

## PENERAPAN IOT (INTERNET OF THINGS) UNTUK CONTROLLING LAMPU MENGUNAKAN PROTOKOL MQTT BERBASIS WEB

**Ari Kurnianto, Joseph Dedy Irawan, FX. Ariwibisono**  
Program Studi Teknik Informatika S1, Fakultas Teknologi Industri  
Institut Teknologi Nasional Malang, Jalan Raya Karanglo km 2 Malang, Indonesia  
1818085@scholar.itn.ac.id

### ABSTRAK

Lampu merupakan alat penerangan cahaya yang sering dijumpai dalam ruangan dengan sumber utama energinya ialah menggunakan sumber energi dari listrik. Seiring perkembangan zaman penerapan *Internet of Things* (IoT) tidak hanya untuk kebutuhan dalam skala kecil saja melainkan dalam skala yang lebih luas atau besar sekalipun seperti halnya memberikan kemudahan dalam pengendalian atau kontrol alat melalui jarak jauh menggunakan *internet*. Kebutuhan akan lampu dalam ruangan tentu menjadi persoalan yang penting apalagi jumlah ruangan yang banyak sehingga untuk menghidupkan dan mematikan lampu secara manual tentu dirasa kurang efektif dan juga membutuhkan tenaga untuk bisa menghidupkan atau memamatkannya. Dengan adanya permasalahan tersebut diperlukan *controlling* lampu melalui penerapan protokol MQTT (*Message Queuing Telemetry Transport*) yang dihubungkan ke *broker server* pada MQTT untuk melakukan pengiriman dan penerimaan pesan atau perintah berdasarkan *topic* yang telah ditentukan untuk menyalakan lampu. Pengujian yang dilakukan menggunakan protokol MQTT pada *client* dapat saling berinteraksi satu sama lain antara *publisher* dan *subscriber* melalui *broker* dengan *topic* melalui jaringan *internet* dengan proses pengiriman pesan atau perintah menggunakan protokol MQTT rata-rata waktu yang dibutuhkan untuk menghidupkan (ON) atau mematikan lampu (OFF) yaitu 0.36167 dan 0.42167 detik.

**Kata kunci :** MQTT, Broker, Topic, Controlling, Lampu, IoT (*Internet of Things*).

### 1. PENDAHULUAN

Perkembangan ilmu pengetahuan dan teknologi semakin hari semakin mengalami perubahan terutama di bidang IoT (*Internet of Things*). *Internet of Things* (IoT) sendiri memungkinkan mereka untuk berkomunikasi satu sama lain antar perangkat dengan perangkat lainnya serta saling bertukar data atau pesan agar dapat berkomunikasi satu sama lainnya. Seiring dengan perkembangan zaman penerapan *Internet of Things* (IoT) tidak hanya untuk kebutuhan dalam skala kecil saja melainkan dalam skala yang lebih luas atau besar sekalipun seperti halnya memberikan kemudahan dalam pengendalian atau kontrol alat melalui jarak jauh menggunakan *internet*.

Lampu merupakan alat penerangan cahaya yang sering dijumpai di dalam ruangan dengan sumber utama energinya ialah menggunakan sumber energi dari listrik yang dimana peran lampu sangatlah penting dalam kehidupan sehari-hari sehingga hampir semua ruangan menggunakan cahaya penerangan dari lampu hal tersebut tentunya menimbulkan suatu permasalahan dimana untuk menghidupkan atau mematikan lampu sebagian orang masih menggunakan cara yang manual yaitu dengan cara menghidupkan dan mematikan saklar yang terdapat pada lampu secara satu persatu atau bergantian. Hal tersebut dirasa kurang efektif jika melihat kondisi lampu dengan jumlah yang cukup banyak sehingga membutuhkan waktu dan juga tenaga untuk bisa menghidupkan atau mematikan lampu seperti kendali jarak jauh terutama pada peralatan elektronik rumah tangga misalnya saja lampu, hal tersebut sangatlah

berguna dan sekaligus mempermudah pekerjaan seperti untuk mengaktifkan atau menonaktifkan perangkat elektronik lampu yang menggunakan energi listrik agar dapat mengontrol dan mengetahui kondisi lampu mana saja yang menyala atau mati [1].

Dengan adanya permasalahan tersebut, untuk mengatasinya memerlukan sebuah alat yang dapat melakukan *controlling* seperti halnya menyalakan atau mematikan lampu pada perangkat elektronik dengan menggunakan protokol MQTT (*Message Queuing Telemetry Transport*) yang dimana merupakan sebuah protokol yang berjalan pada layer aplikasi dengan mekanisme *publish* dan *subscribe* yang dapat melakukan pengiriman dan penerimaan pesan melalui *controlling* ataupun *monitoring* berdasarkan *topic* yang telah ditentukan sesuai dengan keinginan pengguna dimana pada protokol jenis ini bekerja dengan *bandwidth* yang rendah atau lebih sedikit dibandingkan dengan protokol lainnya sehingga dapat bekerja dengan baik di dalam sumber daya yang terbatas [2].

Alat ini nantinya akan dirancang dengan tujuan untuk menghidupkan dan mematikan lampu menggunakan protokol MQTT yang dapat dikontrol dalam jarak jauh melalui penerapan IoT (*Internet of Things*).

### 2. TINJAUAN PUSTAKA

#### 2.1 Penelitian Terkait

Menurut penelitian Hasiholan, Primananda, dan Amron dengan judul penelitian "Implementasi

Konsep *Internet of Things* pada Sistem *Monitoring Banjir Menggunakan Protokol MQTT*". Pada penelitian ini membahas mengenai penerapan konsep *publish/subscribe* menggunakan protokol MQTT untuk sistem *monitoring* banjir. Dimana terdapat *broker* yang digunakan ialah *mosquitto* dan *subscribe* sehingga dapat menampilkan data tersebut ke dalam sebuah aplikasi berbasis *web* dengan pengukuran sensor berupa digit angka yang dimana sensor dan mikrokontroler diletakkan di titik daerah banjir sehingga sensor akan mengirimkan data menggunakan protokol MQTT melalui jaringan *broker* yang aktif di perangkat PC maupun Laptop dan *broker* akan mengirimkan data tersebut ke halaman *web* yang dapat di akses [3].

Menurut penelitian Sutarti, Anharudin, dan Rosadi dengan judul penelitian "*Prototype Sistem Pendeteksi Banjir Menggunakan NodeMCU dan Protokol MQTT Berbasis Internet of Things*". Pada penelitian ini membahas sistem *monitoring* banjir dengan memanfaatkan sensor ultrasonik dan juga protokol MQTT kali ini diimplementasikan untuk sistem *monitoring* perangkat IoT (*Internet of Things*) dengan menggunakan NodeMCU untuk mengukur jarak ketinggian permukaan air yang dimana data hasil pembacaan akan menampilkan sebuah notifikasi pada layar *Telegram* yang ada pada *smartphone* sehingga akan memudahkan masyarakat untuk mengetahui informasi dan juga sebagai antisipasi datangnya banjir [4].

Menurut penelitian Akh. Fahrudin dengan judul "*Sistem Monitoring dan Kontroling Pada Smart Building Dengan Penerapan IOT (Internet of Things)*". Pada penelitian ini membahas terkait akan permasalahan pada gedung maupun pabrik yang dimana sistem kontrol lampu yang diterapkan masih menggunakan cara yang manual sehingga kurang optimal dan juga membutuhkan waktu yang lama dalam *controlling* serta *monitoring* dengan adanya penerapan IoT (*Internet of Things*) memudahkan dalam kontrol dan monitor lampu, suhu dan kelembapan dengan jarak jauh [5].

## 2.2 IoT (Internet of Things)

*Internet of Things* (IoT) merupakan sebuah sistem yang saling terhubung atau gabungan dari beberapa alat mikrokontroler yang dirangkai satu sama lain dan dapat saling berkomunikasi serta bertukar data secara otomatis melalui *internet* atau jaringan *wireless*. Pada dasarnya IoT (*Internet of Things*) menghubungkan semua perangkat yang memiliki tombol *on* ataupun *off* yang terhubung ke dalam jaringan *internet*. Perangkat IoT sendiri di desain untuk penggunaan daya yang kecil sehingga mampu menjalankan perintah-perintah seperti halnya *smart home* ataupun lainnya [6].

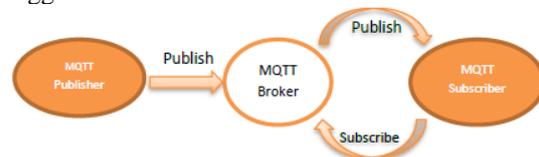
## 2.3 NodeMCU ESP8266

NodeMCU ESP8266 merupakan salah satu *device opensource* yang digunakan pada *platform* IoT

dimana memiliki ukuran *board* yang kecil dengan panjang 4.83 cm, lebar 2.54 cm, dan berat sekitar 7 gram. Selain ukurannya yang kecil NodeMCU ESP8266 sendiri sudah dilengkapi fitur *wifi* dan *firmware* yang tentunya juga bersifat *opensource* dari segi biaya penggunaan NodeMCU ESP8266 juga lebih hemat dibandingkan dengan menggunakan *Arduino Uno* yang belum memiliki modul *wifi* dan hanya berupa mikrokontroler belum berbasis IoT sehingga perlu adanya perangkat tambahan seperti modul *wifi shield* agar bisa terhubung ke dalam sistem IoT [7].

## 2.4 MQTT (Message Queuing Telemetry Transport)

*Message Queuing Telemetry Transport* (MQTT) merupakan jenis protokol jaringan pada IoT (*Internet of Things*) yang dimana berfungsi sebagai komunikasi yang bersifat *machine to machine* atau M2M dapat bekerja pada *bandwidth* yang rendah sangat cocok untuk digunakan dalam *internet of things* (IoT) yang dapat bekerja dalam energi dan media penyimpanan yang minimum serta sifatnya yang ringan dalam pengiriman pesan (*lightweight message*) pada *header* berukuran kecil yaitu 2 bytes. MQTT pertama kali dikembangkan oleh Andy Stanford-Clark dari IBM dan Arlen Nipper dari Arcom pada tahun 1999 dan dengan tujuan untuk meminimalisir daya dari proses serta meminimalkan penggunaan *bandwidth*.



Gambar 1. MQTT (*Message Queuing Telemetry Transport*)

Pada protokol MQTT (*Message Queuing Telemetry Transport*) sendiri menggunakan metode komunikasi *publish/subscribe* yang dimana *publish* berfungsi sebagai pengirim pesan dan *subscribe* sebagai penerima pesan menggunakan protokol MQTT sendiri dimana dalam MQTT terdapat *broker* yang berfungsi sebagai *topic* yang dikirimkan oleh *publisher* untuk di teruskan ke *subscribe* berdasarkan permintaan dari pengguna [8].

## 2.5 Quality of Service (QoS)

*Quality of Service* (QoS) merupakan kualitas pengirim paket atau pesan yang dikirim oleh *publisher* ke *subscriber* atau sebaliknya [9]. Terdapat 3 (tiga) level *Quality of Service* (QoS) pada MQTT yaitu:

### 1. QoS Level 0

Merupakan level terendah yaitu tidak ada jaminan akan paket atau pesan sampai ke *subscriber* jika terjadi kegagalan maka tidak ada pengiriman ulang pesan. Pesan hanya akan di kirimkan sekali atau tidak sama sekali.

2. QoS Level 1

Merupakan level menengah yaitu pesan dijamin sampai minimal 1 kali ke *subscriber* yang sedang meng-*subscribe*.

3. QoS Level 2

Merupakan level paling tertinggi yaitu pesan atau paket dijamin sampai tepat 1 kali ke *subscriber* yang sedang meng-*subscribe*.

2.6 MQTT Broker

*Broker* merupakan salah satu penyedia layanan *server* yang sangat penting pada jaringan protokol MQTT yang dimana perangkat (*device*) pada MQTT membutuhkan *broker* untuk bisa saling berkomunikasi satu sama lain. Beberapa contoh *broker* MQTT yang ada seperti *Mosquito*, *HiveMQ*, *Erlang MQTT*, *Reyax*, dan lain-lain. terhubung, sehingga perangkat yang tidak saling kenal akan dihubungkan ke *broker* melalui *internet* agar bisa saling berkomunikasi, berkiriman pesan ataupun berupa perintah tertentu ke *subscriber*.

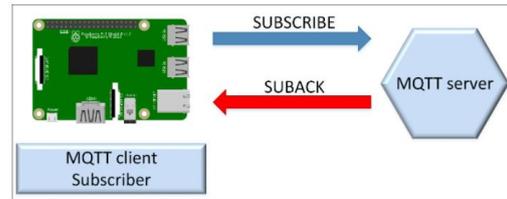


Gambar 2. MQTT broker

Pada setiap pengiriman data antar *publisher* dan *subscribe* harus memiliki suatu *topic* yang mana bertujuan agar data atau pesan yang ada pada *topic* tersebut bisa dikirimkan ke *subscriber* yang menginginkannya sehingga *publisher* dapat mengirimkan pesan berupa *topic* ke MQTT *broker* kemudian pada MQTT *broker* akan menerima dan memfilter *topic* tersebut serta mengirimkannya ke *subscriber* yang dikehendaki oleh pengguna sesuai dengan *topic* yang diminta [10].

2.7 Komunikasi MQTT Client dan Broker

Komunikasi pada MQTT merupakan cara agar *publisher* dan *subscriber* bisa saling bertukar informasi melalui *broker* dengan menggunakan paket kontrol MQTT yang umumnya sering ditemui seperti: *CONNECT*, *CONNACK*, *PUBLISH*, *PUBACK*, *SUBSCRIBE*, *SUBACK* dan *DISCONNECT*. Komunikasi awal diawali dari *client (subscriber)* yang mengirimkan perintah *CONNECT* kontrol paket pada *broker*. Kemudian *broker* akan mengecek dengan membalas perintah kontrol paket yaitu *CONNACK* mengirimkan respon pada *client* dan kode status sehingga ketika sudah terhubung, maka *broker* akan terus membuka koneksi terus menerus semalam tidak ada perintah untuk menutup koneksinya (*disconnect*).



Gambar 3. Komunikasi MQTT

Setelah koneksi berhasil dibangun antara MQTT *client* dan *broker*, maka *broker* akan menjaga koneksi agar tetap terbuka sampai *client* mengalami *loss connection* atau mengirimkan perintah kontrol paket berupa *DISCONNECT* ke *server* untuk menutup atau menghentikan koneksi [11].

2.8 Relay

*Relay* merupakan komponen elektronika berupa saklar elektronik yang digerakkan dengan menggunakan arus listrik. Dimana prinsip *relay* sendiri adalah ketika *solenoid* dialiri arus listrik tuas akan tertarik karena adanya gaya *magnet* sehingga kontak saklar akan menutup. Sehingga pada saat arus listrik dihentikan maka secara otomatis gaya *magnet* akan hilang dan tuas akan kembali ke posisi semula dengan hal itu maka kontak saklar akan kembali terbuka [12].

2.9 Web

*Web* merupakan salah satu layanan *internet* yang umum dijumpai dan juga paling populer sehingga menjadikan *web* sebagai media penyampaian informasi, komunikasi, atau transaksi, dan masih banyak lagi kegunaan *web* lainnya. Penggunaan *web* memungkinkan para *user* mengakses informasi dan data lebih efektif serta efisien dengan menggunakan protokol HTTP (*Hypertext Transfer Protocol*) yang dimana menggunakan sistem *client-server* sehingga penyedia *web server* akan memberikan layanan HTTP (*Hypertext Transfer Protocol*) sesuai dengan permintaan *client* [13].

2.10 Adafruit.IO

Adafruit.IO merupakan sebuah layanan *could* berbasis *website* yang menyediakan fitur-fitur yang dibutuhkan seperti *monitoring* dan juga *controlling* untuk pengembangan sistem IoT yang dapat terhubung langsung melalui *internet*, yang mana berfungsi untuk menyimpan dan mengambil data lebih dari satu.

Pada Adafruit.IO juga menyajikan data secara *real-time* dan *online* untuk membuat sebuah proyek yang langsung tersambung ke *internet* untuk membaca data sebuah sensor, motor kontrol, dan lain-lain serta menyediakan layanan *broker* MQTT *server* IoT untuk mengendalikan NodeMCU ESP8266 dalam jarak jauh dengan menggunakan mekanisme *publish* dan *subscribe* [14].

**2.11 Sensor Cahaya LDR (Light Dependent Resistor)**

Sensor LDR atau modul sensor cahaya LDR (Light Dependent Resistor), merupakan jenis sensor yang digunakan untuk mendeteksi adanya intensitas cahaya di dalam ruangan ataupun luar ruangan. Pada sensor jenis ini biasanya di gunakan untuk menyalakan lampu secara otomatis melalui sensor kecil yang terdapat pada ujung komponen alat sensor cahaya LDR (Light Dependent Resistor) [15].

**2.12 PHP (Hypertext Preprocessor)**

Hypertext Preprocessor (PHP) merupakan bahasa pemrograman yang mana seringkali digunakan pada pemrograman web dan di kombinasikan dengan bahasa pemrograman lainnya seperti html, dan lain-lain. Pmerograman bahasa PHP (Hypertext Preprocessor) tergolong jenis pemrograman web yang bersifat dinamis sehingga menghasilkan website yang bisa berubah-ubah tampilan atau kontennya tanpa harus merubah isi program secara manual. Kemampuan serta fitur dari bahasa pemrograman PHP (Hypertext Preprocessor) sendiri banyak digunakan dalam mendukung basis data (database) seperti MSSQL, MYSQL, Oracle, dan sebagainya [16].

**2.13 Visual Studio Code**

Visual Studio Code (VS Code) merupakan sebuah teks editor yang berfungsi sebagai platform untuk mengembangkan sebuah aplikasi atau program. Pada teks editor sendiri mendukung bahasa pemrograman seperti JavaScript, Typescript, Node.js serta bahasa pemrograman lainnya dengan bantuan plugin dari marketplace VS Code seperti (C++, C#, Python, Go, Java, dan lain-lain) yang dimana VS Code sendiri bersifat open source artinya kode sumbernya dapat dilihat dan juga dapat ikut serta dalam kontribusi untuk pengembangannya. Hal ini juga membuat VS Code menjadi favorit khususnya pengembangan berupa aplikasi dan lain-lain [17].

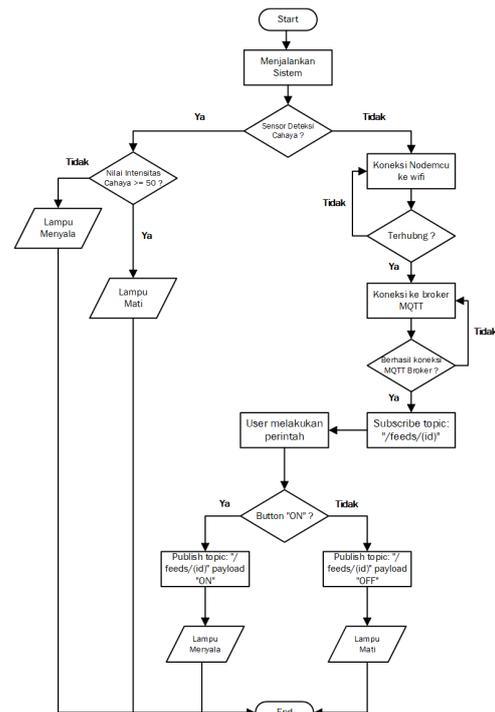
**2.14 XAMPP Server**

XAMPP Server merupakan paket instalasi yang terdiri dari Apache, PHP, dan MySQL yang berfungsi untuk membangun sebuah website dimana XAMPP sendiri berperan sebagai server atau penyediaan layanan berbasis localhost yang dapat diakses melalui alamat local pada komputer. Dengan menggunakan aplikasi XAMPP memudahkan dalam pengembangan pembuatan aplikasi seperti website ataupun aplikasi lainnya yang memerlukan layanan XAMPP [18].

**3. METODE PENELITIAN**

**3.1 Flowchart Sitem**

Dalam penelitian ini, terdapat suatu proses yang berisi flowchart sistem yang dibangun. Berikut dapat dijelaskan proses flowchart sistem.



Gambar 4. Flowchart sistem

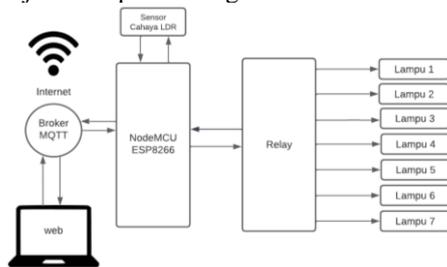
Pada Gambar 4 merupakan flowchart sistem yang mana pada proses pertama yaitu menjalankan sistem terlebih dahulu. Kemudian masuk ke bagian decision pada proses ini terdapat dua proses yaitu sensor LDR dan controlling lampu dengan MQTT. Jika sensor cahaya LDR mendeteksi cahaya lebih dari 50 dimana nilai intensitas cahaya yang masuk lebih dari atau sama dengan 50 ( $\geq 50$ ) ini merupakan nilai mapping (kalibrasi) terhadap alat ukur Lux Meter sehingga lampu akan mati, akan tetapi jika sensor cahaya LDR mendeteksi nilai intensitas cahaya kurang dari 50 maka lampu akan menyala atau aktif.

Kemudian apabila jika sensor cahaya tidak mendeteksi cahaya maka akan masuk ke proses yang satunya yaitu controlling lampu dengan MQTT dimana pada tahap pertama merupakan proses koneksi NodeMCU ke wifi melalui internet. Jika sudah terkoneksi maka NodeMCU akan terhubung langsung ke server broker MQTT namun jika proses tersebut gagal akan terus di ulang sampai berhasil terhubung ke server broker MQTT. Masuk pada proses subscribe topic dimana topic yang telah dibuat sebelumnya yaitu "/>

button dalam kondisi OFF (Tidak) maka sama halnya seperti proses sebelumnya broker akan melakukan proses pengiriman dari subscriber ke publisher dan akan mengontrol relay untuk melakukan perintah mematikan lampu.

### 3.2 Block Diagram Sistem

Dalam penelitian ini, terdapat suatu proses yang berisi diagram blok sistem yang dibangun. Berikut dapat dijelaskan proses diagram blok sistem.

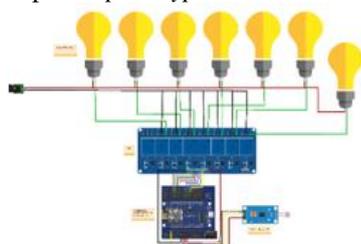


Gambar 5. Blok diagram sistem

Pada Gambar 5 merupakan diagram blok sistem yang memperlihatkan terdapat tujuh buah lampu yang terhubung melalui relay yang dimana relay sendiri berfungsi sebagai saklar untuk menghidupkan atau mematikan arus listrik menuju ke lampu dan sensor cahaya LDR yang dimana ketika sensor mendeteksi adanya intensitas cahaya lampu secara otomatis akan menyala (ON) atau mati (OFF) sesuai dengan nilai intensitas cahaya yang telah ditentukan. Kemudian NodeMCU ESP8266 akan berfungsi sebagai penghubung antara alat mikrokontroler dengan user menggunakan web sebagai interface controlling nyala lampu yang dihubungkan melalui internet agar user dapat mengontrol lampu mana saja yang ingin dinyala matikan menggunakan protokol MQTT. Protokol MQTT sendiri akan terhubung melalui broker sehingga publisher dan subscriber dapat saling terhubung satu sama lain yang nantinya ketika publisher atau user mengirimkan sebuah perintah atau pesan melalui broker maka broker MQTT akan mengirimkan perintah tersebut untuk menyalakan atau mematikan lampu (ON/OFF) dan relay akan secara otomatis menyalakan lampu mana yang di minta oleh user atau publisher melalui web.

### 3.3 Desain Rangkaian Alat

Dalam penelitian ini, terdapat suatu proses yang berisi prototype desain sistem yang dibangun. Berikut dapat dijelaskan proses prototype desain sistem.



Keterangan:

VCC ■ GND ■ Data ■

Gambar 6. Desain rangkaian alat

Pada Gambar 6 merupakan desain rancangan alat yang dibuat dimana terdapat lampu led yang dihubungkan ke relay. Relay yang terhubung ke lampu led akan di hubungkan lagi ke NodeMCU dengan shield board untuk koneksi antara relay dan lampu dan juga terdapat sensor cahaya LDR yang nantinya berfungsi untuk menyalakan lampu secara otomatis.

## 4. HASIL DAN PEMBAHASAN

### 4.1 Implementasi Sistem

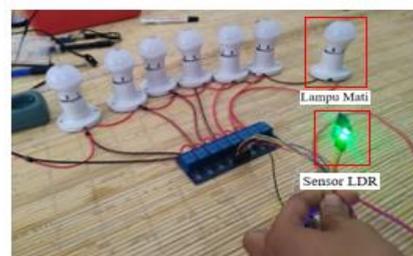
Implementasi sistem merupakan proses penerapan rancangan sistem yang telah dibuat. Ada 2 tahap implementasi yaitu, implementasi pada software dan hardware. Untuk software pada aplikasi berbasis web untuk controlling lampu menggunakan protokol MQTT yang mana pada web tersebut bertujuan untuk dapat menghidupkan atau mematikan saklar pada relay yang dihubungkan ke NodeMCU ESP8266 dengan lampu melalui broker MQTT. Dan untuk hardware pada tahap implementasi hardware ini, alat yang akan dibuat terdiri dari enam buah lampu yang akan di kontrol melalui protokol MQTT dan satu buah lampu lainnya untuk menyalakan dan mematikan lampu secara otomatis menggunakan sensor cahaya LDR.

### 4.2 Pengujian Alat

Pada pengujian alat ini dilakukan dengan cara menguji alat, untuk mengetahui apakah alat ini dapat berfungsi secara baik atau tidak pada saat sistem dijalankan.

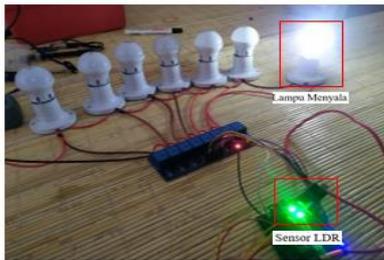
### 4.3 Pengujian Sensor Cahaya LDR

Pada pengujian sensor cahaya LDR dilakukan dengan cara menampilkan nilai intensitas cahaya yang diterima oleh sensor cahaya LDR untuk mengetahui kondisi cahaya gelap atau terang agar sensor dapat menyalakan lampu secara otomatis. Pada proses pengujian ini membandingkan nilai pengukuran standar menggunakan dari alat Lux Meter dengan sensor cahaya LDR.



Gambar 7. Pengujian sensor ldr pada cahaya terang

Pada Gambar 7 merupakan kondisi dimana ketika intensitas cahaya terang lampu sensor ldr akan menyala satu dan lampu led secara otomatis akan berada dalam kondisi mati atau tidak aktif.



Gambar 8. Pengujian sensor ldr pada cahaya gelap  
 Pada Gambar 8 merupakan kondisi dimana ketika kondisi cahaya cukup terang, lampu sensor LDR akan menyala dua dan lampu led secara otomatis dalam kondisi hidup atau menyala.

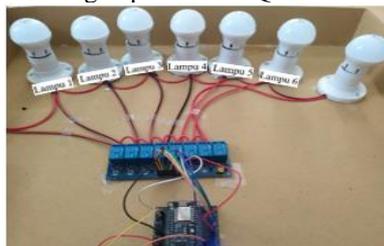
Tabel 1. Hasil pengujian sensor cahaya

No	Waktu (WIB)	Kondisi Cahaya	Sensor LDR	Lux (Lx)	Errorr (%)
1	06.00	Terang	56	56	0%
2	09.00	Terang	68	73	6.84%
3	12.00	Terang	87	90	3.33%
4	16.00	Terang	44	48	8.33%
5	18.00	Gelap	32	30	6.67%
Rata-rata					5.034%

Dari Tabel 1. memperlihatkan bahwa nilai rata-rata error dari hasil pengujian sensor cahaya dengan alat pembanding Lux Meter menghasilkan nilai *error* sebesar 5.034%. Sehingga hal tersebut dapat dikatakan akurasi yang dimiliki oleh sensor cahaya LDR terbilang cukup baik karena dari rata-rata nilai *error* tidak lebih dari 10%.

#### 4.4 Pengujian Lampu Led

Pada pengujian lampu led dilakukan dengan cara mengetahui waktu tunggu (respon) nyala lampu pada masing-masing lampu ketika dinyalakan melalui aplikasi *web* dengan protokol MQTT.



Gambar 9. Pengujian lampu led

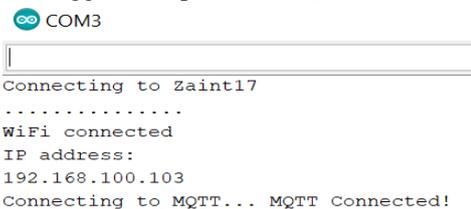
Pada Gambar 9 pengujian lampu led terdapat enam buah lampu led dapat menyala secara baik. Untuk hasil pengujian lampu pada protokol MQTT dilakukan dengan cara menghitung berapa lama waktu yang dibutuhkan tiap masing-masing lampu agar bisa menyala dengan mengontrol alat melalui *web* yang sudah terhubung ke *broker* pada protokol MQTT.

Tabel 2. Hasil pengujian lampu led

No	Uji Coba	Kondisi	Waktu Tunggu (s)	Hasil
1	Lampu 1	ON	0.33	Sesuai
		OFF	0.43	
2	Lampu 2	ON	0.39	Sesuai

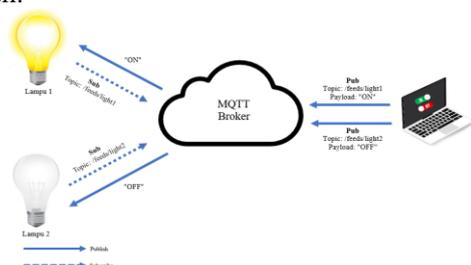
		OFF	0.41	
3	Lampu 3	ON	0.36	Sesuai
		OFF	0.40	
4	Lampu 4	ON	0.38	Sesuai
		OFF	0.42	
5	Lampu 5	ON	0.36	Sesuai
		OFF	0.45	
6	Lampu 6	ON	0.35	Sesuai
		OFF	0.42	

Pada Tabel 2 memperlihatkan hasil pengujian lampu led pada saat ON dan OFF terlihat waktu tunggu dari masing-masing nyala pada lampu memiliki selisih waktu yang berdekatan dan dari hasil pengujian yang telah dilakukan untuk menyalakan lampu menggunakan protokol MQTT membutuhkan waktu sekian detik untuk proses *controlling* lampu ketika menghidupkan dan mematikan lampu yang dimana memberikan hasil sesuai dan telah dilakukan pada uji coba alat yang dimana proses ON dan OFF tersebut menggunakan protokol MQTT.



Gambar 10. Koneksi *client* dengan *mqtt broker*

Pada Gambar 10 merupakan tampilan pada *serial monitor* yang mana koneksi ke MQTT telah berhasil terhubung ke *broker* melalui jaringan *internet*. Proses terhubung ke *broker* membutuhkan waktu sekian detik untuk bisa koneksi ke antara *client* dan *broker*, apabila *client* tidak berhasil terhubung ke *broker* MQTT maka koneksi akan terus diulang oleh *client* sampai mendapatkan *feedback* berupa "MQTT Connected!" dalam hal ini koneksi ke *broker* sudah berhasil.



Gambar 11. Proses alur *controlling* lampu mqtt

Pada Gambar 11 merupakan proses alur sebuah MQTT untuk *controlling* lampu dimana *user* mempublish ke *broker* MQTT dengan *topic* "/feeds/light1" untuk lampu 1 dan "/feeds/light2" untuk lampu 2 dan seterusnya. Kemudian terdapat *payload* atau pesan yang dikirimkan pada *topic* berupa ON dan OFF untuk menyalakan dan mematikan lampu, pada proses ini *publisher* mengirimkan sebuah pesan atau *payload* tersebut ke *subscriber* melalui *broker* MQTT yaitu *io.adafruit.com*. Kemudian pada *broker* MQTT perintah yang dikirimkan oleh *user* (*publisher*) akan di

proses untuk diteruskan ke *subscriber* yang mempunyai kesamaan *topic* dengan *publisher*, jika *topic* pada *subscriber* tidak sesuai dengan *publisher* maka *broker* tidak akan meneruskan perintah tersebut ke *subscriber* sampai *topic* tersebut sesuai dengan yang ada pada *publisher* sehingga pesan atau perintah tersebut akan diterima oleh *subscriber* dan lampu akan menyala pada perintah ON dan lampu akan mati pada perintah OFF.

Tabel 3. Hasil pengujian mqtt

No	Skenario	Pengamatan	Hasil	
			Sesuai	Tidak
1	Menghubungkan <i>client</i> menggunakan NodeMCU ESP8266 ke <i>broker</i> MQTT melalui jaringan internet dengan kondisi <i>wifi</i> di <i>off</i> dan di <i>on</i> lagi untuk skema gangguan pada jaringan.	<i>Client</i> berhasil terhubung ke <i>wifi</i> dan berhasil terkoneksi ke <i>broker</i> dengan kondisi <i>connecting</i> dan <i>reconnect</i> pada saat mengalami gangguan	Ya	
	<i>Publish</i> pada <i>Topic</i> : “ <i>feeds/light1</i> ” dengan <i>message</i> “ON” kemudian “OFF”	<i>Publish</i> pada <i>Topic</i> : “ <i>feeds/light1</i> ” = “ON” <i>relay 1</i> hidup, “ <i>feeds/light1</i> ” = “OFF” <i>relay 1</i> mati.		
2	<i>Subscribe</i> pada <i>Topic</i> : “ <i>feeds/light1</i> ”	<i>Subscribe</i> pada <i>Topic</i> : “ <i>feeds/light1</i> ” lampu akan “ON” dan “OFF” sesuai perintah dari <i>publisher</i> .	Ya	
	<i>Publish</i> pada <i>Topic</i> : “ <i>feeds/light2</i> ” dengan <i>message</i>	<i>Publish</i> pada <i>Topic</i> : “ <i>feeds/light2</i> ” = “ON” <i>relay 2</i> hidup, “ <i>feeds/light1</i> ”		
3	<i>Publish</i> pada <i>Topic</i> : “ <i>feeds/light2</i> ” dengan <i>message</i>	<i>Publish</i> pada <i>Topic</i> : “ <i>feeds/light2</i> ” = “ON” <i>relay 2</i> hidup, “ <i>feeds/light1</i> ”	Ya	

No	Skenario	Pengamatan	Hasil	
			Sesuai	Tidak
	“ON” kemudian “OFF”	= “OFF” <i>relay 2</i> mati.		
	<i>Subscribe</i> pada <i>Topic</i> : “ <i>feeds/light2</i> ”	<i>Subscribe</i> pada <i>Topic</i> : “ <i>feeds/light2</i> ” lampu akan “ON” dan “OFF” sesuai perintah dari <i>publisher</i> .	Ya	
4	<i>Publish</i> pada <i>Topic</i> : “ <i>feeds/light3</i> ” dengan <i>message</i> “ON” kemudian “OFF”	<i>Publish</i> pada <i>Topic</i> : “ <i>feeds/light3</i> ” = “ON” <i>relay 3</i> hidup, “ <i>feeds/light1</i> ” = “OFF” <i>relay 3</i> mati.	Ya	
	<i>Subscribe</i> pada <i>Topic</i> : “ <i>feeds/light3</i> ”	<i>Subscribe</i> pada <i>Topic</i> : “ <i>feeds/light3</i> ” lampu akan “ON” dan “OFF” sesuai perintah dari <i>publisher</i> .		
5	<i>Publish</i> pada <i>Topic</i> : “ <i>feeds/light4</i> ” dengan <i>message</i> “ON” kemudian “OFF”	<i>Publish</i> pada <i>Topic</i> : “ <i>feeds/light4</i> ” = “ON” <i>relay 4</i> hidup, “ <i>feeds/light4</i> ” = “OFF” <i>relay 4</i> mati.	Ya	
	<i>Subscribe</i> pada <i>Topic</i> : “ <i>feeds/light4</i> ”	<i>Subscribe</i> pada <i>Topic</i> : “ <i>feeds/light4</i> ” lampu akan “ON” dan “OFF” sesuai perintah dari <i>publisher</i> .		
6	<i>Publish</i> pada <i>Topic</i> : “ <i>feeds/light5</i> ” dengan <i>message</i>	<i>Publish</i> pada <i>Topic</i> : “ <i>feeds/light5</i> ” = “ON” <i>relay 5</i> hidup, “ <i>feeds/light5</i> ”	Ya	

No	Skenario	Pengamatan	Hasil	
			Sesuai	Tidak
7	“ON” kemudian “OFF”	” = “OFF” <i>relay</i> 5 mati.		
	Subscribe pada <i>Topic</i> : “ <i>feeds/light5</i> ”	Subscribe pada <i>Topic</i> : “ <i>feeds/light5</i> ” lampu akan “ON” dan “OFF” sesuai perintah dari <i>publisher</i> .	Ya	
	Publish pada <i>Topic</i> : “ <i>feeds/light6</i> ” dengan <i>message</i> “ON” kemudian “OFF”	Publish pada <i>Topic</i> : “ <i>feeds/light6</i> ” = “ON” <i>relay</i> 6 hidup, “ <i>feeds/light6</i> ” = “OFF” <i>relay</i> 6 mati.	Ya	
	Subscribe pada <i>Topic</i> : “ <i>feeds/light6</i> ”	Subscribe pada <i>Topic</i> : “ <i>feeds/light6</i> ” lampu akan “ON” dan “OFF” sesuai perintah dari <i>publisher</i> .	Ya	

Pada Tabel 3 memperlihatkan hasil pengujian pada mwt yang perintah yang dikirimkan oleh *user* (*publisher*) akan di proses untuk diteruskan ke *subscriber* yang mempunyai kesamaan *topic* dengan *publisher*, jika *topic* pada *subscriber* tidak sesuai dengan *publisher* maka *broker* tidak akan meneruskan perintah tersebut ke *subscriber* sampai *topic* tersebut sesuai dengan yang ada pada *publisher* sehingga pesan atau perintah tersebut akan diterima oleh *subscriber* dan lampu akan menyala pada perintah ON dan lampu akan mati pada perintah OFF.

#### 4.5 Pengujian Aplikasi Web

Pada pengujian aplikasi *web* dilakukan dengan cara mengetahui kondisi tombol *button* ON atau OFF sudah berjalan dengan baik atau tidak.



Gambar 12. Tampilan *controlling* aplikasi *web*

Pada Gambar 12 merupakan tampilan halaman *web controlling* lampu menggunakan *web*.

Tabel 4. Hasil pengujian *controlling* lampu menggunakan *web*

No	Skenario	Kondisi perangkat dan aplikasi	Hasil	
			Sesuai	Tidak
1	Menyalakan Lampu 1 melalui <i>button</i> pada halaman <i>web</i> .	Lampu 1 menyala dengan indikator <i>button</i> sebelumnya berwarna merah berubah menjadi warna hijau.	Ya	
2	Mematikan Lampu 1 melalui <i>button</i> pada halaman <i>web</i> .	Lampu 1 mati dengan indikator <i>button</i> sebelumnya berwarna hijau berubah menjadi warna merah.	Ya	
3	Menyalakan Lampu 2 melalui <i>button</i> pada halaman <i>web</i> .	Lampu 2 menyala dengan indikator <i>button</i> sebelumnya berwarna merah berubah menjadi warna hijau.	Ya	
4	Menyalakan Lampu 2 melalui <i>button</i> pada halaman <i>web</i> .	Lampu 2 mati dengan indikator <i>button</i> sebelumnya berwarna hijau berubah menjadi warna merah.	Ya	
5	Menyalakan Lampu 3 melalui <i>button</i> pada halaman <i>web</i> .	Lampu 3 menyala dengan indikator <i>button</i> sebelumnya berwarna merah berubah menjadi	Ya	

No	Skenario	Kondisi perangkat dan aplikasi	Hasil	
			Sesuai	Tidak
		warna hijau.		
6	Mematikan Lampu 3 melalui <i>button</i> pada halaman <i>web</i> .	Lampu 3 mati dengan indikator <i>button</i> sebelumnya berwarna hijau berubah menjadi warna merah.	Ya	
7	Menyalakan Lampu 4 melalui <i>button</i> pada halaman <i>web</i> .	Lampu 4 menyala dengan indikator <i>button</i> sebelumnya berwarna merah berubah menjadi warna hijau.	Ya	
8	Mematikan Lampu 4 melalui <i>button</i> pada halaman <i>web</i> .	Lampu 4 mati dengan indikator <i>button</i> sebelumnya berwarna hijau berubah menjadi warna merah.	Ya	
9	Menyalakan Lampu 5 melalui <i>button</i> pada halaman <i>web</i> .	Lampu 5 menyala dengan indikator <i>button</i> sebelumnya berwarna merah berubah menjadi warna hijau.	Ya	
10	Mematikan Lampu 5 melalui <i>button</i> pada halaman <i>web</i> .	Lampu 5 mati dengan indikator <i>button</i> sebelumnya berwarna hijau berubah menjadi warna merah.	Ya	

No	Skenario	Kondisi perangkat dan aplikasi	Hasil	
			Sesuai	Tidak
11	Menyalakan Lampu 6 melalui <i>button</i> pada halaman <i>web</i> .	Lampu 6 menyala dengan indikator <i>button</i> sebelumnya berwarna merah berubah menjadi warna hijau.	Ya	
12	Mematikan Lampu 6 melalui <i>button</i> pada halaman <i>web</i> .	Lampu 6 mati dengan indikator <i>button</i> sebelumnya berwarna hijau berubah menjadi warna merah.	Ya	

Dari Tabel 5 memperlihatkan bahwa pengujian *Controlling* Lampu Menggunakan *Web* berjalan dengan baik dan tombol *button* dapat berfungsi dan juga dapat melakukan *controlling* lampu dengan indikator ON dan OFF untuk melakukan perintah nyala dan matikan lampu yang dikirimkan melalui protokol MQTT.

### 5. KESIMPULAN DAN SARAN

Dari beberapa pengujian yang telah dilakukan, dapat ditarik beberapa kesimpulan yaitu, pengujian yang dilakukan menggunakan protokol MQTT pada *client* dapat saling berinteraksi satu sama lain antara *publisher* dan *subscriber* melalui *broker* dengan *topic* melalui jaringan *internet* dengan proses pengiriman pesan atau perintah menggunakan protokol MQTT rata-rata waktu yang dibutuhkan untuk menghidupkan (ON) atau mematikan lampu (OFF) yaitu 0.36167 dan 0.42167 detik dan saran untuk pengembangan selanjutnya bisa menambahkan komponen alat lebih banyak lagi serta membuat aplikasi *controlling* berbasis *mobile* atau *android*.

### DAFTAR PUSTAKA

[1] J. Budiarto dan S. Hadi, "Sistem Kendali Peralatan Elektronik Rumah Tangga Berbasis Internet Of Things Menggunakan Protokol MQTT Jurnal BITE: Jurnal Bumigora Information Technology Jurnal BITE: Jurnal Bumigora Information Technology," *J. BITE Vol.2 No.1 2020, Hal 1-11 Sist.*, vol. 2, no. 1, hal. 1–11, 2020, doi: 10.30812/bite.v2i1.799.

[2] R. P. Pratama, "Pengendali Lampu Rumah Berbasis Esp8266 Dengan Protokol Mqtt,"

- TESLA *J. Tek. Elektro*, vol. 22, no. 1, hal. 56, 2020, doi: 10.24912/tesla.v22i1.7862.
- [3] C. Hasiholan, R. Primananda, dan K. Amron, "Implementasi Konsep Internet of Things pada Sistem Monitoring Banjir menggunakan Protokol MQTT," *Pengemb. Teknol. Inf. dan Ilmu Komput.*, vol. 2, no. 12, hal. 6128–6135, 2018, [Daring]. Tersedia pada: <https://j-ptiik.ub.ac.id/index.php/j-ptiik/article/download/3529/1389/>.
- [4] S. Rosadi, P. Studi, dan S. Komputer, "PROTOTYPE SISTEM PENDETEKSI BANJIR MENGGUNAKAN," vol. 5, no. 1, hal. 38–48, 2022.
- [5] A. Farid, "SISTEM MONITORING DAN KONTROLING PADA SMART BUILDING DENGAN PENERAPAN IOT (Internet of Things)," *JATI (Jurnal Mhs. Tek. Inform.)*, vol. 1, no. 1, hal. 89–95, 2017.
- [6] Z. Mindriawan, I. W. A. Arimbawa, dan I. G. P. S. Wijaya, "Implementasi Internet of Things Pada Sistem Monitoring Suhu dan Kontrol Air Pada Kandang Burung Puyuh Petelur dengan Menggunakan Protokol MQTT," *Univ. Mataram Repos.*, hal. 1–8, 2018, [Daring]. Tersedia pada: <http://eprints.unram.ac.id/11498/>.
- [7] Y.; Herdiana dan E. Awaludin, "APLIKASI RADIO FREQUENCY IDENTIFICATION MENGGUNAKAN NODEMCU V3 ESP8266 UNTUK ABSENSI PEGAWAI DI SMK NEGERI 7 BALEENDAH (Studi Kasus SMK Negeri 7 Baleendah) Yudi," *Angew. Chemie Int. Ed.* 6(11), 951–952., hal. 2013–2015, 2021.
- [8] Z. B. Abilovani, W. Yahya, dan F. A. Bakhtiar, "Implementasi Protokol MQTT Untuk Sistem Monitoring Perangkat IoT," *J. Pengemb. Teknol. Inf. dan Ilmu Komput.*, vol. 2, no. 12, hal. 7521–7527, 2018, [Daring]. Tersedia pada: <http://j-ptiik.ub.ac.id>.
- [9] G. Y. Saputra, A. D. Afrizal, F. K. R. Mahfud, F. A. Pribadi, dan F. J. Pamungkas, "Penerapan Protokol MQTT Pada Teknologi Wan (Studi Kasus Sistem Parkir Univeristas Brawijaya)," *Inform. Mulawarman J. Ilm. Ilmu Komput.*, vol. 12, no. 2, hal. 69, 2017, doi: 10.30872/jim.v12i2.653.
- [10] D. I. Saputra, G. M. Karmel, dan Y. B. Zainal, "Perancangan Dan Implementasi Rapid Temperature Screening Contactless Dan Jumlah Orang Berbasis Iot Dengan Protokol Mqtt," *J. Energy Electr. Eng.*, vol. 2, no. 1, hal. 20–30, 2020, doi: 10.37058/jeee.v2i1.2147.
- [11] B. Sutomo dan T. Aristi Saputri, "Remote Home Monitoring Menggunakan Protokol MQTT," *Semin. Nas. Teknol. dan Bisnis 2018*, hal. 146–153, 2018.
- [12] B. Artono dan R. G. Putra, "Penerapan Internet Of Things (IoT) Untuk Kontrol Lampu Menggunakan Arduino Berbasis Web," *J. Teknol. Inf. dan Terap.*, vol. 5, no. 1, hal. 9–16, 2019, doi: 10.25047/jtit.v5i1.73.
- [13] Hasri Awal, "Perancangan Prototype Smart Home Dengan Konsep Internet of Thing ( IoT) Berbasis Web Server," *Maj. Ilm. UPI YPTK*, no. 26, hal. 65–79, 2019, doi: 10.35134/jmi.v26i2.53.
- [14] A. Shafitri dan A. Mashuri, "PERANCANGAN PENGENDALI LAMPU KANTOR," vol. 9, no. 1, 2022
- [15] F. Sintia dan P. Triadyaksa, "Sistem Otomatisasi Dan Kendali Jarak Jauh Lampu Smart House Berbasis Nodemcu Esp8266," vol. 23, no. 4, 2020.
- [16] R. Muzawi, Y. Efendi, dan W. Agustin, "Sistem Pengendalian Lampu Berbasis Web dan Mobile," *SATIN - Sains dan Teknol. Inf.*, vol. 4, no. 1, hal. 29, 2018, doi: 10.33372/stn.v4i1.292.
- [17] A. H. Hendri dan Mochammad Arief Sutisna, "Article Desktop Based National Police Commission Activities Information System," *J. CoSciTech (Computer Sci. Inf. Technol.)*, vol. 2, no. 1, hal. 14–23, 2021, doi: 10.37859/coscitech.v2i1.2393.
- [18] A. Azura dan W. Wildian, "Rancang Bangun Sistem Absensi Mahasiswa Menggunakan Sensor RFID dengan Database MySQL XAMPP dan Interface Visual Basic," *J. Fis. Unand*, vol. 7, no. 2, hal. 186–193, 2018, doi: 10.25077/jfu.7.2.186-193.2018.