

Judul Artikel Seminar SANTIKA 2024

Punto Adji Bhirawa¹, Penulis Kedua^{2*}, Penulis Ketiga³

^{1,3} Afiliasi1 (Jurusan/Program Studi, Universitas)

¹penulis.pertama@universitas.ac.id

³penulis.ketiga@universitas.ac.id

² Afiliasi2 (Jurusan/Program Studi, Universitas)

*Corresponding author email: penulis.kedua@universitas.ac.id

Abstrak—Penelitian ini bertujuan untuk merancang dan mengimplementasikan sistem kendali perangkat berbasis Internet of Things (IoT) menggunakan Arduino ESP32 dan protokol Message Queuing Telemetry Transport (MQTT) dengan aplikasi MQTT Panel sebagai antarmuka pengguna. Sistem ini memungkinkan pengendalian motor IMCLab secara jarak jauh melalui perangkat smartphone dengan memanfaatkan konsep komunikasi *publish-subscribe* yang disediakan oleh MQTT. Arduino ESP32 berfungsi sebagai mikrokontroler utama yang terhubung ke jaringan internet dan berkomunikasi dengan broker MQTT untuk menerima perintah kendali. Aplikasi MQTT Panel digunakan untuk mengirimkan pesan berupa payload tertentu melalui topik MQTT yang telah ditentukan sebelumnya. Hasil pengujian menunjukkan bahwa ESP32 mampu terhubung secara stabil ke broker MQTT dan menerima perintah dari aplikasi MQTT Panel secara real-time dengan tingkat keterlambatan yang relatif kecil. Motor IMCLab dapat merespons setiap perintah yang dikirimkan sesuai dengan logika program yang telah dirancang. Dengan demikian, sistem yang dikembangkan dapat berfungsi dengan baik sebagai solusi kendali perangkat IoT jarak jauh yang sederhana, fleksibel, dan mudah diimplementasikan.

Kata Kunci— Internet of Things, Arduino ESP32, MQTT, MQTT Panel, Motor IMCLab

I. PENDAHULUAN

Perkembangan Internet of Things (IoT) telah mendorong integrasi antara perangkat fisik dan jaringan internet untuk mendukung sistem kendali dan monitoring jarak jauh. IoT memungkinkan perangkat seperti sensor dan aktuator untuk saling berkomunikasi secara real-time, sehingga meningkatkan efisiensi dan fleksibilitas sistem otomasi. Konsep IoT pertama kali diperkenalkan sebagai solusi untuk menghubungkan objek fisik ke dunia digital dan hingga saat ini telah banyak diterapkan pada berbagai bidang seperti industri, transportasi, dan pendidikan (Ashton, 2009). Dalam implementasi IoT, pemilihan protokol komunikasi menjadi aspek yang sangat penting, terutama untuk perangkat dengan sumber daya terbatas. MQTT merupakan salah satu protokol yang dirancang khusus untuk kebutuhan tersebut karena memiliki overhead data yang kecil serta menggunakan mekanisme *publish/subscribe*. Protokol ini terbukti efektif dalam menjaga kestabilan komunikasi pada jaringan dengan bandwidth rendah dan latensi tinggi, sehingga banyak digunakan dalam sistem IoT modern (Banks & Gupta, 2014).

Arduino ESP32 merupakan mikrokontroler yang banyak digunakan dalam pengembangan sistem IoT karena telah dilengkapi modul WiFi terintegrasi serta memiliki performa pemrosesan yang cukup tinggi. ESP32 mampu berperan sebagai client MQTT yang dapat mengirim dan menerima pesan dari broker secara andal. Berdasarkan dokumentasi teknis Espressif Systems, ESP32 mendukung berbagai protokol jaringan dan sangat sesuai untuk aplikasi kendali jarak jauh berbasis internet (Espressif Systems, 2023).

Penggunaan MQTT dalam sistem kendali aktuator, seperti motor, telah banyak diteliti dan menunjukkan hasil yang baik. Sistem kendali berbasis MQTT memungkinkan pengguna mengontrol perangkat dari jarak jauh melalui berbagai platform, termasuk smartphone. Penelitian sebelumnya menunjukkan bahwa MQTT mampu memberikan respon yang cepat dan stabil dalam aplikasi kontrol perangkat IoT dibandingkan dengan protokol komunikasi konvensional (Lian, 2021).

Berdasarkan uraian tersebut, penelitian ini memfokuskan pada perancangan dan implementasi sistem kontrol motor IMCLab menggunakan Arduino ESP32 dengan protokol MQTT yang dihubungkan ke aplikasi MQTT Panel pada smartphone. Sistem ini diharapkan dapat menjadi contoh penerapan IoT sederhana namun efektif untuk kendali perangkat keras secara jarak jauh, sekaligus menjadi referensi bagi pengembangan sistem IoT di bidang pendidikan dan penelitian.

II. TINJAUAN PUSTAKA

Cara paling mudah untuk memenuhi persyaratan format penulisan adalah dengan menggunakan dokumen ini sebagai *template*. Kemudian ketikkan teks Anda ke dalamnya

A. Internet of Things

Internet of Things (IoT) merupakan paradigma yang memungkinkan objek fisik dilengkapi dengan kemampuan komputasi, sensor, aktuator, dan konektivitas jaringan sehingga dapat saling berkomunikasi melalui internet. Konsep IoT pertama kali dipopulerkan untuk menggambarkan bagaimana objek sehari-hari dapat diintegrasikan ke dalam sistem informasi global guna meningkatkan efisiensi dan otomatisasi proses [1]. Dalam perkembangannya, IoT menjadi fondasi utama bagi sistem cerdas pada berbagai sektor, termasuk industri, transportasi, dan pendidikan.

Penerapan IoT memungkinkan pengumpulan dan pertukaran data secara real-time tanpa interaksi manusia secara langsung.

Sistem IoT umumnya terdiri dari tiga lapisan utama, yaitu lapisan persepsi (sensor dan aktuator), lapisan jaringan (komunikasi data), dan lapisan aplikasi (pengolahan dan visualisasi data). Arsitektur ini memberikan fleksibilitas dalam pengembangan sistem berskala kecil hingga besar [2].

Dalam konteks sistem kendali, IoT memberikan kemudahan dalam melakukan kontrol jarak jauh terhadap perangkat keras melalui jaringan internet. Beberapa penelitian menunjukkan bahwa implementasi IoT pada sistem kendali mampu meningkatkan efektivitas operasional dan mengurangi ketergantungan pada sistem manual, terutama ketika dikombinasikan dengan protokol komunikasi yang ringan dan andal [3].

B. ESP32

ESP32 merupakan mikrokontroler berbasis System on Chip (SoC) yang dikembangkan oleh Espressif Systems dan dirancang khusus untuk aplikasi IoT. ESP32 memiliki prosesor dual-core, memori yang cukup besar, serta modul WiFi dan Bluetooth terintegrasi sehingga mampu menangani komunikasi data dan pemrosesan secara bersamaan [4]. Keunggulan ini menjadikan ESP32 banyak digunakan dalam penelitian dan pengembangan sistem IoT.

Dari sisi fungsionalitas, ESP32 mendukung berbagai protokol komunikasi seperti TCP/IP, HTTP, dan MQTT. Dukungan ini memungkinkan ESP32 berperan sebagai client maupun server dalam sistem jaringan. Selain itu, ESP32 juga memiliki General Purpose Input Output (GPIO) yang dapat digunakan untuk mengendalikan sensor dan aktuator, termasuk motor [4]. Beberapa penelitian menunjukkan bahwa ESP32 memiliki kinerja yang stabil dan konsumsi daya yang relatif rendah dibandingkan mikrokontroler sekelasnya. Hal ini menjadikan ESP32 sangat cocok untuk sistem IoT berbasis kendali jarak jauh yang membutuhkan konektivitas kontinu dan respon cepat terhadap perintah yang diterima [5].

C. Protokol MQTT

Message Queuing Telemetry Transport (MQTT) adalah protokol komunikasi ringan yang menggunakan model publish/subscribe dengan broker sebagai pusat pertukaran pesan. Protokol ini dirancang untuk perangkat dengan sumber daya terbatas serta kondisi jaringan yang tidak stabil [6]. MQTT banyak digunakan dalam sistem IoT karena overhead data yang kecil dan efisiensi bandwidth yang tinggi.

Dalam mekanisme MQTT, publisher mengirimkan pesan ke broker melalui topik tertentu, sementara subscriber menerima pesan berdasarkan topik yang sama. Model ini memungkinkan komunikasi yang fleksibel dan terdesentralisasi, sehingga perangkat tidak perlu saling mengetahui alamat satu sama lain secara langsung [6].

Penelitian sebelumnya menunjukkan bahwa MQTT memiliki performa yang lebih baik dibandingkan protokol konvensional dalam hal latensi dan reliabilitas pada sistem IoT. Oleh karena itu, MQTT sering digunakan pada aplikasi kendali dan monitoring real-time, termasuk sistem kontrol motor berbasis mikrokontroler [7].

D. MQTT Panel

MQTT Panel merupakan aplikasi berbasis smartphone yang berfungsi sebagai client MQTT untuk melakukan monitoring dan kontrol perangkat IoT. Aplikasi ini menyediakan antarmuka grafis yang dapat dikonfigurasi sesuai kebutuhan pengguna, seperti tombol, slider, dan indikator status [8]. Dengan antarmuka tersebut, pengguna dapat mengirimkan perintah kontrol secara intuitif.

Penggunaan MQTT Panel mempermudah implementasi sistem kendali jarak jauh karena tidak memerlukan pengembangan aplikasi khusus. Pengguna cukup mengatur alamat broker, topik, serta payload pesan untuk berkomunikasi dengan perangkat IoT. Hal ini sangat membantu dalam proses pengujian dan pengembangan sistem [8].

Beberapa studi menunjukkan bahwa penggunaan aplikasi MQTT client pada smartphone mampu meningkatkan fleksibilitas dan aksesibilitas sistem IoT. Pengguna dapat melakukan kontrol dan monitoring kapan saja dan di mana saja selama terhubung ke jaringan internet [7].

III. METODE PENELITIAN

Paragraf harus teratur. Semua paragraf harus rata, yaitu sama-sama rata kiri dan dan rata kanan.

A. Tahap Persiapan Alat dan Bahan

Tahap persiapan alat dan bahan merupakan langkah awal yang sangat penting dalam perancangan dan implementasi sistem Internet of Things (IoT) berbasis Arduino ESP32 dan MQTT. Pada tahap ini dilakukan identifikasi dan pengumpulan seluruh komponen perangkat keras yang dibutuhkan agar sistem dapat berjalan sesuai dengan perancangan. Perangkat utama yang digunakan adalah Arduino ESP32 sebagai mikrokontroler yang memiliki kemampuan komunikasi WiFi terintegrasi, motor IMCLab sebagai aktuator yang akan dikendalikan, serta kabel USB yang berfungsi sebagai media pemrograman sekaligus sumber daya selama proses pengembangan dan pengujian. Selain itu, digunakan laptop atau komputer sebagai perangkat pendukung untuk menjalankan perangkat lunak pemrograman dan melakukan monitoring sistem.

Selain perangkat keras, tahap persiapan juga mencakup penyediaan perangkat lunak pendukung yang diperlukan untuk pengembangan sistem. Perangkat lunak yang digunakan antara lain Arduino IDE sebagai lingkungan pengembangan untuk menulis, mengkompilasi, dan mengunggah kode program ke Arduino ESP32. Library tambahan seperti library WiFi dan MQTT juga dipersiapkan untuk mendukung proses komunikasi data antara perangkat IoT dan broker MQTT. Di sisi pengguna, aplikasi MQTT Panel pada smartphone dipersiapkan sebagai antarmuka untuk mengirim perintah kendali ke sistem. Dengan persiapan alat dan bahan yang matang, proses perancangan, implementasi, dan pengujian sistem dapat dilakukan secara lebih terstruktur serta meminimalkan terjadinya kesalahan teknis.

B. Tahap Perancangan Sistem

Tahap perancangan sistem merupakan tahapan penting yang bertujuan untuk memastikan seluruh komponen perangkat keras dan perangkat lunak dapat terintegrasi dengan baik sebelum sistem diimplementasikan secara penuh. Pada tahap

ini dilakukan perancangan hubungan antara Arduino ESP32, motor IMCLab, sumber daya, serta perangkat pengembang berupa laptop. Perancangan sistem difokuskan pada bagaimana ESP32 dapat menerima perintah dari aplikasi MQTT Panel dan menerjemahkannya menjadi aksi fisik berupa pergerakan motor.

Pada sisi perangkat keras, Arduino ESP32 dipasang pada papan pengembangan (development board) dan dihubungkan dengan motor IMCLab melalui jalur input/output yang sesuai. Motor IMCLab dikendalikan menggunakan sinyal digital dari ESP32 melalui rangkaian driver yang telah tersedia pada modul IMCLab. Selain itu, ESP32 juga dihubungkan ke laptop menggunakan kabel USB yang berfungsi sebagai media pemrograman sekaligus sumber daya listrik selama tahap pengujian dan pengembangan sistem.

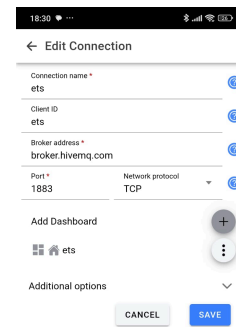
Pada sisi perangkat lunak, perancangan sistem mencakup penyusunan logika program yang akan dijalankan pada ESP32. Program dirancang agar ESP32 dapat terhubung ke jaringan WiFi, melakukan koneksi ke broker MQTT, serta melakukan subscribe pada topik tertentu. Setiap pesan yang diterima dari broker akan diproses oleh ESP32 dan diterjemahkan menjadi perintah kendali motor. Tahap perancangan ini memastikan alur kerja sistem, mulai dari pengiriman perintah dari aplikasi hingga eksekusi pada motor, dapat berjalan secara terstruktur dan efisien.



Gbr. 1 Perancangan Sistem IoT

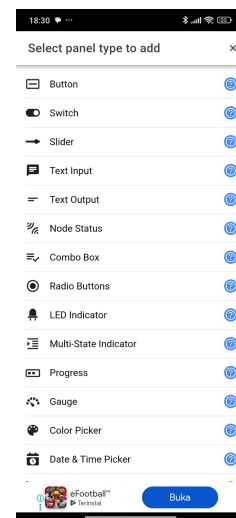
C. Tahap Konfigurasi Aplikasi MQTT Panel

Tahap konfigurasi aplikasi MQTT Panel merupakan langkah penting agar smartphone dapat berkomunikasi dengan perangkat IoT melalui broker MQTT. Konfigurasi diawali dengan pembuatan *connection* baru pada aplikasi MQTT Panel. Pada menu Edit Connection, pengguna mengisi beberapa parameter utama, yaitu *Connection Name* sebagai identitas koneksi, *Client ID* yang bersifat unik, alamat broker MQTT, serta port komunikasi. Pada penelitian ini digunakan broker publik broker.hivemq.com dengan port 1883 dan protokol jaringan TCP.



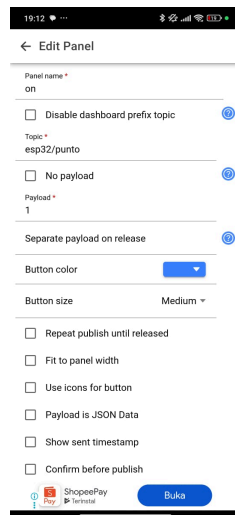
Gbr. 2 Konfigurasi Koneksi

Setelah koneksi berhasil dibuat dan disimpan, langkah selanjutnya adalah pembuatan *dashboard* sebagai antarmuka kontrol. Dashboard berfungsi sebagai media interaksi antara pengguna dan sistem IoT. Pada tahap ini pengguna menambahkan panel kontrol dengan memilih menu Select panel type to add yang menyediakan berbagai jenis panel seperti *Button*, *Switch*, *Slider*, dan indikator lainnya.



Gbr. 3 Setup Panel

Dalam penelitian ini, panel *Button* digunakan sebagai pengendali utama motor IMCLab. *Button* dikonfigurasi dengan topik MQTT yang sama dengan topik yang disubscribe oleh ESP32. Selain itu, payload pesan diatur sesuai dengan logika program pada ESP32, misalnya nilai tertentu untuk menyalakan atau mematikan motor. Ketika pengguna menekan tombol pada dashboard, aplikasi MQTT Panel akan mengirimkan pesan ke broker MQTT yang selanjutnya diteruskan ke ESP32.



Gbr. 4 Setup Button

Setelah dashboard berhasil dibuat, tahap selanjutnya adalah melakukan konfigurasi panel *Button* yang berfungsi sebagai media pengirim perintah dari aplikasi ke perangkat Arduino ESP32. Pada konfigurasi ini, pengguna memilih jenis panel *Button* kemudian masuk ke menu *Edit Panel* untuk mengatur parameter komunikasi MQTT. Parameter pertama yang diatur adalah *Panel name*, yang pada penelitian ini diberi nama “on” untuk merepresentasikan fungsi tombol sebagai perintah mengaktifkan motor IMCLab. Penamaan panel yang jelas bertujuan untuk memudahkan pengguna dalam memahami fungsi tombol saat aplikasi digunakan.

Selanjutnya, pada bagian *Topic*, diisi topik MQTT yang digunakan sebagai jalur komunikasi antara aplikasi dan ESP32, yaitu *esp32/punto*. Topik ini harus sama dengan topik yang telah didefinisikan di dalam kode program Arduino ESP32 agar pesan yang dikirim dapat diterima dan diproses dengan benar. Pada bagian *Payload*, diisi nilai “1” yang berfungsi sebagai data perintah untuk menyalakan atau menggerakkan motor. Payload ini akan dikirimkan ke broker MQTT setiap kali tombol ditekan, kemudian diteruskan ke ESP32 untuk dieksekusi sesuai logika program yang telah dibuat.

Selain pengaturan utama tersebut, beberapa opsi tambahan juga tersedia untuk menyesuaikan perilaku tombol, seperti pengaturan warna tombol (*Button color*) dan ukuran tombol (*Button size*) agar tampilan dashboard lebih informatif dan mudah digunakan. Opsi lain seperti *Repeat publish until released*, *Payload is JSON Data*, dan *Confirm before publish* dapat disesuaikan sesuai kebutuhan sistem, namun pada penelitian ini pengaturan standar sudah mencukupi karena sistem hanya memerlukan pengiriman payload sederhana. Dengan konfigurasi panel button yang tepat, aplikasi MQTT Panel mampu mengirim perintah secara real-time ke Arduino ESP32, sehingga motor IMCLab dapat dikendalikan langsung melalui smartphone dengan respons yang cepat dan stabil.

IV. HASIL DAN PEMBAHASAN

A. Hasil Implementasi Sistem

Hasil implementasi sistem menunjukkan bahwa perangkat IoT berbasis Arduino ESP32 berhasil terhubung dengan broker MQTT dan dapat dikendalikan melalui aplikasi MQTT Panel pada smartphone. ESP32 mampu melakukan koneksi ke jaringan WiFi dan selanjutnya terhubung ke broker MQTT publik HiveMQ menggunakan protokol TCP pada port 1883. Setelah koneksi berhasil, perangkat dapat melakukan proses *subscribe* pada topik yang telah ditentukan, yaitu *esp32/punto*, sehingga siap menerima pesan perintah dari aplikasi.

Pada sisi aplikasi, MQTT Panel berhasil dikonfigurasi sebagai antarmuka pengguna untuk mengirim perintah kendali. Dashboard yang dibuat menampilkan panel *Button* yang telah diatur dengan payload tertentu. Ketika tombol ditekan, aplikasi mengirimkan pesan MQTT ke broker, kemudian pesan tersebut diteruskan ke ESP32. Berdasarkan hasil pengujian, setiap pesan yang dikirim dari aplikasi dapat diterima dengan baik oleh ESP32 tanpa adanya keterlambatan yang signifikan.

Perangkat keras berupa motor IMCLab yang terhubung ke ESP32 juga dapat merespons perintah dengan baik. Saat payload “1” diterima oleh ESP32, motor aktif dan bergerak sesuai perancangan, sedangkan saat payload berbeda atau kondisi tertentu diberikan, motor berhenti. Hal ini menunjukkan bahwa integrasi antara perangkat lunak (MQTT Panel dan program ESP32) dan perangkat keras (motor IMCLab) telah berjalan sesuai dengan tujuan sistem.

B. Hasil Pengujian Sistem

Pengujian sistem dilakukan untuk memastikan bahwa setiap komponen bekerja sesuai dengan fungsinya. Pengujian pertama dilakukan pada konektivitas ESP32 terhadap jaringan WiFi dan broker MQTT. Berdasarkan hasil pengujian, ESP32 mampu terhubung secara stabil ke broker MQTT selama jaringan internet tersedia. Status koneksi dapat dipantau melalui serial monitor Arduino IDE, yang menunjukkan bahwa proses *connect* dan *subscribe* berhasil dilakukan.

Pengujian berikutnya dilakukan pada pengiriman pesan dari aplikasi MQTT Panel. Tombol yang telah dikonfigurasi diuji dengan menekan tombol secara berulang untuk melihat konsistensi pengiriman payload. Hasil pengujian menunjukkan bahwa setiap penekanan tombol menghasilkan pesan MQTT yang diterima oleh ESP32 secara real-time. Tidak ditemukan kegagalan pengiriman pesan selama proses pengujian berlangsung.

Pengujian terakhir dilakukan pada respons aktuator, yaitu motor IMCLab. Motor diuji untuk merespons perintah ON yang dikirim melalui MQTT Panel. Berdasarkan hasil pengamatan, motor dapat aktif dan berhenti sesuai dengan perintah yang diberikan. Waktu respons motor relatif cepat karena protokol MQTT bersifat ringan (*lightweight*), sehingga sangat sesuai digunakan untuk sistem kendali IoT berbasis jaringan internet.

V. KESIMPULAN

Berdasarkan hasil perancangan, implementasi, dan pengujian yang telah dilakukan, dapat disimpulkan bahwa sistem kendali perangkat IoT berbasis Arduino ESP32 menggunakan protokol MQTT telah berhasil direalisasikan dengan baik. ESP32 mampu terhubung ke jaringan internet dan

berkomunikasi dengan broker MQTT secara stabil, sehingga memungkinkan pertukaran data antara perangkat dan aplikasi pengguna melalui jaringan internet.

Penggunaan aplikasi MQTT Panel sebagai antarmuka kendali terbukti efektif dan mudah digunakan. Konfigurasi koneksi, dashboard, serta panel button dapat dilakukan dengan relatif sederhana tanpa memerlukan pengembangan aplikasi khusus. Melalui pengaturan topik dan payload yang sesuai, aplikasi MQTT Panel mampu mengirimkan perintah kendali ke ESP32 secara real-time, sehingga sistem dapat dioperasikan langsung melalui smartphone.

Hasil pengujian menunjukkan bahwa motor IMCLab sebagai aktuator dapat merespons perintah yang dikirimkan melalui MQTT dengan baik. Setiap payload yang diterima oleh ESP32 diproses sesuai dengan logika program, sehingga motor dapat aktif dan berhenti sesuai perintah pengguna. Hal ini menunjukkan bahwa integrasi antara perangkat keras dan perangkat lunak dalam sistem telah berjalan sesuai dengan perancangan yang ditetapkan.

Secara keseluruhan, sistem yang dikembangkan telah memenuhi tujuan penelitian, yaitu membangun sistem kendali jarak jauh berbasis IoT menggunakan ESP32 dan MQTT. Meskipun demikian, sistem masih memiliki keterbatasan, terutama pada aspek keamanan dan ketergantungan terhadap koneksi internet. Oleh karena itu, pengembangan lebih lanjut dapat dilakukan dengan menerapkan mekanisme keamanan tambahan serta penggunaan broker MQTT privat untuk meningkatkan keandalan sistem.

UCAPAN TERIMA KASIH

Judul untuk ucapan terima kasih dan referensi tidak diberi nomor. Terima kasih disampaikan kepada Tim SANTIKA yang telah meluangkan waktu untuk membuat *template* ini.

REFERENSI

- [1] [1] K. Ashton, "That 'Internet of Things' Thing," *RFID Journal*, Jun. 2009.
- [2] [2] L. Atzori, A. Iera, and G. Morabito, "The Internet of Things: A survey," *Computer Networks*, vol. 54, no. 15, pp. 2787–2805, Oct. 2010.
- [3] [3] A. Zanella, N. Bui, A. Castellani, L. Vangelista, and M. Zorzi, "Internet of Things for Smart Cities," *IEEE Internet of Things Journal*, vol. 1, no. 1, pp. 22–32, Feb. 2014.
- [4] [4] Espressif Systems, *ESP32 Technical Reference Manual*, Espressif Systems Inc., 2023.
- [5] [5] M. S. Hossain and G. Muhammad, "Cloud-assisted Industrial Internet of Things (IIoT) – Enabled framework for health monitoring," *Computer Networks*, vol. 101, pp. 192–202, 2016.
- [6] [6] A. Banks and R. Gupta, *MQTT Version 3.1.1*, OASIS Standard, Oct. 2014.
- [7] [7] S. H. Lian, "Implementation of IoT Control System Using MQTT Protocol," *International Journal of Engineering Research and Technology (IJERT)*, vol. 10, no. 3, pp. 245–250, 2021.
- [8] [8] J. Lightfoot, "MQTT Client Applications for Mobile-Based IoT Monitoring," *International Journal of Computer Applications*, vol. 178, no. 7, pp. 15–20, 2019.
- [9] [9] R. A. Pratama, D. S. Putra, and Y. Christyono, "Rancang Bangun Sistem Kendali Motor DC Berbasis Internet of Things," *Jurnal Teknik Elektro*, vol. 8, no. 2, pp. 85–92, 2020.