B. Dasar Teori

Message Queue Telemetry Transport atau yang biasa disebut sebagai MQTT merupakan suatu protokol pesan yang dirancang pada tahun 1999, oleh Dr. Andy dari Stanford, Clark dari IBM dan Arlen Nipper. MQTT ini adalah salah satu protokol komunikasi sederhana dan ringan yang biasa diterapkan pada konsep *Internet of Thing* (IoT). Protokol ini dirancang untuk komunikasi antar perangkat dengan sumber daya dan *bandwidth* jaringan yang terbatas. Selain itu, MQTT relatif mudah diimplementasikan dalam suatu sistem dan dapat menangani ribuan sampai jutaan komunikasi antar *client* hanya dengan satu *server*. Protokol MQTT menerapkan konsep *publish/subscribe* sehingga mampu melakukan komunikasi dua arah, dimana setiap *client* dapat bertindak sebagai *publisher* (pengirim pesan) sekaligus sebagai *subscriber* (penerima pesan). Pada protokol MQTT, *broker* berperan sebagai pusat komunikasi yang bertanggung jawab atas proses pendistribusian pesan antar *client*. *Broker* bertugas mengatur, mengelola, dan melanjutkan pesan dari *publisher* ke *subscriber* berdasarkan topik yang sesuai. Pesan akan langsung diteruskan oleh *broker* apabila terdapat kesesuaian topik antara *publisher* dan *subscriber* [14]. Pada penelitian ini, konsep *publish/subscribe* yang dimiliki MQTT akan digunakan dalam sistem pengendali *Raspberry Pi* berbasis web yang digunakan untuk komunikasi antara *website* dengan *Raspberry Pi*. *Website* akan dijadikan *publisher* sebagai pengirim pesan berupa nilai interval waktu, GPIO pin *activation* dan *manual capture*, sedangkan mikrokomputer *Raspberry Pi* yang telah diintegrasikan dengan perangkat penunjang keamanan akan bertindak sebagai *subscriber*.

Raspberry Pi merupakan sebuah komputer dengan ukuran kecil atau biasa disebut dengan mikrokomputer. *Raspberry Pi* memiliki ukuran sebesar kartu kredit, namun dengan ukuran sekecil itu, mikrokomputer ini masih memiliki komponen layaknya komputer pada umumnya namun dengan ukuran yang lebih minimalis [15]. Pada penelitian ini, *Raspberry Pi* akan berfungsi sebagai unit kontrol alat penunjang keamanan dalam kendaraan dan juga bertindak sebagai *subscriber* yang akan menerima data dari *website*.

JSON atau *JavaScript Object Notation* merupakan salah satu format pertukaran data yang bentuknya berupa pasangan *key* dan *value* atau biasa disebut dengan *light-weight key-value*. JSON sendiri memiliki format berbentuk teks dan struktur data yang sederhana sehingga dapat dibaca dengan mudah oleh manusia [16]. Format JSON ini akan digunakan dalam proses transmisi data antara *website* dan *Raspberry Pi*.

Framework *Laravel* merupakan salah satu *framework PHP* yang bersifat *open source* yang dirancang untuk membuat pengembangan aplikasi berbasis web dengan lebih mudah dan cepat melalui fitur bawaan yang dimiliki [17]. *Framework* ini lebih fokus pada bagian *end-user* sehingga sangat baik dalam hal penulisan maupun tampilan karena kejelasan dan kesederhanaannya, dengan menggunakan *Laravel*, *website* yang dibuat menjadi lebih mudah dikembangkan dan lebih mudah mengatur data dari sumber daya yang ada serta pengembangan *website* dengan waktu yang relatif cepat [13]. Pada penelitian ini, *Framework Laravel* akan digunakan untuk mempermudah dan mempercepat pengembangan aplikasi berbasis *website*.

Latency merupakan jumlah waktu yang dibutuhkan untuk suatu paket, data ataupun pesan dikirim dari komputer pengirim ke komputer tujuan. Besarnya angka *Latency* juga dapat dipengaruhi oleh jarak, media pengiriman atau waktu proses yang lama [18], [19]. Berikut merupakan standarisasi nilai *latency* menurut ITU-T (*International Telecommunication Union-Telecommunication*).

TABEL 1
(STANDARISASI LATENCY VERSI ITU-T)

Category	Latency
Baik	< 150 ms
Cukup	150 s.d. 400 ms
Buruk	> 400 ms

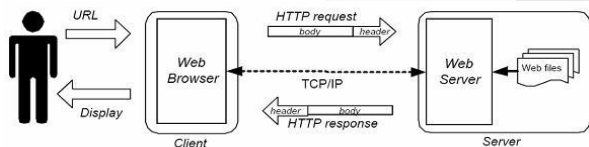
Berikut merupakan rumus *latency* yang ditunjukkan pada persamaan 1.

$$Latency = waktu\ data\ diterima - waktu\ data\ dikirim \quad (1)$$

III. METODE

A. Gambaran Sistem Saat Ini

Sistem berbasis *Internet of Thing (IoT)* pada umumnya masih menggunakan *HTTP* dalam proses transmisi atau komunikasi data. *HTTP* merupakan protokol komunikasi data yang paling populer dan penggunaannya yang sangat luas dalam komunikasi *website*.

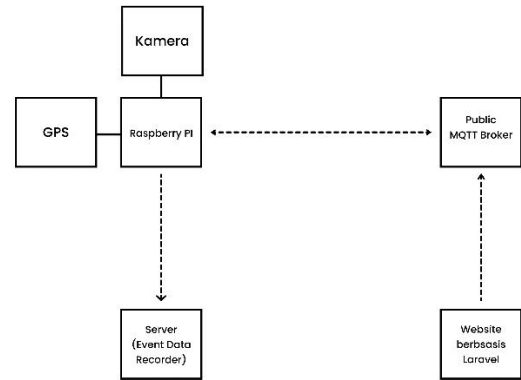


GAMBAR 1
(ALUR KERJA HTTP)

Berdasarkan alur kerja *HTTP* yang ditunjukkan pada Gambar 1 *HTTP* bekerja dengan konsep *request* dan *response*. *Client* yang ditunjukkan sebagai *web browser* akan mengirim request atau permintaan ke *web server*, selanjutnya *web server* akan memproses *request* tersebut. Tahap terakhir adalah *web server* melakukan *response* atau mengirimkan jawaban dengan menampilkan apa yang diinginkan *client* melalui *web browser*. Suatu penelitian membandingkan performansi antara *MQTT* dan *HTTP* berdasarkan parameter *throughput*, *delay*, dan *packet loss*. Data menunjukkan bahwa performa *throughput* yang dimiliki *MQTT* jauh lebih tinggi. Sedangkan untuk

parameter *delay* dan *packet loss* keduanya tidak jauh berbeda, namun *MQTT* tetap lebih unggul [26].

B. Perancangan Sistem



GAMBAR 2
(BLOK DIAGRAM PENGENDALI RASPBERRY PI)

Gambar 2 blok diagram dari sistem pengendali *Raspberry Pi* jarak jauh menggunakan protokol *MQTT* berbasis *website*. Ada beberapa bagian utama dalam sistem ini, yaitu *Publisher*, *Broker*, dan *Subscriber* serta *server* sebagai tempat penyimpanan utama dari data informasi kendaraan. *Website* berbasis *Laravel* dalam sistem ini bertindak sebagai *Publisher* (pengirim pesan). Topik yang digunakan adalah *serial id* yang terdapat pada perangkat *Raspberry Pi*. Sehingga satu *serial id* merepresentasikan satu perangkat *Raspberry Pi*. Pesan dan topik yang dikirim dari *publisher* selanjutnya masuk ke *MQTT broker*. Pesan tersebut selanjutnya akan dipilah dan dilanjutkan ke *subscriber* apabila ada kesesuaian topik.

Raspberry Pi dalam sistem ini bertindak sebagai *Subscriber* (penerima pesan). Di dalam *Raspberry Pi* telah terpasang *MQTT client* yang diprogram untuk *subscribe* pada topik *unique id* sesuai *Raspberry Pi* yang terkait. *Subscriber (Raspberry Pi)* selanjutnya akan menerima pesan atau data sesuai topik yang di *subscribe* dari *broker*. Pesan atau data diterima *Raspberry Pi* berupa nilai interval waktu selanjutnya akan dimasukkan ke dalam program. Program ini digunakan untuk mengirim data informasi kendaraan menuju *server* utama. Pesan atau data yang telah diterima *Raspberry Pi* ini selanjutnya digunakan untuk mengubah interval waktu pengiriman data yang telah digunakan sebelumnya, *Raspberry Pi* juga dapat menerima data maupun sinyal berupa *GPIO trigger activation* atau *GPIO pin activation* dan *manual capture* yang digunakan untuk mengontrol kamera maupun *GPS* pada kendaraan.

C. Perancangan Komunikasi MQTT

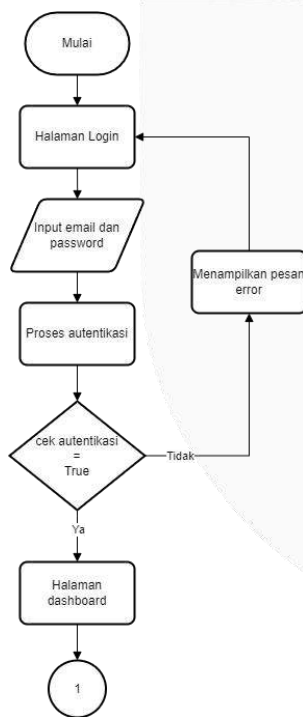
Pada rancangan sistem yang diusulkan, protokol *MQTT* digunakan sebagai media pertukaran data antara aplikasi dengan perangkat *Raspberry Pi*. Pada *MQTT* sistem kerja pengiriman datanya menggunakan konsep *Publish* dan *Subscribe* serta menggunakan *Topic* dalam setiap pengiriman data. *Topic* ini menjadi acuan atau *header* dari tiap data yang dikirim. Setiap data yang dikirim ditempatkan pada *Topic* nya masing-masing yang sesuai. Berikut merupakan struktur *Topic* yang digunakan pada sistem yang diusulkan. Berikut merupakan contoh pembagian topik dan pesan yang dijelaskan pada Tabel 2.

TABEL 2
(CONTOH PEMBAGIAN TOPIK MQTT)

Serial ID <i>Raspberry Pi</i>	Topic	Pesan
10000000f6d2d721	device/10000000f6d2d721	{interval:3, GPIO: 1} dan {capture: 1} atau {interval:3}
100000007g8bc2zz	device/100000007g8bc2zz	{interval:4, GPIO: 0} dan {capture: 1} atau {interval:3}
100000008c8bc2aa	device/100000008c8bc2aa	{interval:5, GPIO: 1} dan {capture: 1} atau {interval:3}

Pada Tabel 2 dapat dilihat bahwa struktur *Topic* yang digunakan adalah *device/{serial Id Raspberry Pi}*. Selain itu, pesan yang dikirimkan pada *Raspberry Pi* dirancang dalam format JSON, hal ini dilakukan agar data yang dikirim lebih mudah dibaca oleh penerima, karena struktur datanya yang jelas dan sederhana.

D. Alur Diagram Login

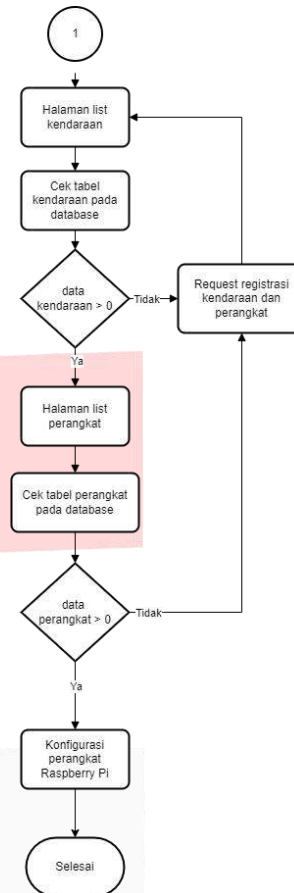


GAMBAR 3
(ALUR DIAGRAM LOGIN)

Gambar 3 menjelaskan mengenai proses login pada aplikasi. Proses dimulai dengan mengakses *website* atau mengakses halaman *login website*. Selanjutnya adalah memasukkan *e-mail* dan *password* yang sesuai. Apabila *e-mail* dan *password* yang dimasukkan benar, maka akan diarahkan ke halaman *dashboard*. Sebaliknya, apabila *e-mail* dan *password* yang dimasukkan salah maka *website*

akan menampilkan pesan kesalahan dan mengarahkan kembali ke halaman *login*.

E. Alur Diagram Aplikasi

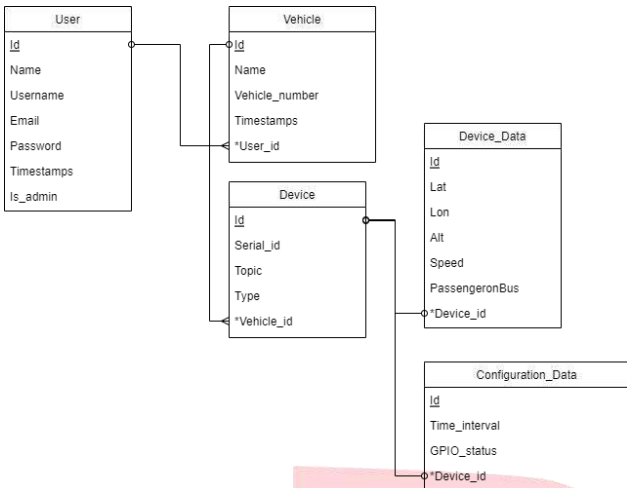


GAMBAR 4
(ALUR DIAGRAM APLIKASI)

Gambar 4 menjelaskan mengenai proses *client* dalam mengkonfigurasi perangkatnya. Proses dimulai dengan membuka halaman *Vehicle List*, apabila sudah ada kendaraan yang terdaftar, maka *client* dapat mengakses halaman *Device List*. Apabila belum ada kendaraan yang terdaftar, maka *client* dapat mengirimkan pesan permintaan kepada admin untuk mendaftarkan kendaraan dan perangkat yang diinginkan. Selanjutnya apabila sudah ada kendaraan dan perangkat yang terdaftar, maka *client* dapat langsung melakukan konfigurasi perangkat *Raspberry Pi*.

F. Perancangan Tabel Basis Data.

Pada suatu sistem atau *website* yang dibangun dibutuhkan *database* yang digunakan sebagai tempat penyimpanan serta pengelolaan data. *Database Management System (DBMS)* yang digunakan pada sistem adalah *MySQL*. Oleh karena itu, diperlukan rancangan tabel basis data untuk memudahkan pengertian dalam hal struktur data, sehingga informasi yang dihasilkan dapat terpenuhi dengan baik.

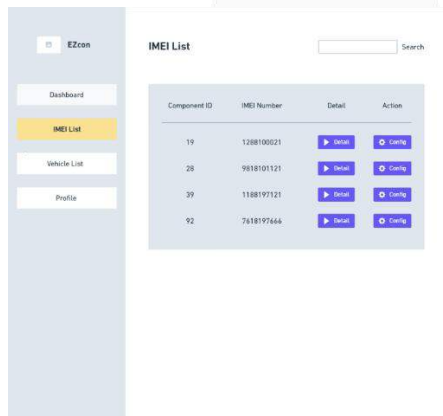


GAMBAR 5 (TABEL BASIS DATA)

Pada sistem yang dibuat dibutuhkan lima entitas, yaitu entitas *User*, *Device*, *Vehicle*, *Device data*, dan *Configuration data* yang setiap entitasnya memiliki atributnya masing-masing. Pada Gambar 5 menunjukkan entitas *User* dan *Vehicle* memiliki kardinalitas *one to many* (1 to N), selanjutnya, entitas *Vehicle* dan *Device* juga saling berelasi dengan kardinalitas *one to many*. Selain itu, entitas *Device* juga memiliki relasi dengan entitas *Device data* dan *Configuration data* dengan hubungan kardinalitas *one to one* (1 to 1)

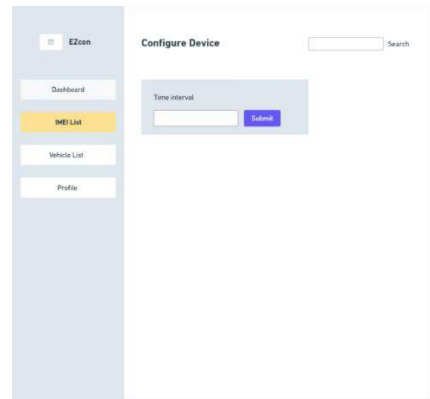
G. Perancangan Antar Muka Aplikasi

Perancangan antar muka atau *interface* aplikasi perlu dilakukan untuk memudahkan *user* dalam menggunakan dan mengoperasikan *website* dengan mengeluarkan sedikit usaha untuk menyelesaikan tugasnya serta dapat menyajikan tampilan yang menarik.



GAMBAR 6 (DESAIN MOCKUP HALAMAN LIST PERANGKAT)

Gambar 5 menunjukkan rancangan antar muka dari halaman list perangkat. Halaman ini menunjukkan daftar perangkat *Raspberry Pi* yang dimiliki oleh *client*.

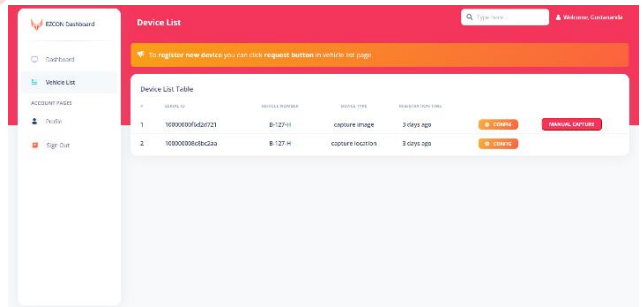


GAMBAR 7 (DESAIN MOCKUP HALAMAN KONFIGURASI PERANGKAT)

Gambar 6 menunjukkan rancangan antar muka dari halaman konfigurasi perangkat. Halaman ini merupakan tempat *client* melakukan konfigurasi terhadap perangkat *Raspberry Pi* yang dimiliki.

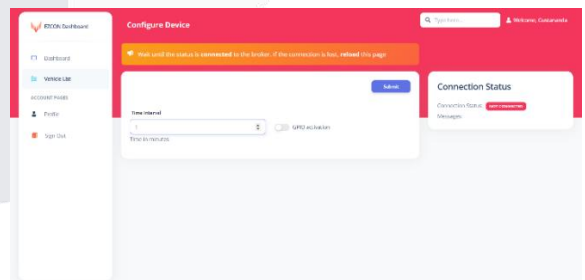
IV. HASIL DAN PEMBAHASAN

A. Implementasi Antar Muka Aplikasi



GAMBAR 8 (ANTAR MUKA HALAMAN DEVICE LIST)

Gambar 8 merupakan tampilan antar muka dari halaman *Device List*. halaman ini digunakan untuk melihat daftar perangkat *Raspberry Pi* yang dimiliki. Selain itu, *client* juga dapat melakukan konfigurasi pada perangkat tertentu dengan menekan *button config*.



GAMBAR 9 (ANTAR MUKA HALAMAN CONFIGURE DEVICE)

Gambar 9 merupakan tampilan antar muka dari halaman *configure device*. Halaman digunakan *client* untuk melakukan konfigurasi terhadap perangkat yang dimiliki.

B. Implementasi MQTT pada Aplikasi

Protokol MQTT (*Message Queue Telemetry Transport*) digunakan pada sistem sebagai penghubung antara *website*

dengan *Raspberry Pi*. Berikut merupakan implementasi MQTT pada aplikasi.

```

$mqtt->publish(
// topic
// $topic,
// payload
json_encode($message),
// qos
0,
// retain
true
);
    
```

GAMBAR 10 (IMPLEMENTASI MQTT PADA APLIKASI)

Gambar 10 menunjukkan fungsi pengiriman atau *Publish* data melalui protokol MQTT, dimana pada fungsi *publish* terdapat 4 parameter yang perlu diisi. Parameter pertama adalah *topic*, *topic* disini berisi kombinasi dari karakter *device* dan *serial id* dari *Raspberry Pi*. Parameter selanjutnya adalah *payload* yang berisi pesan atau data yang akan dikirimkan pada *Subscriber* dengan data yang dikirimkan berbentuk JSON. Selanjutnya adalah QoS atau *Quality of Service* yang menggambarkan keandalan suatu pesan yang dikirim. Parameter terakhir adalah *retain* yang berfungsi untuk menentukan apakah suatu pesan yang dikirim terakhir akan dipertahankan atau tidak.

C. Pengujian Fungsionalitas Aplikasi

Dari hasil pengujian yang telah dilakukan dengan metode *blacbox* didapatkan hasil bahwa pengujian terhadap fungsi atau fitur sistem meliputi fungsi *login*, *register account*, penambahan, pengubahan, dan penghapusan data kendaraan dan data perangkat *Raspberry Pi* berhasil dilakukan dengan hasil uji sesuai yang diharapkan. Selain itu, fungsi untuk mengubah data akun diantaranya *e-mail* dan *password* untuk setiap *user* berhasil dilakukan dengan hasil uji sesuai yang diharapkan. Selanjutnya, terdapat fitur *request* yang digunakan *client* untuk meminta admin untuk mendaftarkan kendaraan dan perangkat *Raspberry Pi* ke dalam sistem berhasil dilakukan dengan hasil uji sesuai yang diharapkan. Berdasarkan hasil pengujian yang telah diuraikan, dapat disimpulkan bahwa seluruh fungsi atau fitur pada sistem dapat digunakan sesuai dengan yang diharapkan dengan tingkat keberhasilan mencapai 100%.

D. Pengujian Kesesuaian Pengiriman Data Melalui Protokol MQTT

Pengujian dilakukan untuk memastikan kesesuaian data antara data yang dikirim dari *website* dengan data yang diterima pada *Raspberry Pi*. Pengujian proses pengiriman data dilakukan dengan hanya menggunakan satu topik atau dengan kata lain pengujian hanya dilakukan pada satu perangkat *Raspberry Pi*. Data yang dikirim berupa nilai interval serta GPIO pin *activation* dan *manual capture*.

TABEL 3 (PENGUJIAN KESESUIAN DATA)

No.	Topik	Data dikirim	Data diterima	Keterangan
1.		{interval : 3, GPIO : 0}	{interval : 3, GPIO : 0}	Sesuai
2.		{interval : 4, GPIO : 0}	{interval : 4, GPIO : 0}	Sesuai
3.		{interval : 7, GPIO : 0}	{interval : 7, GPIO : 0}	Sesuai
4.		{interval : 2, GPIO : 0}	{interval : 2, GPIO : 0}	Sesuai
5.		{interval : 6, GPIO : 0}	{interval : 6, GPIO : 0}	Sesuai

6.	device/10000000f6d2d783	{interval : 2, GPIO : 0}	{interval : 2, GPIO : 0}	Sesuai
7.		{interval : 7, GPIO : 0}	{interval : 7, GPIO : 0}	Sesuai
8.		{interval : 2, GPIO : 0}	{interval : 2, GPIO : 0}	Sesuai
9.		{interval : 6, GPIO : 0}	{interval : 6, GPIO : 0}	Sesuai
10.		{interval : 8, GPIO : 0}	{interval : 8, GPIO : 0}	Sesuai
11.		{interval : 10, GPIO : 0}	{interval : 10, GPIO : 0}	Sesuai
12.		{interval : 2, GPIO : 0}	{interval : 2, GPIO : 0}	Sesuai
13.		{capture : 1}	{capture : 1}	Sesuai
14.		{interval : 6, GPIO : 0}	{interval : 6, GPIO : 0}	Sesuai
15.		{interval : 3, GPIO : 0}	{interval : 3, GPIO : 0}	Sesuai
16.		{interval : 4, GPIO : 0}	{interval : 4, GPIO : 0}	Sesuai
17.		{interval : 5, GPIO : 0}	{interval : 5, GPIO : 0}	Sesuai
18.		{interval : 3, GPIO : 0}	{interval : 3, GPIO : 0}	Sesuai
19.		{interval : 2, GPIO : 0}	{interval : 2, GPIO : 0}	Sesuai
20.		{interval : 5, GPIO : 0}	{interval : 5, GPIO : 0}	Sesuai
21.		{interval : 4, GPIO : 0}	{interval : 4, GPIO : 0}	Sesuai
22.		{interval : 8, GPIO : 0}	{interval : 8, GPIO : 0}	Sesuai
23.		{capture : 1}	{capture : 1}	Sesuai
24.		{interval : 9, GPIO : 0}	{interval : 9, GPIO : 0}	Sesuai
25.		{interval : 8, GPIO : 0}	{interval : 8, GPIO : 0}	Sesuai
26.		{interval : 6, GPIO : 0}	{interval : 6, GPIO : 0}	Sesuai
27.		{interval : 3, GPIO : 0}	{interval : 3, GPIO : 0}	Sesuai
28.		{interval : 2, GPIO : 0}	{interval : 2, GPIO : 0}	Sesuai
29.		{interval : 3, GPIO : 0}	{interval : 3, GPIO : 0}	Sesuai
30.		{interval : 6, GPIO : 0}	{interval : 6, GPIO : 0}	Sesuai

Dari hasil pengujian yang telah dilakukan sebanyak tiga puluh kali dapat disimpulkan bahwa data konfigurasi yang dikirim dari *website* melalui protokol MQTT dapat diterima dengan benar dan sesuai oleh perangkat *Raspberry Pi*, dengan tingkat kesesuaian mencapai 100%.

E. Pengujian *Latency* Pengiriman Data Melalui Protokol MQTT

Pengujian dilakukan untuk mengetahui waktu yang dibutuhkan untuk suatu data dikirim dari *website* sampai ke perangkat *Raspberry Pi* tujuan. Pengujian Pengiriman data dilakukan dengan menggunakan satu perangkat *Raspberry Pi*. Data yang dikirim berupa nilai interval serta GPIO pin *activation* dan *manual capture* yang bertipe data *boolean*. Aplikasi dan *Raspberry Pi* terkoneksi pada *router Wi-Fi* yang sama dengan *bandwith* 20 Mb/s.

No	Waktu kirim	Waktu terima	Latency (ms)
1	9:46:40.681	9:46:41.155	474
2	9:47:00.253	9:47:00.801	548
3	9:47:27.219	9:47:27.659	440
4	9:47:40.349	9:47:40.743	394
5	9:47:50.937	9:47:51.348	410
6	9:48:02.651	9:48:03.117	465
7	9:48:14.735	9:48:15.193	458
8	9:48:28.819	9:48:29.218	398
9	9:48:42.958	9:48:43.435	476
10	9:49:14.814	9:49:15.260	445
11	9:49:26.535	9:49:26.990	454
12	9:49:37.895	9:49:38.433	537
13	9:49:50.990	9:49:51.397	406
14	9:50:02.849	9:50:03.255	405
15	9:50:13.467	9:50:13.896	429
16	9:50:24.628	9:50:25.050	422
17	9:50:35.603	9:50:36.140	537
18	9:50:47.884	9:50:48.413	529
19	9:50:59.896	9:51:00.415	518
20	9:51:13.897	9:51:14.374	476
21	9:51:25.281	9:51:25.748	467
22	9:51:36.015	9:51:36.487	472
23	9:51:46.285	9:51:46.677	391
24	9:51:57.426	9:51:57.867	440
25	9:52:08.983	9:52:09.429	445
26	9:52:21.200	9:52:21.329	128
27	9:52:32.025	9:52:32.472	447
28	9:52:42.888	9:52:43.311	422
29	9:52:54.122	9:52:54.559	436
30	9:53:06.597	9:53:07.000	449
Rata-rata latency (ms)			443.93

Berdasarkan hasil pengujian pengiriman data melalui protokol MQTT yang telah dilakukan sebanyak tiga puluh kali, didapatkan rata-rata *latency* sebesar 443,93 ms.

V. KESIMPULAN

Berdasarkan pengujian yang telah dilakukan, sistem berbasis *website* telah dapat melakukan konfigurasi nilai interval yang akan digunakan *Raspberry Pi* untuk mengatur interval waktu pengiriman data ke *server*. Selanjutnya sistem telah dapat mengatur GPIO pin *activation* yang akan digunakan *Raspberry Pi* untuk mengaktifkan pin GPIO untuk memanfaatkan sinyal dari GPS yang terpasang. Selain itu, sistem juga telah dapat melakukan *manual capture* yang digunakan untuk memberi sinyal pada *Raspberry Pi* untuk melakukan *capture* gambar atau *event* yang terjadi di dalam kendaraan. Keseluruhan fungsi dan fitur aplikasi dapat digunakan sesuai dengan hasil yang diharapkan dengan tingkat keberhasilan sebesar 100%. Selain itu, pengujian kesesuaian dan *latency* terhadap pengiriman data melalui

protokol MQTT telah dilakukan dengan hasil kesesuaian data mencapai 100% dan rata-rata *latency* yang dihasilkan sebesar 443.93 ms.

REFERENSI

- [1] B. P. Statistik, "Perkembangan Jumlah Kendaraan Bermotor Menurut Jenis (Unit), 2017-2019." <https://www.bps.go.id/indicator/17/57/1/jumlah-kendaraan-bermotor.html> (accessed Jun. 29, 2021).
- [2] Bintang Pradewo, "2021 Indeks Kemacetan Jakarta 34 Persen, Duduki Peringkat 46 di Dunia." <https://www.jawapos.com/jabodetabek/11/02/2022/2021-indeks-kemacetan-jakarta-34-persen-duduki-peringkat-46-di-dunia/>.
- [3] R. Yati, "Nataru 2021, Penumpang Kendaraan Umum dan Pribadi Naik 13,91 Persen." <https://ekonomi.bisnis.com/read/20220103/98/1484933/nataru-2021-penumpang-kendaraan-umum-dan-pribadi-naik-1391-persen>.
- [4] S. Aminah, "Transportasi Publik dan Aksesibilitas Masyarakat Perkotaan," *Tek. Sipil Untan*, vol. 12, no. DESEMBER, pp. 175–176, 2012, [Online]. Available: [file:///C:/Users/Nur Ali Rahmatullah/Downloads/1435-4550-1-PB.pdf](file:///C:/Users/Nur%20Ali%20Rahmatullah/Downloads/1435-4550-1-PB.pdf).
- [5] Y. P. Wijayaatmaja, "Angka Kriminalitas Naik 38,45%," 2020. <https://mediaindonesia.com/megapolitan/321027/polri-sebut-angka-kriminalitas-naik-3845>.
- [6] J. Sembiring, "Jurnal Penelitian Transportasi Darat," *J. Chem. Inf. Model.*, vol. 53, no. 9, pp. 1689–1699, 2020.
- [7] E. Irawan, "Petunjuk Teknis Alat Pemantau Pergerakan Kendaraan secara Elektronik pada Angkutan Orang dengan Kendaraan Bermotor Umum."
- [8] E. Fitriah, D. Permatasari, A. G. Putrada, and M. Abdurohman, "Analisis Perbandingan Performansi MQTT dan HTTP pada Platform IoT Node-," vol. 6, no. 2, pp. 8559–8569, 2019.
- [9] M. Kumbhar, M. Survase, P. Mastud, and A. Salunke, "Real Time Web Based Bus Tracking System," *Int. Res. J. Eng. Technol.*, vol. 3, no. 2, pp. 632–635, 2016.
- [10] T. Budioko, "Peluang Riset dan Inovasi Bisnis Menggunakan Internet of Things," *Semin. Ris. Teknol. Inf.*, vol. VIII, pp. 353–358, 2016.
- [11] B. M. Susanto, E. S. J. Atmadji, and W. L. Brenkman, "Implementasi Mqtt Protocol Pada Smart Home Security Berbasis Web," *J. Inform. Polinema*, vol. 4, no. 3, p. 201, 2018, doi: 10.33795/jip.v4i3.207.
- [12] A. S. Gumilang, R. Primananda, and M. Data, "Pemodelan Sistem Pemantauan Posisi Kereta Api Berbasis RFID Menggunakan Protokol Message Queue Telemetry Transport (MQTT)," *J. Pengemb. Teknol. Inf. dan Ilmu Komput. Univ. Brawijaya*, vol. 1, no. 12, pp. 1475–1484, 2017.
- [13] D. Mediana and A. I. Nurhidayat, "Rancang Bangun

- Aplikasi Helpdesk (A-Desk) Berbasis Web Menggunakan Framework Laravel (Studi Kasus di PDAM Surya Sembada Kota Surabaya),” *J. Manaj. Inform.*, vol. 8, no. 2, pp. 75–81, 2018.
- [14] C. Hasiholan, R. Primananda, and K. Amron, “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.
- [15] S. Tomasua, D. Triyanto, and I. Nirmala, “Sistem Kendali Dan Monitoring Penggunaan Peralatan Listrik Di Rumah Menggunakan Raspberry Pi Dan Web Service,” *Jur. Sist. Komputer, Fak. MIPA Univ. Tanjungpura*, vol. Volume 4, no. 3, p. 96, 2016.
- [16] A. Roihan, A. A. Wisanto, Y. Sulaeman, F. M. Nur, and S. Williandi, “Implementasi Metode Realtime, Live Data Dan Parsing JSON Berbasis Mobile Dengan Menggunakan Android Studio Dan PHP Native,” *J. Teknol. Inf.*, vol. 5, no. 2, 2019.
- [17] “Laravel - The PHP Framework For Web Artisans.”
- [18] Rodiah Machdi Agustini, “Analisa Implementasi PowerLine Communication Sebagai Backbone WI-FI Extender,” *J. Tek.*, vol. 18, no. 1, pp. 1–7, 2017.
- [19] P. R. Utami, “Analisis Perbandingan Quality of Service Jaringan Internet Berbasis Wireless Pada Layanan Internet Service Provider (Isp) Indihome Dan First Media,” *J. Ilm. Teknol. dan Rekayasa*, vol. 25, no. 2, pp. 125–137, 2020, doi: 10.35760/tr.2020.v25i2.2723.