

IMPLEMENTASI INTERNET OF THINGS (IoT) DENGAN PROTOKOL KOMUNIKASI MQTT PADA SISTEM KONTROL LAMPU RUANGAN

Moh. Ali Makhrus⁻¹, Risqy Siwi Pradini⁻², Nindynar Rikatsih⁻³

Informatika, Fakultas Sains dan Teknologi, Institut Teknologi, Sains, dan Kesehatan RS. DR. Soepraoen Kesdam
V/BRW^{-1,2,3}

email: mohalimakhrus@itsk.soepraoen.ac.id⁻¹ risqypradini@itsk.soepraoen.ac.id⁻² nindynar@itsk.soepraoen.ac.id⁻³

1

Abstract- Meningkatnya kebutuhan otomasi perangkat elektronik mendorong pemanfaatan teknologi Internet of Things (IoT) guna meningkatkan efisiensi energi, kemudahan kendali, dan fleksibilitas akses. Penelitian ini merancang dan mengimplementasikan sistem kontrol lampu berbasis IoT dengan menggunakan protokol Message Queuing Telemetry Transport (MQTT) yang efisien dan ringan dalam komunikasi data. Sistem memanfaatkan mikrokontroler NodeMCU ESP8266, modul Real Time Clock (RTC) DS3231 untuk penjadwalan otomatis, dan Wi-Fi Manager untuk konfigurasi jaringan melalui antarmuka web tanpa pengunggahan ulang kode. Pengujian mencakup evaluasi waktu respons, konsumsi bandwidth, dan stabilitas koneksi saat perpindahan jaringan. Hasil menunjukkan sistem memiliki waktu respons rata-rata 0,05–0,08 detik dan konsumsi bandwidth minimum 70 byte per detik. Sistem juga mampu berpindah jaringan secara otomatis tanpa gangguan fungsi, menjadikannya solusi andal dan efisien untuk kontrol pencahayaan pada skala rumah tangga, institusional, dan komersial.

Kata Kunci: Internet of Things, MQTT, NodeMCU ESP8266, Kendali Otomatis, Wi-Fi Manager

1. PENDAHULUAN

Perkembangan teknologi Internet of Things (IoT) telah memberikan kontribusi signifikan dalam otomasi dan kendali perangkat elektronik melalui jaringan internet [1]. IoT memungkinkan integrasi antara perangkat keras dan perangkat lunak, sehingga memungkinkan pemantauan serta pengendalian perangkat secara real-time dari jarak jauh [2]. Salah satu implementasi yang berkembang pesat adalah sistem pengendalian pencahayaan, yang tidak hanya meningkatkan efisiensi energi tetapi juga kenyamanan pengguna [3].

Sistem pencahayaan di dalam ruangan, khususnya pada lingkungan rumah tinggal, umumnya masih menggunakan metode konvensional dengan saklar

manual [4]. Dalam situasi tertentu, seperti ketika penghuni sedang terburu-buru atau tidak berada di rumah, terdapat risiko lampu dibiarkan menyala, yang berdampak pada pemborosan energi [5]. Selain itu, keterbatasan fisik untuk menjangkau saklar secara langsung menjadi kendala, terutama ketika kontrol diperlukan secara simultan di beberapa ruangan. Tantangan semakin besar ketika pengguna berada di luar kota atau provinsi, sehingga tidak memungkinkan untuk memantau atau mengendalikan kondisi lampu secara langsung [6].

Sejumlah penelitian telah mengembangkan sistem kendali lampu berbasis IoT. Herlina et al. (2022) merancang sistem otomatisasi lampu menggunakan Blynk dan NodeMCU ESP8266 untuk meningkatkan efisiensi energi [7]. Kurnianto et al. (2022) menunjukkan penerapan sistem kendali lampu berbasis MQTT dengan antarmuka web. Hasil pengujian menunjukkan sistem dapat bekerja secara real-time, mendukung efektivitas MQTT dalam kendali jarak jauh yang efisien [8]. Namun, sebagian besar sistem masih mengandalkan kredensial jaringan yang ter-"hard-code", sehingga memerlukan pemrograman ulang saat terjadi perubahan jaringan Wi-Fi. Solusi teknis seperti library WiFiManager untuk ESP8266/ESP32 memungkinkan penyimpanan kredensial Wi-Fi secara dinamis melalui captive portal, sehingga perangkat dapat berpindah jaringan tanpa perlu mengunggah ulang kode program [9].

Dalam konteks ini, protokol komunikasi Message Queuing Telemetry Transport (MQTT) menjadi alternatif yang ringan dan efisien, terutama dalam hal konsumsi bandwidth [10]. MQTT juga mendukung komunikasi data yang cepat dan andal antara perangkat IoT dan server

[11]. Meskipun demikian, tantangan utama tetap terletak pada pengelolaan jaringan Wi-Fi, yang hingga kini masih menjadi penghambat dalam penerapan sistem yang benar-benar fleksibel dan mandiri.

Penelitian ini bertujuan untuk merancang dan mengimplementasikan sistem pengendalian lampu berbasis IoT menggunakan protokol MQTT yang dipadukan dengan fitur Wi-Fi Manager. Fitur ini memungkinkan konfigurasi ulang jaringan Wi-Fi melalui antarmuka web tanpa perlu mengunggah ulang kode program. Dengan demikian, perangkat dapat berpindah jaringan secara otomatis tanpa mengganggu fungsionalitas sistem. Solusi ini diharapkan mampu menghadirkan sistem kontrol pencahayaan yang andal, efisien, dan adaptif untuk berbagai lingkungan, mulai dari rumah tangga hingga institusi dan sektor komersial.

2. METODE PENELITIAN

Penelitian ini menggunakan metode eksperimen dengan pendekatan rekayasa prototipe, di mana sistem kontrol lampu berbasis Internet of Things (IoT) direalisasikan dalam bentuk perangkat fungsional. Prototipe dirancang menyerupai kondisi nyata di lingkungan rumah tinggal untuk mensimulasikan implementasi sesungguhnya. Tahapan penelitian meliputi: (1) tahap perancangan sistem, (2) implementasi sistem, serta (3) analisis dan pengujian sistem untuk mengevaluasi kinerja dalam berbagai kondisi operasional [12] yang diilustrasikan pada gambar 1.



Gambar 1. Alur Metode Penelitian

2.1. Perancangan Sistem

Tahap ini mencakup perancangan awal sistem secara menyeluruh, dimulai dari penentuan dimensi dan tata letak komponen dalam prototipe ruangan. Desain mencakup penempatan modul elektronik seperti NodeMCU ESP8266, relay, RTC DS3231, serta lampu sebagai aktuator. Selain itu, sistem kendali berbasis web dirancang dengan integrasi protokol MQTT untuk menghubungkan mikrokontroler dengan antarmuka pengguna melalui jaringan internet. Tujuan dari perancangan ini adalah memastikan integrasi fungsional seluruh komponen dalam sistem kontrol lampu jarak jauh dan otomatis [13].

2.2. Implementasi Sistem

Tahap ini melibatkan serangkaian analisis dan pengujian untuk memastikan bahwa sistem kontrol lampu berbasis IoT berfungsi sesuai dengan rancangan. Pengujian dimulai dengan verifikasi perangkat keras, termasuk pemeriksaan pada rangkaian NodeMCU ESP8266, modul relay, RTC DS3231, serta lampu sebagai aktuator. Setelah perangkat keras dinyatakan berfungsi dengan baik, pengujian dilanjutkan pada sistem kendali berbasis web yang menggunakan protokol MQTT. Evaluasi dilakukan terhadap konektivitas antara mikrokontroler dan broker MQTT, serta keandalan proses pengiriman dan penerimaan data kontrol secara real-time melalui jaringan Wi-Fi [14].

2.3 Analisis dan Pengujian Sistem

Tahap ini bertujuan untuk mengevaluasi kinerja sistem kontrol lampu berbasis IoT secara menyeluruh, meliputi pengujian perangkat keras, perangkat lunak, protokol komunikasi, serta integrasi sistem secara keseluruhan. Proses evaluasi dilakukan untuk memastikan bahwa sistem berjalan sesuai spesifikasi desain dan mampu beroperasi secara andal dalam berbagai kondisi operasional [15].

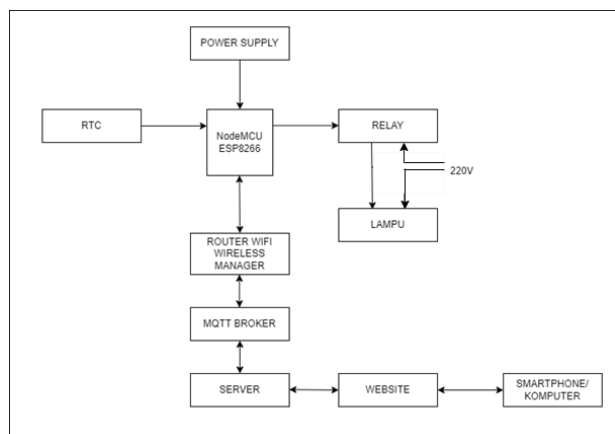
3. HASIL DAN PEMBAHASAN

3.1. Hasil Perancangan Sistem

Sistem pengendalian lampu ruangan berbasis Internet of Things (IoT) ini memanfaatkan protokol komunikasi Message Queuing Telemetry Transport (MQTT), koneksi internet, dan antarmuka website untuk melakukan pemantauan dan pengendalian lampu secara otomatis dalam sebuah ruangan.

3.1.1 Blok Diagram dan Flowchart

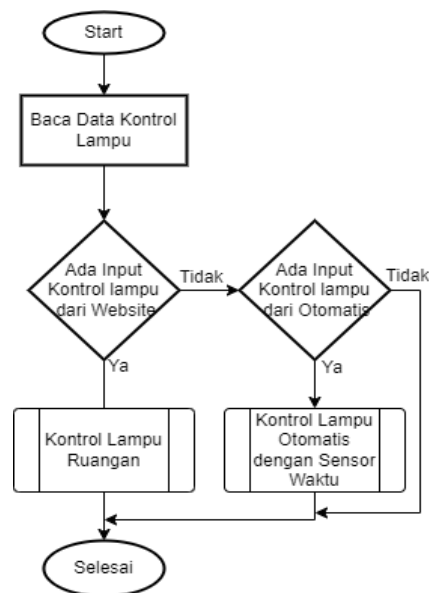
Sistem ini mengandalkan konektivitas Wi-Fi untuk menghubungkan perangkat ke jaringan internet dan memungkinkan pengendalian melalui dashboard berbasis web [16]. Pengguna dapat mengirimkan perintah melalui dashboard, yang diproses oleh mikrokontroler menggunakan protokol MQTT dengan metode publish/subscribe. Mikrokontroler kemudian mengendalikan relay untuk mengaktifkan atau menonaktifkan lampu sesuai perintah yang diterima. Modul Real-Time Clock (RTC) juga digunakan untuk pengaturan otomatis berdasarkan waktu, misalnya mematikan lampu pada pukul 06.00 dan menyalakan kembali pada pukul 17.00 guna meningkatkan efisiensi energi.



Gambar 2. Blok Diagram

Pada gambar 2 blok diagram sistem diatas menunjukkan hubungan antar komponen utama dalam sistem pengendalian lampu berbasis IoT dengan protokol MQTT. Sistem terdiri atas NodeMCU ESP8266, modul relay, power supply, modul RTC, lampu, router Wi-Fi dengan Wi-Fi Manager, MQTT broker, server, website, serta perangkat pengguna (smartphone atau komputer). Berikut penjelasan masing-masing komponen:

1. Smartphone/Komputer: Perangkat pengguna untuk mengakses dashboard dan mengirim perintah kendali.
2. Website: Antarmuka berbasis web yang digunakan untuk memonitor status dan mengendalikan lampu melalui browser.
3. Server: Menjembatani komunikasi antara antarmuka web dan mikrokontroler.
4. MQTT Broker: Mengelola pertukaran data antar perangkat menggunakan metode publish dan subscribe.
5. Router Wi-Fi / Wi-Fi Manager: Menyediakan koneksi internet serta memungkinkan konfigurasi jaringan secara fleksibel.
6. NodeMCU ESP8266: Mikrokontroler yang menerima perintah dari MQTT broker untuk mengendalikan perangkat keras.
7. RTC (DS3231): Modul penentu waktu untuk mengatur jadwal otomatis hidup/matinya lampu.
8. Relay: Komponen saklar elektronik yang menghubungkan atau memutus arus ke lampu.
9. Power Supply: Menyediakan tegangan dan arus yang dibutuhkan untuk mendukung operasi seluruh sistem.



Gambar 3. Flowchart Keseluruhan Sistem

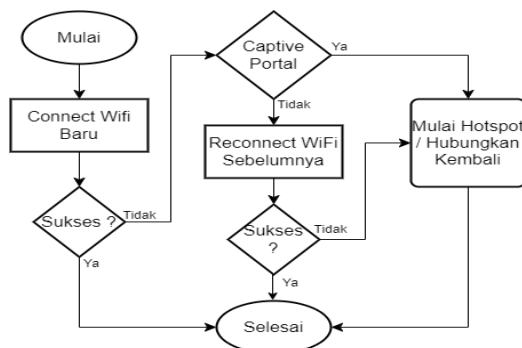
Pada gambar 3 flowchart keseluruhan sistem diatas menggambarkan alur logika pengendalian lampu yang dimulai dari pembacaan status variabel lampu. Sistem menerima input baik dari pengguna melalui website maupun dari modul RTC secara otomatis. Jika terdapat input dari website, NodeMCU akan menjalankan perintah untuk menyalakan atau mematikan lampu sesuai permintaan. Jika input berasal dari RTC, maka sistem akan secara otomatis mengaktifkan atau menonaktifkan lampu berdasarkan jadwal waktu yang telah ditentukan, seperti menyala pada sore hari dan mati di pagi hari. Setelah perintah dijalankan, sistem kembali memantau kondisi dan menunggu input berikutnya.

3.1.2 Wi-fi Manager

Wi-Fi Manager merupakan pustaka (library) yang digunakan untuk menghubungkan mikrokontroler ke jaringan internet tanpa perlu melakukan pemrograman ulang saat terjadi perubahan jaringan. Konfigurasi jaringan dilakukan melalui antarmuka berbasis web (captive portal), yang secara otomatis muncul ketika perangkat belum terhubung ke jaringan Wi-Fi. Melalui portal ini, pengguna dapat memasukkan nama jaringan (SSID) dan kata sandi Wi-Fi yang tersedia, sehingga memudahkan pengaturan konektivitas tanpa akses langsung ke kode program.

Pada gambar 4 flowchart wifi manager dibawah menggambarkan alur kerja Wi-Fi Manager dalam mengelola koneksi jaringan. Ketika perangkat dinyalakan, Wi-Fi Manager akan mencoba menghubungkan mikrokontroler ke jaringan Wi-Fi

terakhir yang berhasil digunakan. Jika koneksi berhasil, sistem akan langsung terhubung ke internet dan siap digunakan. Namun, jika koneksi gagal, Wi-Fi Manager akan mengaktifkan mode akses titik (access point) dan menjalankan captive portal.



Gambar 4. Flowchart Wi-Fi Manager

Dalam mode captive portal, pengguna dapat terhubung langsung ke mikrokontroler melalui jaringan Wi-Fi sementara yang disediakan perangkat. Setelah itu, pengguna akan diarahkan ke halaman konfigurasi melalui browser, tempat mereka dapat memasukkan SSID dan password jaringan Wi-Fi yang tersedia. Informasi jaringan ini kemudian disimpan secara otomatis oleh perangkat. Jika koneksi berhasil terjalin, mikrokontroler akan menggunakan pengaturan tersebut untuk terhubung ke jaringan, dan captive portal akan dinonaktifkan.

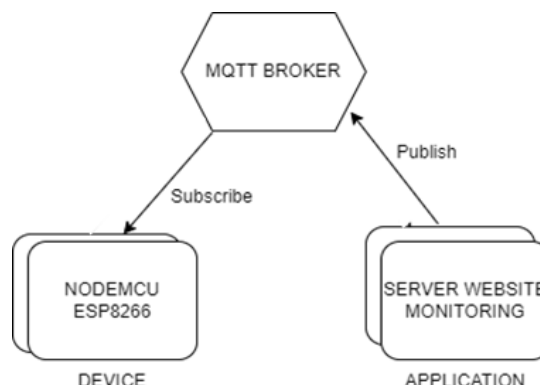
Jika jaringan yang baru dimasukkan tidak tersedia atau terjadi kesalahan koneksi, Wi-Fi Manager akan mencoba kembali ke jaringan lama (jika masih disimpan). Jika semua koneksi gagal, sistem akan kembali mengaktifkan captive portal dan menunggu konfigurasi ulang dari pengguna.

3.1.3 Protokol Komunikasi MQTT

Protokol Message Queuing Telemetry Transport (MQTT) merupakan protokol komunikasi ringan yang dirancang untuk mentransmisikan data secara efisien antar perangkat, khususnya dalam ekosistem Internet of Things (IoT). MQTT dirancang untuk bekerja di atas protokol TCP/IP dan sangat cocok digunakan pada perangkat dengan sumber daya terbatas serta jaringan dengan bandwidth rendah [17].

MQTT menggunakan pendekatan komunikasi berbasis publish/subscribe, di mana perangkat pengirim (publisher) mengirimkan pesan ke topik tertentu, sedangkan perangkat penerima (subscriber) akan menerima pesan dari topik yang telah mereka daftarkan sebelumnya [18]. Dalam arsitektur ini, peran MQTT broker sangat penting sebagai perantara untuk mengelola dan mendistribusikan pesan antar perangkat.

Dalam sistem ini, mikrokontroler NodeMCU bertindak sebagai publisher dan subscriber sekaligus, tergantung pada konteks komunikasi. Pengguna mengirimkan perintah melalui website (dashboard), yang kemudian dipublish ke broker MQTT. Broker akan meneruskan pesan tersebut kepada NodeMCU yang telah subscribe ke topik terkait, dan sistem akan merespons dengan menyalakan atau mematikan lampu.



Gambar 5. Protokol Komunikasi MQTT

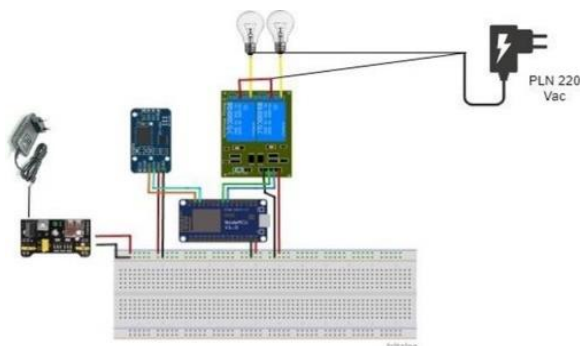
Pada gambar 5 blok diagram komunikasi data protokol MQTT diatas menggambarkan interaksi antar komponen dalam sistem pengendalian lampu berbasis IoT. Diagram yang diilustrasikan pada gambar 5 ini juga menjelaskan bagaimana data dikirim dan diterima antara perangkat pengguna, broker MQTT, dan mikrokontroler. Dalam sistem ini, pengguna (melalui website) bertindak sebagai publisher, mengirimkan perintah ke broker MQTT menggunakan topik tertentu, seperti TA/relay atau SK/relay. Broker kemudian mendistribusikan pesan tersebut kepada perangkat yang telah subscribe pada topik yang sama, yaitu NodeMCU ESP8266 [19]. Mikrokontroler akan membaca pesan yang diterima dan mengeksekusinya untuk menghidupkan atau mematikan lampu melalui modul relay.

Sebaliknya, mikrokontroler juga dapat bertindak sebagai publisher dengan mengirimkan status terkini dari lampu ke broker, yang kemudian dapat ditampilkan kembali di dashboard sebagai informasi monitoring. Mekanisme ini memungkinkan komunikasi dua arah yang efisien dan real-time antar perangkat. MQTT broker di sini berfungsi sebagai pusat komunikasi, yang tidak hanya menerima dan meneruskan pesan, tetapi juga memastikan bahwa pesan dikirim ke perangkat yang benar sesuai dengan topik yang relevan.

3.1.4 Desain Perancangan Tata Letak

Desain tata letak sistem menunjukkan susunan fisik komponen utama yang ditempatkan di dalam box kendali. Komponen yang digunakan dalam sistem ini antara lain

NodeMCU ESP8266 sebagai mikrokontroler utama, modul RTC DS3231 sebagai pengatur waktu otomatis, relay 2-channel sebagai saklar elektronik, modul MB102 power supply untuk penyedia daya, serta lampu sebagai perangkat output.

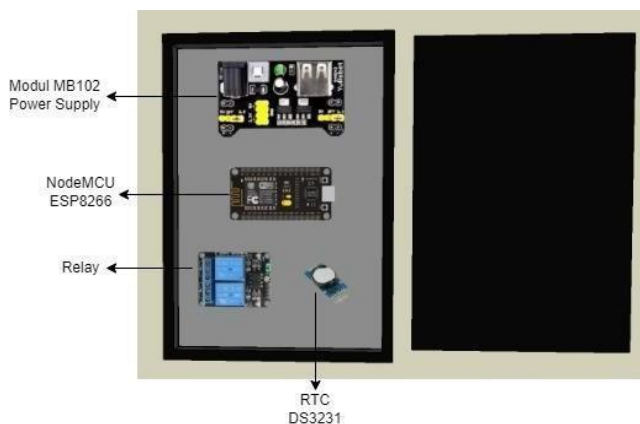


Gambar 6. Desain Sistem Pengontrolan Lampu

Beberapa sambungan penting dalam rangkaian antara NodeMCU dengan komponen lain, antara lain:

1. Power Supply ke NodeMCU: V+ ke Vin (merah), V- ke GND (hitam)
2. RTC ke NodeMCU: Vcc ke Vin (merah), GND ke GND (hitam), SCL ke D1 (oranye), SDA ke D2 (cyan)
3. Relay ke NodeMCU: Vcc ke Vin (merah), GND ke GND (hitam), IN1 ke D5 (hijau), IN2 ke D6 (biru).

Perakitan dilakukan pada box berukuran 20×15×10 cm berbahan plastik kokoh dan rapi. Sistem dirancang agar seluruh komponen terhubung sesuai dengan skema wiring untuk menjamin keandalan dan keamanan operasional. Kabel listrik jenis transparan 2x50 dengan panjang fleksibel digunakan untuk koneksi lampu ke relay, sehingga memungkinkan pemasangan alat di berbagai lokasi ruangan, termasuk pada atap atau dinding. Tata letak ini dirancang untuk mempermudah perawatan dan pemantauan sistem serta meminimalkan gangguan teknis akibat kesalahan koneksi atau posisi pemasangan komponen yang tidak ergonomis



Gambar 7. Desain Perancangan Alat Dalam Box

Gambar 7 menunjukkan tampilan fisik dari sistem yang telah dirakit di dalam box kendali. Komponen utama seperti NodeMCU ESP8266, relay 2-channel, modul RTC DS3231, dan power supply MB102 ditempatkan secara teratur agar mudah diakses dan dirawat. Penataan kabel dan konektor mengikuti skema wiring yang telah dirancang sebelumnya. Posisi tiap komponen disesuaikan untuk menjaga aliran udara dan mencegah gangguan fungsional saat alat beroperasi. Desain ini memastikan alat siap diuji dan digunakan secara praktis pada lingkungan nyata.

3.2. Hasil Implementasi Sistem

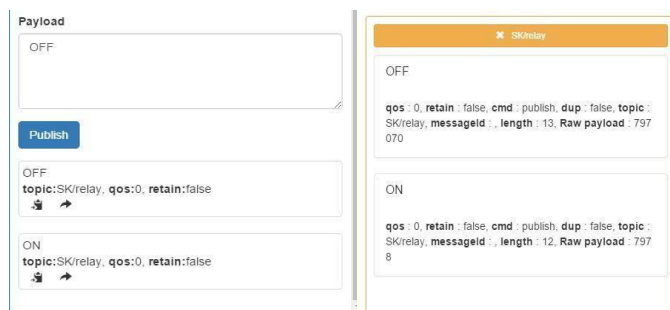
3.2.1 Implementasi Protokol Komunikasi MQTT

Protokol komunikasi MQTT (Message Queuing Telemetry Transport) diimplementasikan sebagai medium utama untuk proses komunikasi data (pengiriman dan penerimaan perintah) secara efisien pada sistem IoT (Internet of Things) yang dibangun. Sistem ini menggunakan broker publik EMQX dan aplikasi MQTTBox untuk manajemen serta pengujian komunikasi data.

Implementasi MQTT dalam sistem ini menggunakan dua topik utama:

- **TA/relay:** Digunakan untuk kontrol lampu 1
- **SK/relay:** Digunakan untuk kontrol lampu 2

Penggunaan broker publik `ws://broker.emqx.io:8083/mqtt` dipilih karena kemudahan akses dan keandalannya untuk komunikasi MQTT dasar [20]. Broker ini mendukung mekanisme publish/subscribe dengan efisiensi bandwidth yang optimal, sesuai dengan kebutuhan aplikasi IoT dan komunikasi machine-to-machine (M2M).



Gambar 8. Publish and subscribe di MQTTBox

Gambar 8 menunjukkan proses publish dan subscribe yang berhasil dilakukan menggunakan MQTTBox. Data ON & OFF telah berhasil dipublikasi dan disubskripsi dengan sukses pada topik SK/Relay menggunakan broker publik yang telah ditentukan.

```
Message (Enter to send message to 'NodeMCU 1.0 (ESP-12E Module)' on 'COM4')
10:29:02.634 -> [WiFi] Connecting to wifi...
10:29:02.659 -> [WiFi] Connecting to WiFi...
10:29:02.661 -> [WiFi] Connected!
10:29:02.661 -> 192.168.1.13
10:29:02.766 -> [MQTT] Connected!
10:29:29.366 -> [MQTT] Message received: SK/relay - OFF
10:29:37.564 -> [MQTT] Message received: SK/relay - ON
10:30:20.702 -> [MQTT] Message received: SK/relay - ON
10:30:31.831 -> [MQTT] Message received: SK/relay - OFF
```

Gambar 9. Subscribe MQTT pada Serial Monitor Mikrokontroler

Pada Gambar 9, dapat diamati bahwa mikrokontroler berfungsi sebagai subscriber yang efektif untuk topik MQTT tertentu. Data yang diterima dari broker dapat diakses dan ditampilkan melalui Serial Monitor pada lingkungan pengembangan Arduino IDE [21]. Hal ini memverifikasi kemampuan sistem dalam melakukan monitoring dan pengendalian perangkat secara real-time melalui protokol MQTT. Salah satu keunggulan protokol MQTT yang terimplementasi dalam sistem ini adalah fitur time decoupling, yang memungkinkan publisher dan subscriber tidak perlu terhubung secara bersamaan untuk bertukar data [22]. Subscriber dapat terputus setelah berlangganan ke broker dan kemudian terhubung kembali untuk menerima data terbaru yang dikirimkan selama periode offline, suatu proses yang dikenal sebagai "mode offline".

Tabel 1. Data Latency Protokol Komunikasi MQTT

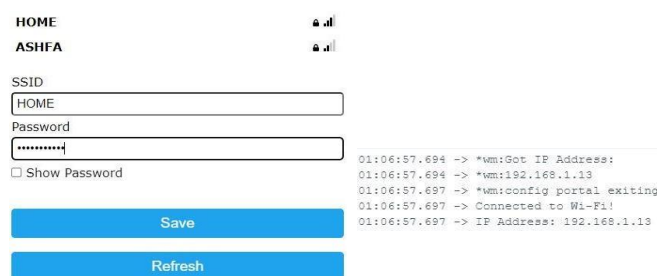
ID	Topic Relay	Status	Time Stamp
21	TA/relay	ON	14/07/2024 21:54
22	TA/relay	OFF	14/07/2024 21:54

23	TA/relay	ON	14/07/2024 21:54
24	TA/relay	OFF	14/07/2024 21:54
25	SK/relay	ON	14/07/2024 21:54
26	SK/relay	OFF	14/07/2024 21:54
27	SK/relay	ON	14/07/2024 21:54
28	SK/relay	OFF	14/07/2024 21:54

3.2.2 Implementasi Protokol Komunikasi MQTT

Wi-Fi Manager diimplementasikan untuk memberikan fleksibilitas dalam konfigurasi jaringan tanpa perlu memodifikasi kode program. Setelah perangkat NodeMCU ESP8266 dinyalakan dan tidak dapat terhubung ke jaringan yang tersimpan sebelumnya, sistem secara otomatis beralih ke mode Access Point dengan SSID "ESP8266" dan membuat portal web captive pada alamat IP default 192.168.4.1.

Melalui portal web ini, pengguna dapat dengan mudah mengkonfigurasi berbagai parameter jaringan, terutama SSID dan password untuk jaringan Wi-Fi yang ingin dihubungkan. Proses ini secara signifikan mengurangi hambatan teknis bagi pengguna yang ingin mengimplementasikan sistem di berbagai lokasi dengan konfigurasi jaringan yang berbeda.



Gambar 10. Connect to Wi-Fi

Gambar 10 menunjukkan antarmuka Wi-Fi Manager yang berhasil melakukan pemindaian terhadap jaringan Wi-Fi di sekitar. Setelah pengguna memasukkan SSID dan Password yang valid, sistem berhasil terhubung dengan jaringan Wi-Fi target dan memperoleh alamat IP 192.168.1.13. Dengan implementasi ini, sistem dapat terhubung dengan jaringan Wi-Fi dan internet tanpa perlu mengubah kode program untuk SSID dan password secara manual.

3.2.3 Implementasi Dashboard Website

Dashboard website berfungsi sebagai antarmuka utama bagi pengguna untuk melakukan monitoring dan kontrol terhadap sistem pencahayaan. Website ini dapat diakses secara publik melalui alamat <https://tugasakhirali2024.my.id/> menggunakan berbagai browser web seperti Chrome, Firefox, Safari, dan Internet

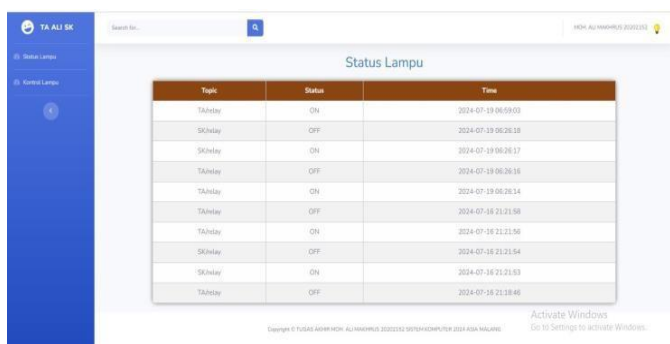
Explorer. Dashboard dirancang dengan mempertimbangkan aspek kegunaan (usability) dan responsivitas, sehingga dapat diakses baik dari perangkat desktop maupun mobile [22].

Struktur dashboard terdiri dari dua komponen utama yang saling terintegrasi, yaitu panel kontrol lampu dan panel monitoring status. Panel kontrol lampu memungkinkan pengguna untuk menyalakan atau mematikan lampu secara langsung melalui antarmuka switch yang tersedia pada dashboard. Sementara itu, panel monitoring status berfungsi untuk menampilkan informasi status terkini dari lampu, seperti kondisi menyala atau mati, serta mencatat riwayat aktivitas perubahan status sebagai bentuk dokumentasi sistem secara real-time.



Gambar 11. Tampilan Dashboard Kontrol Lampu

Gambar 11 menunjukkan dashboard yang berhasil terhubung ke broker MQTT dengan indikator status koneksi "connected". Dashboard ini mampu mendeteksi pemutusan koneksi dan melakukan upaya rekoneksi secara otomatis untuk menjamin ketersediaan layanan. Tombol saklar pada dashboard berfungsi dengan responsif untuk mengontrol lampu, dan status lampu diperbarui secara real-time tanpa latensi yang signifikan.



Gambar 12. Tampilan Dashboard History Status Lampu

Gambar 12 menampilkan riwayat status lampu yang disimpan dalam database. Data status yang dikirimkan dari dashboard kontrol lampu dicatat dengan presisi

waktu dalam database MySQL dan ditampilkan kembali pada panel monitoring. Status ini tetap konsisten bahkan setelah halaman di-refresh, menunjukkan kehandalan sistem penyimpanan data yang diimplementasikan.

3.2.1 Implementasi Keseluruhan Alat

Setelah seluruh komponen elektronik berhasil dirakit dan dikonfigurasi, terbentuklah prototipe sistem kontrol lampu ruangan berbasis Internet of Things sesuai dengan desain yang telah direncanakan. Prototipe ini mengintegrasikan NodeMCU ESP8266, modul RTC DS3231, relay dua channel, dan power supply MB102 dalam satu kesatuan fungsional.



Gambar 13. Rangkaian Keseluruhan

Gambar 13 menunjukkan hasil akhir perakitan sistem yang telah dikemas dalam box pengendali berukuran 20×15×10 cm berbahan plastik. Penataan komponen dirancang dengan mempertimbangkan aspek efisiensi ruang, pendinginan, dan kemudahan akses untuk pemeliharaan. Kabel listrik transparan 2x50 yang digunakan memberikan fleksibilitas dalam penempatan lampu di berbagai lokasi dalam ruangan.

3.3 Analisis dan Pengujian

Sesuai dengan tahapan metode penelitian yang telah diuraikan pada bagian 2.3, analisis dan pengujian sistem dilakukan secara komprehensif terhadap sistem kontrol lampu ruangan berbasis IoT. Pengujian ini mencakup evaluasi kinerja seluruh komponen sistem, meliputi NodeMCU ESP8266, sensor waktu RTC DS3231, modul relay, dashboard website, Wi-Fi Manager, modul power supply MB102, serta protokol komunikasi MQTT sebagai medium interkoneksi data. Berikut adalah hasil dan analisis dari serangkaian pengujian tersebut:

3.3.1 Pengujian Perangkat

Pengujian kendali perangkat NodeMCU ESP8266 dilaksanakan melalui dashboard website yang telah dirancang dengan mengimplementasikan protokol komunikasi MQTT. Dalam pengujian ini, setiap mikrokontroler yang terpasang pada sistem kontrol lampu

dievaluasi kemampuannya dalam menerima dan merespons perintah untuk menyalakan dan mematikan lampu dari jarak jauh. Pengujian dilakukan dalam kondisi lingkungan yang merepresentasikan penggunaan nyata, dengan ruangan berukuran 7 hingga 10 Meter persegi. Tujuan utama dari pengujian ini adalah untuk memverifikasi bahwa komunikasi antara dashboard website dan perangkat berjalan dengan stabilitas dan responsivitas yang memadai dalam berbagai kondisi operasional.



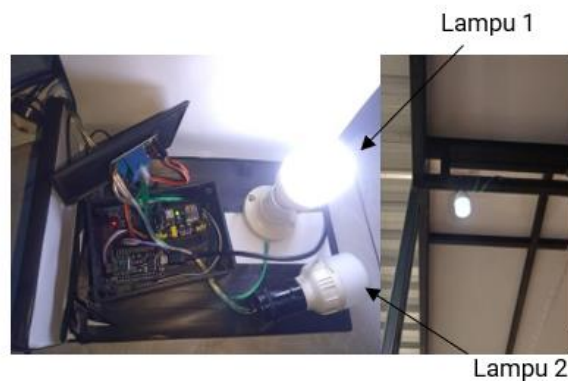
Gambar 14. Data dikirim dari dashboard

Gambar 14 menunjukkan status koneksi "Connected" yang mengindikasikan bahwa dashboard berhasil terhubung dengan server broker MQTT. Status ini akan berubah menjadi "Not Connected" jika koneksi terputus. Pengujian menunjukkan bahwa pengguna dapat mengontrol lampu 1 dan lampu 2 menggunakan tombol switch yang disediakan pada dashboard.

```
11:40:53.756 -> [MQTT] Message received: TA/relay - ON
11:40:53.756 -> TA Relay turned ON
11:40:57.483 -> [MQTT] Message received: TA/relay - OFF
11:40:57.483 -> TA Relay turned OFF
11:41:10.899 -> [MQTT] Message received: SK/relay - ON
11:41:10.899 -> SK Relay turned ON
11:41:13.700 -> [MQTT] Message received: SK/relay - OFF
11:41:13.700 -> SK Relay turned OFF
```

Gambar 15. Pengiriman Data

Gambar 15 memvisualisasikan proses pengiriman data dari dashboard ke mikrokontroler dan respons yang dihasilkan, memverifikasi aliran data dua arah yang berfungsi dengan baik.



Gambar 16. Kondisi Lampu 1 Menyala

Gambar 16 menunjukkan hasil pengujian lampu 1 dalam ruangan terbuka menggunakan lampu 5 watt. Pengujian ini memvalidasi bahwa sistem kontrol dapat mengoperasikan lampu dengan efektif dalam area jangkauan standar. Instalasi menggunakan kabel listrik transparan berukuran 2x50 30yard yang dapat disesuaikan panjangnya memberikan fleksibilitas dalam pemasangan, memungkinkan penempatan pada atap atau dinding di berbagai lantai ruangan.



Gambar 17. Kondisi Lampu 2 Menyala otomatis

Pengujian fitur otomatisasi waktu dilakukan dengan mengatur modul RTC DS3231 untuk menyalakan lampu 2 pada pukul 17.00 WIB, seperti yang ditunjukkan pada Gambar 17. Hasil pengujian mengonfirmasi bahwa fungsi otomatisasi berbasis waktu berjalan sesuai dengan parameter yang ditetapkan. Perlu dicatat bahwa akurasi waktu dari modul RTC sangat bergantung pada sinkronisasi awal dengan sistem komputer pada saat program diunggah.

Berdasarkan serangkaian pengujian, modul RTC DS3231 terbukti mampu mempertahankan ketepatan waktu bahkan setelah sistem mengalami pemutusan daya. Tidak ditemukan deviasi waktu yang signifikan sebelum dan sesudah siklus daya, yang mengindikasikan kinerja modul RTC sesuai dengan spesifikasi yang diharapkan. Keandalan ini dimungkinkan oleh baterai cadangan yang terintegrasi dalam modul, yang menjaga konsistensi

osilator kristal dan sirkuit penyimpanan waktu selama periode tanpa daya eksternal.

3.3.2 Pengujian Protokol Komunikasi MQTT

Pengujian kinerja komunikasi dilakukan terhadap dashboard website kontrol lampu yang diakses melalui alamat <https://tugasakhirali2024.my.id/>. Hasil pengujian menunjukkan bahwa integrasi antara program dan perangkat keras berfungsi dengan optimal. Data yang dikirim ke broker MQTT dan diterima oleh mikrokontroler secara otomatis terekam dalam database untuk keperluan pencatatan dan analisis.

Tabel 2. Pengujian Waktu Respon Perangkat

No	Topic Relay	Status	Time Stamp	Waktu Respon
1	Lampu 1	ON	14/07/2024 21:54	0,0823
2	Lampu 1	OFF	14/07/2024 21:54	0,0592
3	Lampu 1	ON	14/07/2024 21:54	0,0582
4	Lampu 1	OFF	14/07/2024 21:54	0,0756
5	Lampu 2	ON	14/07/2024 21:55	0,0841
6	Lampu 2	OFF	14/07/2024 21:55	0,0754
7	Lampu 2	ON	14/07/2024 21:55	0,0575
8	Lampu 2	OFF	14/07/2024 21:55	0,0891
9	Lampu 2	ON	14/07/2024 21:55	0,0650

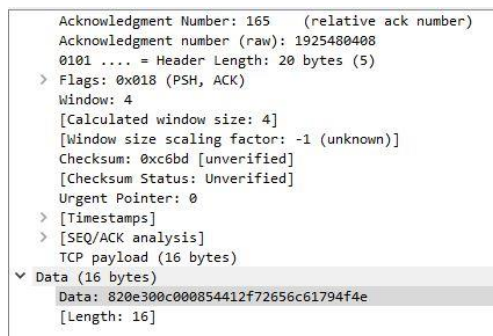
Berdasarkan data pada Tabel 3, analisis terhadap latensi sistem dapat dijabarkan sebagai berikut:

1. Waktu Respons: Kedua relay (untuk Lampu 1 dan Lampu 2) menunjukkan transisi yang cepat antara status ON dan OFF, dengan interval waktu kurang dari 0,1 detik pada semua percobaan.
2. Rata-rata Latensi: Rata-rata waktu respons sistem berada pada rentang 0,05 hingga 0,08 detik, yang mengindikasikan kinerja real-time yang sangat baik untuk aplikasi kontrol rumah tangga.
3. Konsistensi: Kedua relay menunjukkan konsistensi dalam perubahan status, dengan variasi waktu respons yang minimal, menunjukkan stabilitas sistem yang tinggi.
4. Frekuensi Perubahan: Pengujian mencakup beberapa siklus ON dan OFF dalam waktu

singkat, memverifikasi kemampuan sistem untuk menangani perintah berurutan dengan efisien.

Latensi sistem yang terukur dapat dikategorikan sangat rendah dan konsisten, yang mendukung klaim keandalan dalam transmisi status relay. Hasil pengujian kontrol lampu ruangan dengan NodeMCU ESP8266 menunjukkan tingkat keberhasilan 100% dalam pengontrolan lampu melalui dashboard website, yang menegaskan kehandalan implementasi sistem.

3.3.3 Pengujian Bandwidth



Gambar 18. Pengujian bytes pada Wireshark

Gambar 18 menampilkan hasil pengujian konsumsi bandwidth menggunakan aplikasi Wireshark. Berdasarkan data yang diperoleh, beberapa parameter penting dapat diidentifikasi sebagai berikut:

1. Acknowledgment Number: 165, mengindikasikan bahwa penerima telah menerima semua byte hingga nomor urut 164.
2. Header Length: 20 byte, menunjukkan ukuran header TCP tanpa opsi tambahan.
3. Flags: 0x018 (PSH, ACK), mengindikasikan bahwa paket ini menandakan data yang harus segera diproses oleh penerima (Push) dan merupakan bagian dari sesi komunikasi aktif (Acknowledgment).
4. Window Size: 4, menunjukkan ukuran jendela penerimaan yang digunakan dalam transmisi.
5. Payload (Data): 16 byte, ditampilkan dalam format heksadesimal sebagai: 82 0e 30 0c 00 88 54 41 27 25 56 66 17 94 f4 ae.

Pengujian besaran data yang dikirimkan melalui protokol komunikasi MQTT dilakukan dengan memonitor paket data menggunakan aplikasi Wireshark, dengan menerapkan filter pada host alamat broker.emqx.io:8083/mqtt. Untuk menganalisis konsumsi bandwidth secara komprehensif, perlu dipertimbangkan bahwa data payload 16 byte tidak dikirimkan secara

terpisah, melainkan dikemas bersama dengan header dari berbagai lapisan protokol jaringan.

Tabel 3. Estimasi ukuran total satu paket

No	Komponen	Ukuran
1	Ethernet Header	14 byte
2	IP Header	20 byte
3	TCP Header	20 byte
4	TCP Payload	16 byte
	Total	70 byte

Berdasarkan analisis pada tabel 3 ini, untuk mengirimkan 16 byte data payload melalui jaringan TCP/IP, dibutuhkan sekitar 70 byte bandwidth total. Temuan ini mengungkapkan adanya overhead protokol yang cukup signifikan, terutama ketika data dikirimkan dalam potongan-potongan kecil. Untuk meningkatkan efisiensi transmisi data, implementasi sistem dapat memanfaatkan teknik penggabungan data (buffering) agar rasio antara data payload dan overhead protokol menjadi lebih optimal. Dengan pendekatan ini, sistem dapat mengoptimalkan penggunaan bandwidth sambil tetap mempertahankan responsivitas yang tinggi.

4. SIMPULAN

Penelitian ini telah berhasil merancang dan mengimplementasikan sistem kontrol lampu ruangan berbasis Internet of Things (IoT) dengan protokol komunikasi Message Queuing Telemetry Transport (MQTT) menggunakan mikrokontroler NodeMCU ESP8266, modul RTC DS3231, dan Wi-Fi Manager. Sistem menunjukkan performa tinggi dengan waktu respons rata-rata 0,05–0,08 detik dan konsumsi bandwidth yang efisien sekitar 70 byte per paket data. Keandalan sistem dibuktikan dengan kestabilannya dalam berpindah jaringan tanpa gangguan fungsi, serta kemampuan kendali dan monitoring lampu secara real-time melalui dashboard web. Kontribusi utama penelitian ini terletak pada penerapan konfigurasi jaringan yang fleksibel tanpa pemrograman ulang dan integrasi sistem kendali otomatis berbasis waktu serta kendali manual melalui web, yang menjadikannya solusi hemat energi dan mudah digunakan dalam lingkungan rumah tangga, institusi, maupun komersial. Untuk pengembangan selanjutnya, disarankan penambahan fitur keamanan komunikasi MQTT, integrasi sensor arus untuk pemantauan beban listrik, serta pengembangan aplikasi mobile guna mendukung skalabilitas sistem dan memperluas adopsinya di berbagai platform pengguna.

DAFTAR PUSTAKA

- [1] Y. Efendi, "Internet Of Things (Iot) Sistem Pengendalian Lampu Menggunakan Raspberry Pi Berbasis Mobile," *J. Ilm. Ilmu Komput.*, vol. 4, no. 2, pp. 21–27, 2018, doi: 10.35329/jiik.v4i2.41.
- [2] A. Jalil, "Sistem Kendali Perangkat Elektronik Jarak Jauh Berbasis Jaringan Nirkabel Menggunakan Secure Shell (SSH) dan robot Operating System (ROS)," *J. Teknol. Inf. dan Ilmu Komput.*, vol. 7, no. 6, pp. 1205–1212, 2020, doi: 10.25126/jtiik.2020722737.
- [3] A. Setiawan, J. Maulindar, and Nurchim, "Perancangan Sistem Kendali Otomatis Lampu Jalan Berbasis Internet of Things," *INFOTECH J.*, vol. 9, no. 1, pp. 243–251, 2023, doi: 10.31949/infotech.v9i1.5502.
- [4] A. Shafitri, Suhardianto, A. Mashuri, and A. Aditya, "Perancangan Pengendali Lampu Kantor Berbasis Internet of Thing," *PROSISKO J. Pengemb. Ris. dan Obs. Sist. Komput.*, vol. 9, no. 1, pp. 53–59, 2022, doi: 10.30656/prosisko.v9i1.4672.
- [5] M. Y. Iqbar and K. P. Kartika Riyanti, "Rancang Bangun Lampu Portable Otomatis Menggunakan Rtc Berbasis Arduino," *Antivirus J. Ilm. Tek. Inform.*, vol. 14, no. 1, pp. 61–72, 2020, doi: 10.35457/antivirus.v14i1.1115.
- [6] F. Ilhami, P. Sokibi, and A. Amroni, "Perancangan Dan Implementasi Prototype Kontrol Peralatan Elektronik Berbasis Internet of Things Menggunakan Nodemcu," *J. Digit.*, vol. 9, no. 2, p. 143, 2019, doi: 10.51920/jd.v9i2.115.
- [7] A. Herlina, M. I. Syahbana, M. A. Gunawan, and M. M. Rizqi, "Sistem Kendali Lampu Berbasis Iot Menggunakan Aplikasi Blynk 2.0 Dengan Modul Nodemcu Esp8266," *INSANtek*, vol. 3, no. 2, pp. 61–66, 2022, doi: 10.31294/instk.v3i2.1532.
- [8] A. Kurnianto, J. Dedy Irawan, F. X. Ariwibisono, and A. Wardhana, "Penerapan IoT (Internet Of Things) Untuk Controlling Lampu Menggunakan Protokol MQTT Berbasis Web," *JATI (Jurnal Mhs. Tek. Inform.)*, vol. 6, no. 2, p. 1153, 2022, [Online]. Available: <https://www.embedded.com/>
- [9] T. Zulupeker, "WiFiManager: ESP8266 WiFi Connection Manager with Captive Portal," GitHub. Accessed: Jun. 12, 2025. [Online]. Available: <https://github.com/tzapu/WiFiManager>
- [10] U. Ristian, I. Ruslianto, H. Hasfani, and K. Sari, "Perancangan Arsitektur Node Nirkabel dalam Efisiensi Bandwidth Smart Greenhouse Berbasis Protokol MQTT," *J. Edukasi dan Penelit. Inform.*, vol. 9, no. 2, p. 218, 2023, doi:

- 10.26418/jp.v9i2.63885.
- [11] N. R. Ahsy, A. Bhawiyuga, and D. P. Kartikasari, "Implementasi Sistem Kontrol dan Monitoring Smart Home Menggunakan Integrasi Protokol Websocket dan MQTT," *J. Pengemb. Teknol. Inf. dan Ilmu Komput.*, vol. 3, no. 4, pp. 3709–3718, 2019, [Online]. Available: <http://j-ptiik.ub.ac.id>
- [12] A. R. AL TAHTAWI, T. D. HENDRAWATI, A. ABDURRAHIM, and E. ANDIKA, "Perancangan dan Analisis Kinerja Sistem Kontrol dan Penjadwalan Lampu Berbasis IoT," *ELKOMIKA J. Tek. Energi Elektr. Tek. Telekomun. Tek. Elektron.*, vol. 7, no. 3, p. 533, 2019, doi: 10.26760/elkomika.v7i3.533.
- [13] Rochman H, Primananda R, and Nurwasito H, "Sistem Kendali Berbasis Mikrokontroler Menggunakan Protokol MQTT pada Smarthome," *J. Pengemb. Teknol. Inf. dan Ilmu Komput.*, vol. 1, no. 6, pp. 445–455, 2017, [Online]. Available: <https://j-ptiik.ub.ac.id/index.php/j-ptiik/article/view/132>
- [14] Rahmat Irsyada, Muhdlor Auhah Haq, Naila Afina Rohmah, Prima Angga Hadi Saputra, and Roikhatul Jannah, "Implementasi NodeMCU ESP8266 dan Sensor Cahaya Pada Lampu Berbasis Internet Of Things," *J. Ilm. Sist. Inf. dan Ilmu Komput.*, vol. 2, no. 1, pp. 22–32, 2022, doi: 10.55606/juisik.v2i1.514.
- [15] R. Nurul Hidayatullah, N. Ariesanto Ramdhan, and A. Khamid, "Pengembangan Kendali Lampu Menggunakan Mikrokontroler Nodemcu Esp32 Dan Arduino Ide Berbasis Internet of Things (Iot)," *JATI (Jurnal Mhs. Tek. Inform.)*, vol. 8, no. 4, pp. 7762–7767, 2024, doi: 10.36040/jati.v8i4.10461.
- [16] E. T. Alawiah, M. Bayu, and R. Mubaraq, "2897-6517-2-Pb," vol. 4, no. 2, pp. 79–91, 2021.
- [17] D. Rahman, "Pemanfaatan Internet Sebagai Sumber Belajar dan Informasi," *J. Perpust. dan Inf.*, vol. 1, no. 1, pp. 9–14, 2021.
- [18] 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.
- [19] F. Nadhiem, M. D. Ibnugraha, P. and P. Kurniawan, A, "Pengendali Raspberry Pi Jarak Jauh Menggunakan Protokol MQTT Berbasis Web," *e-Proceeding Appl. Sci.*, vol. 9, no. 4, pp. 2056–2063, 2022.
- [20] D. Sadewo, "Analisis Komunikasi data menggunakan Internet Of Things dengan protokol MQTT pada alat Swimming Lap Counter," vol. 11, no. 5, pp. 5654–5658, 2024.
- [21] M. E. Sugiharto and A. S. Budi, "Station Node untuk Rental Sepeda Menggunakan MQTT," *J. Pengemb. Teknol. Inf. dan Ilmu Komput.*, vol. 8, no. 3, pp. 2548–964, 2024, [Online]. Available: <http://j-ptiik.ub.ac.id>
- [22] Z. H. Kudadiri, "Rancang Bangun Aplikasi Manajemen Reservasi Restoran," vol. 21, no. 1, pp. 249–257, 2025.