

**SISTEM KONTROL LAMPU PARKIR POLMAN BABEL  
BERBASIS *INTERNET OF THINGS* (IoT)**

**PROYEK AKHIR**

Laporan akhir ini dibuat dan diajukan untuk memenuhi salah satu syarat kelulusan  
Diploma III Politeknik Manufaktur Negeri Bangka Belitung



Disusun oleh:

Muhammad Asqal      NIM:    0032020

Eby Jhoanca          NIM:    0032012

**POLITEKNIK MANUFAKTUR NEGERI  
BANGKA BELITUNG  
TAHUN 2023**

**LEMBAR PENGESAHAN**

**JUDUL PROYEK AKHIR**  
**SISTEM KONTROL LAMPU PARKIR POLMAN BABEL BERBASIS**  
***INTERNET OF THINGS (IoT)***

Oleh:  
Muhammad Asqal/0032020  
Eby Jhoanca/0032012

Laporan akhir ini telah disetujui dan disahkan sebagai salah satu syarat kelulusan  
Program Diploma III Politeknik Manufaktur Negeri Bangka Belitung

Menyetujui,

**Pembimbing 1**



Ocsirendi, M.T.

**Pembimbing 2**



Eko Sulistyio, M.T.

**Penguji 1**



Zanu Saputra, M.Tr.T.

**Penguji 2**



Ahmad Josi, S. Kom., M. Kom.

## PERNYATAAN BUKAN PLAGIAT

Yang bertanda tangan di bawah ini:

Nama Mahasiswa 1: Muhammad Asqal                      NIM: 0032020

Nama Mahasiswa 2: Eby Jhoanca                         NIM: 0032012

Dengan Judul: SISTEM KONTROL LAMPU PARKIR POLMAN  
BABEL BERBASIS *INTERNET OF THINGS* (IoT)


Menyatakan bahwa laporan akhir ini adalah hasil kerja kami sendiri dan bukan merupakan plagiat. Pernyataan ini kami buat dengan sebenarnya dan bila ternyata di kemudian hari melanggar pernyataan ini, kami bersedia menerima sanksi yang berlaku.

Sungailiat, 28 Mei 2023


Nama Mahasiswa

Tanda Tangan

1. Muhammad Asqal

  
.....

2. Eby Jhoanca

  
.....

## ABSTRAK

*Lampu merupakan suatu perangkat yang dapat menghasilkan cahaya saat dialiri listrik. Namun, pengaktifan lampu yang berlebihan sering dijumpai bahkan lampu pada parkir Polman Babel yang dapat mengakibatkan pemborosan listrik. Oleh karena itu, diperlukan sebuah sistem pengontrolan lampu parkir Polman Babel secara otomatis agar dapat mempermudah pekerjaan. Tujuan pembuatan alat ini adalah merancang sistem kontrol lampu untuk menyalakan dan mematikan lampu dengan menggunakan aplikasi telegram berbasis Internet of Things (IoT). Alat ini dibuat untuk mengontrol lampu yang ada di lahan parkir Polman Babel. Pengontrolan lampu dilakukan dengan cara mengontrol 9 buah lampu yang dibagi menjadi 3 zona dimana setiap zona terdapat 3 buah lampu. Ketiga zona ini saling terhubung dalam sebuah aplikasi Telegram. Didalam aplikasi Telegram sendiri terdapat 9 buah tombol yang digunakan untuk menyalakan dan mematikan lampu. Metode penelitian yang digunakan yaitu perancangan dan pembuatan sistem kontrol lampu menggunakan ESP8266, dan perancangan dan pembuatan kontrol lampu melalui Telegram serta pengujian alat. Hasil dari pembuatan sistem kontrol lampu telah dilakukan dengan baik, dimana hasil yang didapat sistem kontrol bisa melakukan pengontrolan untuk menyalakan dan mematikan lampu sesuai dengan perintah yang di berikan.*

*Kata kunci: Internet of Things (IoT), Sistem Kontrol, Telegram.*

## **ABSTRACT**

A lamp is a device that can produce light when electrified. However, excessive activation of lights is often found even lights in the parking lot of Polman Babel which can result in waste of electricity. Therefore, an automatic Polman Babel parking light control system is needed in order to facilitate work. The purpose of making this tool is to design a lamp control system to turn lights on and off using the Internet of Things (IoT)-based telegram application. This tool is made to control the lights in the parking lot of Polman Babel. Controlling the lights is done by controlling 9 lights which are divided into 3 zones where each zone has 3 lights. These three zones are interconnected in a Telegram application. In the Telegram application itself there are 9 buttons used to turn on and off the lights. The research methods used are designing and manufacturing lamp control systems using ESP8266, and designing and manufacturing lamp controls via Telegram and tool testing. The results of making a lamp control system have been done well, where the results obtained by the control system can control to turn on and off the lights in accordance with the commands given.

**Keywords:** Internet of Things (IoT), Control System, Telegram.

## KATA PENGANTAR

Bismillahirrahmanirrahim

Puji dan syukur penulis panjatkan kehadirat Tuhan Yang Maha Esa berkat rahmat dan karunia-Nya sehingga penulis dapat menyelesaikan makalah proyek akhir ini dengan baik dan sesuai dengan waktu yang telah ditentukan. Makalah ini dibuat untuk memenuhi salah satu syarat wajib kelulusan Diploma III di Politeknik Manufaktur Negeri Bangka Belitung. Dengan adanya makalah ini diharapkan para pembaca dapat mengetahui gambaran proyek akhir yang dibuat oleh penulis. Makalah proyek akhir ini dibuat dengan melakukan pengembangan dari jurnal-jurnal penelitian yang sudah pernah dilakukan sebelumnya. Penulis mencoba menerapkan ilmu pengetahuan yang didapat selama 3 tahun menempuh pendidikan di Politeknik Manufaktur Negeri Bangka Belitung dalam pembuatan proyek akhir ini. Selain itu, penulis juga mendapatkan informasi berupa data-data pendukung dari makalah-makalah proyek akhir mahasiswa Politeknik Manufaktur Negeri Bangka Belitung tahun-tahun sebelumnya.

Selama menyusun makalah proyek akhir ini penulis mendapatkan banyak bantuan dan bimbingan dari berbagai pihak sehingga proses penulisan makalah ini dapat diselesaikan dengan baik. Oleh karena itu, penulis ingin mengucapkan terima kasih kepada:

1. Orangtua dan keluarga tercinta yang selalu memberikan dukungan baik secara moral maupun materil sehingga penulis dapat memberikan hasil yang terbaik.
2. Bapak I Made Andik Setiawan, M.Eng., Ph.D. selaku Direktur Politeknik Manufaktur Negeri Bangka Belitung.
3. Bapak Ocsirendi, M.T. selaku Ketua Program Studi DIII Teknik Elektronika Politeknik Manufaktur Negeri Bangka Belitung sekaligus dosen pembimbing 1 dalam proyek akhir ini.

4. Bapak Eko Sulisty, M.T. selaku Wakil Direktur III Politeknik Manufaktur Negeri Bangka Belitung sekaligus dosen pembimbing 2 dalam proyek akhir ini.
5. Bapak Zanu Saputra M.Tr.T selaku Ketua Jurusan Teknik Elektro dan Informatika Politeknik Manufaktur Negeri Bangka Belitung.
6. Seluruh tenaga pendidik dan kependidikan Politeknik Manufaktur Negeri Bangka Belitung.
7. Rekan-rekan mahasiswa Politeknik Manufaktur Negeri Bangka Belitung yang telah membantu dalam penyelesaian proyek akhir ini.
8. Seluruh pihak yang ikut terlibat baik secara langsung maupun tidak langsung dalam penyelesaian proyek akhir ini yang tidak bisa disebutkan satu persatu.

Penulis menyadari bahwa dalam penulisan laporan ini masih terdapat banyak kekurangan karena terbatasnya pengetahuan dan kemampuan penulis. Oleh karena itu, penulis mengharapkan kritik dan saran yang membangun dari semua pihak demi perbaikan dan pengembangan penulisan makalah ini di masa yang akan datang. Penulis berharap makalah ini dapat bermanfaat bagi pihak yang berkepentingan pada khususnya dan bagi perkembangan ilmu pengetahuan dan teknologi pada umumnya.

Sungailiat, 28 Mei 2022



Penulis

## DAFTAR ISI

LEMBAR PENGESAHAN .....	ii
PERNYATAAN BUKAN PLAGIAT .....	iii
ABSTRAK .....	iv
ABSTRACT .....	v
KATA PENGANTAR .....	vi
DAFTAR ISI .....	viii
DAFTAR TABEL .....	x
DAFTAR GAMBAR .....	xi
BAB I .....	1
PENDAHULUAN .....	1
1.1 Latar Belakang .....	1
1.2 Rumusan Masalah .....	2
1.3 Batasan Masalah .....	2
1.4 Tujuan .....	3
BAB II .....	4
DASAR TEORI .....	4
2.1 Sistem Kontrol Lampu .....	4
2.2 Pengontrolan Lampu dengan IoT .....	7
BAB III .....	11
METODE PELAKSANAAN .....	11
3.1 Perancangan Sistem yang Digunakan .....	11
3.2 Perancangan <i>Hardware</i> .....	13
3.3 Perancangan <i>Software</i> .....	14
3.4 Pengujian Keseluruhan Sistem .....	15
3.5 Pengumpulan Data Uji .....	15
3.6 Analisis Data .....	15
3.7 Pembuatan Makalah Proyek Akhir .....	16
BAB IV .....	17
PEMBAHASAN .....	17
4.1 Pembuatan <i>Software</i> .....	17
4.2 Pembuatan <i>Hardware</i> .....	19



4.3 Pengujian Sistem.....	20
4.4 Pengujian dengan Aplikasi Telegram .....	23
4.5 Pengujian Kontrol Lampu.....	27
4.6 Pengujian Menggunakan Provider .....	29
4.7 Analisa Seluruh Pengujian .....	35
BAB V.....	37
PENUTUP.....	37
5.1 Kesimpulan .....	37
5.2 Saran.....	37
DAFTAR PUSTAKA .....	39



## DAFTAR TABEL

Tabel 4.1 Pengujian Fungsi Tombol Lampu 1 Dan 2 Untuk Menyalakan dan Mematikan Lampu Dalam Aplikasi Telegram .....	20
Tabel 4.2 Pengujian Fungsi Tombol Lampu 1 dan 2 Menyalakan dan Mematikan Lampu Secara Langsung dengan Telegram.....	22
Tabel 4.3 Hasil Pengujian dengan Aplikasi Telegram .....	24
Tabel 4.4 Pengujian Delay Lampu 1 dan 2 didalam Aplikasi Telegram .....	25
Tabel 4.5 Hasil Pengujian Kontrol Lampu dengan Aplikasi Telegram.....	28
Tabel 4.6 Hasil pengujian dengan beberapa Provider .....	29
Tabel 4.7 Pengujian Kecepatan <i>delay</i> Menyalakan dan Mematikan Lampu 1 dan Lampu 2 dengan Menggunakan Wi-Fi Indihome .....	30
Tabel 4.8 Pengujian Kecepatan <i>delay</i> Menyalakan dan Mematikan Lampu 1 dan Lampu 2 dengan Menggunakan Ichonnect .....	31
Tabel 4.9 Pengujian Kecepatan <i>delay</i> Menyalakan dan Mematikan Lampu 1 dan Lampu 2 dengan Menggunakan Smartfren .....	32
Tabel 4.10 Pengujian Kecepatan <i>delay</i> Menyalakan dan Mematikan Lampu 1 dan Lampu 2 dengan Menggunakan Telkomsel .....	33
Tabel 4.11 Pengujian Kecepatan <i>delay</i> Menyalakan dan Mematikan Lampu 1 dan Lampu 2 dengan Menggunakan Indosat .....	34
Tabel 4.12 Pengujian Kecepatan <i>delay</i> Menyalakan dan Mematikan Lampu 1 dan Lampu 2 dengan Menggunakan XL-Axiata.....	35

## DAFTAR GAMBAR

Gambar 2.1 <i>Flowchart</i> Kerja Sistem .....	4
Gambar 2.2 <i>Flowchart</i> Kerja Sistem .....	5
Gambar 2.3 Diagram Blok Alat.....	6
Gambar 2.4 ESP8266.....	7
Gambar 2.5 Prinsip Kerja Aplikasi <i>Blynk</i> .....	8
Gambar 2.6 Cara Kerja Pengontrolan Lampu Dengan Menggunakan <i>Web Browser</i> .....	9
Gambar 2.7 Pengontrolan Sistem Menggunakan Telegram .....	10
Gambar 3.1 <i>Flowchart</i> Sistem .....	11
Gambar 3.2 Diagram Blok.....	12
Gambar 3.3 Rancangan <i>Hardware</i> Elektrik .....	13
Gambar 3.4 Skema Rangkaian.....	14
Gambar 4.1 Tampilan <i>start</i> pada <i>BotFather</i> .....	17
Gambar 4.2 Tampilan Kontrol Lampu dengan Aplikasi Telegram bot.....	18
Gambar 4.3 Pembuatan <i>Hardware</i> .....	20

## DAFTAR LAMPIRAN

- Lampiran 1 Daftar Riwayat Hidup  
Lampiran 2 Progam ESP8266 ESP8266



# BAB I

## PENDAHULUAN

### 1.1 Latar Belakang

Lampu merupakan suatu perangkat yang dapat menghasilkan cahaya saat dialiri listrik dan menjadi salah satu alat listrik yang sangat dibutuhkan dalam kehidupan manusia terutama untuk beraktivitas di malam hari [1]. Namun, penggunaan lampu yang berlebihan sering dijumpai di kalangan masyarakat seperti lampu yang tidak dimatikan meskipun sudah pagi akan mengakibatkan pemborosan listrik [2]. Oleh karena itu, penggunaan lampu yang lebih efisien dan perlu dipertimbangkan. Seiring berkembangnya ilmu pengetahuan dan teknologi saat ini maka akan terus muncul pemikiran-pemikiran baru yang berguna di kehidupan sehari-hari untuk merubah sistem manual menjadi sistem digital serta otomatis [3]. Kemajuan teknologi kini memicu pola pikir manusia untuk dapat menciptakan inovasi-inovasi untuk memudahkan pekerjaan demi kinerja yang lebih baik, salah satunya modernisasi menyalakan dan mematikan lampu dari jarak jauh [4].

Contoh permasalahan yang ada terletak pada lampu parkir dosen Polman Babel. Diketahui bahwa parkir tersebut belum mempunyai penerangan. Padahal keberadaan lampu penerangan di area parkir itu sangat dibutuhkan karena pada malam hari area tersebut sering digunakan untuk memarkirkan kendaraan oleh mahasiswa, dosen, maupun staff yang berkegiatan di area kampus. Oleh karena itu, selain mengatasi masalah penerangan, juga diperlukan sebuah sistem pengontrolan lampu parkir Polman Babel dari jarak jauh agar dapat mempermudah pekerjaan.

Pengontrolan pada lampu sudah dilakukan dalam beberapa penelitian sebelumnya yaitu pada penelitian Abdaoe dkk. yang membuat sistem kendali lampu otomatis berbasis *Internet of Things* (IoT) menggunakan ESP8266 pada lampu pijar sebanyak dua dan membuat aplikasi di android menggunakan *software tool Android Studio* [5]. Kemudian pada penelitian David Setiawan membuat sistem kontrol empat buah lampu menggunakan metode manual dan otomatis berbasis *handphone* yang dihubungkan melalui *bluetooth* dengan membuat aplikasi di *handphone* menggunakan *MIT App Inventor* [6]. Adapun selanjutnya pada penelitian Sidauruk

dkk. membuat rancang bangun sistem dua buah lampu otomatis berbasis *Internet of Things* (IoT) dengan aplikasi *Raspberry Pi* [7]. Namun, penelitian di atas hanya mengontrol paling banyak 3 buah lampu. Belum ada penelitian yang mengontrol lampu dalam jumlah banyak.

Berlandaskan dari beberapa penelitian yang telah dijalankan di atas maka dalam proyek akhir ini akan dibuat sebuah alat untuk mengontrol 9 buah lampu parkir Polman Babel secara otomatis. Pada Saat ini kemajuan dalam teknologi komunikasi sudah membuat sebuah konsep *Internet of Things* (IoT) yang memberikan keringanan terhadap manusia dalam membagikan data dengan penggunaan internet sebagai medianya. Dari konsep ini, proses yang berlangsung dalam sebuah sistem diperoleh menjadi lebih efektif serta efisien. Berhubungan dengan implementasi dari teknologi itu, maka dalam proyek akhir ini akan dibuat “Sistem Kontrol Lampu Parkir Polman Babel Berbasis *Internet of Things* (IoT). Dengan adanya alat ini diharapkan dapat memberi kenyamanan dan kemudahan kepada pengguna untuk dapat mengontrol lampu.

## **1.2 Rumusan Masalah**

Rumusan masalah yang dibuat dalam proyek akhir ini adalah sebagai berikut.

1. Bagaimana cara membuat sistem kontrol menggunakan ESP8266 untuk mengontrol lampu parkir?
2. Bagaimana cara mengontrol lampu parkir dengan menggunakan *smartphone*?

## **1.3 Batasan Masalah**

Untuk mempertajam arah pembahasan dari proyek akhir ini, maka akan diberikan batasan masalah sebagai berikut.

1. Penggunaan sistem kontrol lampu parkir berbasis *Internet of Things* (IoT) yang menghubungkan perangkat dan sistem secara *online* dan sangat bergantung pada koneksi internet.
2. Keterbatasan status lampu nyala atau mati setelah diberi perintah. Pada aplikasi hanya ada status *relay* yang aktif atau tidak. Jika menyalakan lampu

walaupun lampu dalam kondisi rusak, status pada aplikasi tetap lampu sudah menyala.

#### **1.4 Tujuan**

Adapun tujuan yang akan ditargetkan dalam pembuatan proyek akhir ini adalah sebagai berikut.

1. Membangun sistem kontrol menggunakan ESP8266 untuk mengontrol lampu parkir.
2. Mengontrol lampu parkir dengan menggunakan *smartphone*.

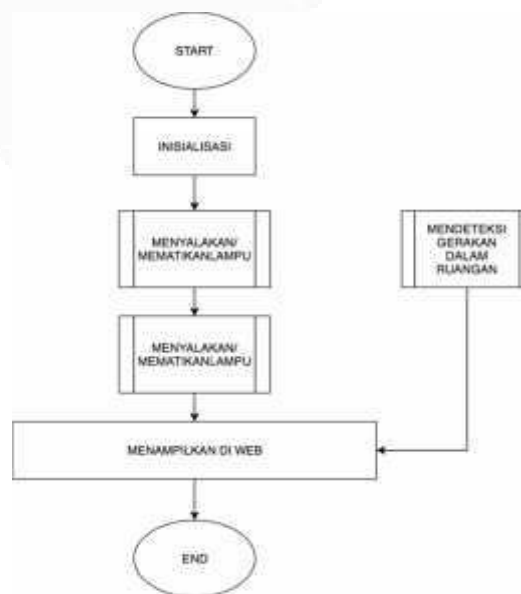


## BAB II DASAR TEORI

### 2.1 Sistem Kontrol Lampu

Sistem kontrol disebut juga sistem pengendali dikatakan sebagai sebuah sistem yang mengatur suatu input atau masukan untuk digunakan sebagai pengontrol output atau luaran. Sistem kontrol dapat mengurutkan proses ataupun membuat suatu output jika berbagai kondisi terpenuhi [8].

Sistem kontrol lampu merupakan cara untuk mengendalikan atau mengontrol lampu jika ingin dalam keadaan nyala atau mati. Sistem kontrol lampu ini sebelumnya sudah diimplementasikan dalam beberapa penelitian, seperti pada penelitian yang telah dilakukan oleh Virgina Anathasya, dkk. yang berjudul *Rancang Bangun Sistem Kontrol Lampu Berbasis Arduino Mega 2560*. Adapun sistem kerja dari penelitian ini dapat dilihat pada Gambar 2.1 berikut ini.



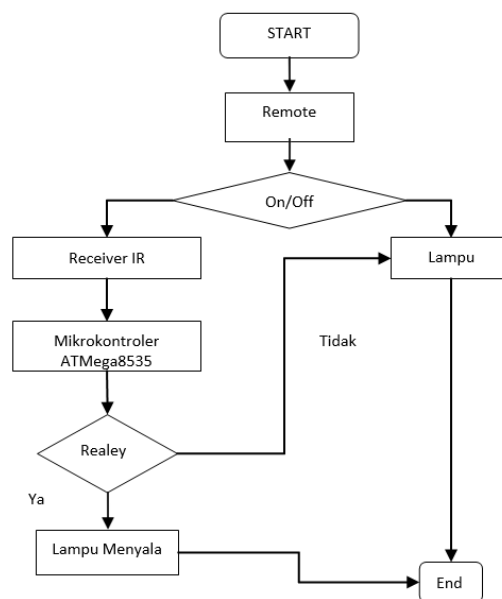
Gambar 2.1 *Flowchart* Kerja Sistem [8]

Penelitian ini memiliki tujuan untuk mempermudah pekerjaan satpam dan bagian umum di Universitas Pembangunan Jaya dalam pemantauan penggunaan dan kondisi ruangan di Universitas Pembangunan Jaya, serta memberikan



kenyamanan kepada pengguna lampu di Universitas Pembangunan Jaya dalam menggunakan ruangan. Penelitian [8] ini dengan meyalakan atau mematikan lampu diberbagai ruangan dalam suatu gedung. Dalam penelitian ini digunakan dua tombol *on* dan *off* untuk menyala dan mematikan lampu pada ruangan. Terdapat delapan lampu yang akan dikontrol dengan membagi menjadi empat grup lampu dimana masing-masing grup lampu terdapat dua bohlam lampu yang akan disimulasikan. Selain dengan menggunakan saklar, lampu dapat dinyalakan atau dimatikan dengan menggunakan Web. Namun, pada penelitian ini sistem kontrol lampu masih dalam bentuk *prototype* dan belum ada penerapan langsung dalam dunia nyata.

Kemudian, pada penelitian yang dilakukan oleh Muksin Hi. Abdullah yang berjudul *Rancang Bangun Sistem Kontrol Lampu Listrik Menggunakan Remote Berbasis Mikrokontroler AtMega 8535*. Adapun cara kerja sistem dari penelitian ini dapat dilihat pada Gambar 2.2 berikut ini.

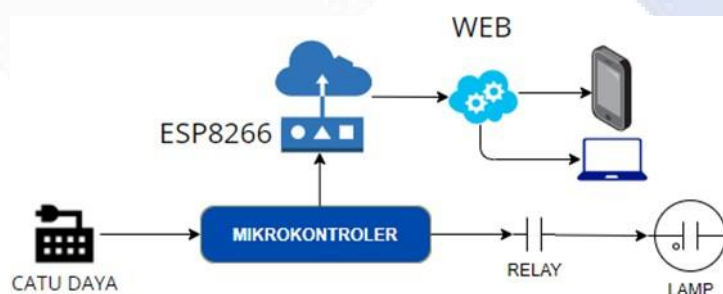


Gambar 2.2 *Flowchart* Kerja Sistem [9]

Penelitian [9] ini juga dilakukan atau dibuat sistem kontrol lampu dengan menggunakan *remote*. Pada penelitian ini merancang sistem kontrol lampu listrik menggunakan *remote* bertujuan agar proses menghidupkan lampu lebih mudah dengan hanya memberikan pilihan tombol tidak. Jika menekan *on* maka lampu

dapat dinyalakan dan menekan *off* maka lampu dapat dipadamkan. Sistem kontrol lampu yang dirancang oleh peneliti dioperasikan dengan menggunakan *remote control* untuk mengaktifkan *relay* kemudian lampu yang terhubung akan menyala dan padam. *Remote control* tersebut terhubung dengan *receiver* yang ada pada mikrokontroler dengan menekan tombol B, kemudian lampu menyala/hidup ketika menekan tombol A, dan lampu padam/mati melalui sistem kontrol. Namun, pada penelitian ini kontrol lampu masih menggunakan *remote* dan belum adanya pemanfaatan teknologi *Internet of Things* yang sebenarnya dapat lebih mempermudah pekerjaan, sedangkan pada sistem ini jarak inframerah yang dipancarkan oleh *remote* hanya terdeteksi pada batas maksimal 15 meter.

Selanjutnya, pada penelitian Laban Yoel Ishak Frare, dkk. yang berjudul *Rancang Bangun Alat Kontrol Lampu Jarak Jauh Menggunakan Mikrokontroler Berbasis Website*. Adapun diagram kontrol pada penelitian ini dapat dilihat pada Gambar 2.3 berikut ini.



Gambar 2.3 Diagram Blok Alat [10]

Penelitian ini mengontrol lampu secara otomatis dengan cara mengaktifkan *relay* melalui arus listrik. Pengujian pada penelitian ini memanfaatkan ESP8266 sebagai penghubung dan dikontrol melalui *smartphone* dengan jarak pengontrolan  $\pm 10$  meter yang otomatis akan mengaktifkan setiap lampu. Pengontrolan lampu jarak jauh melalui Web memanfaatkan ESP8266 dan menggunakan akses *point*. Cara kerja pengontrolan ini dengan menggunakan program khusus yang dibuat untuk rangkaian ESP8266. Modul Wi-Fi akan mengirim dan menerima data melalui Web dengan menggunakan 2 akses *point* berupa *smartphone* kemudian dimasukan kode dari IP Address yang terhubung pada ESP8266 [10].

Dari beberapa penelitian yang dilakukan untuk mengontrol lampu dengan berbagai cara, pada proyek akhir sebagai pengembangan akan dibuat sistem kontrol lampu yang digunakan untuk menyalakan atau mematikan lampu dari jarak jauh dengan memanfaatkan teknologi *Internet of Things* (IoT) dengan pengontrolan melalui aplikasi karena dapat mempermudah pengguna dalam mengontrol lampu.

## 2.2 Pengontrolan Lampu dengan IoT

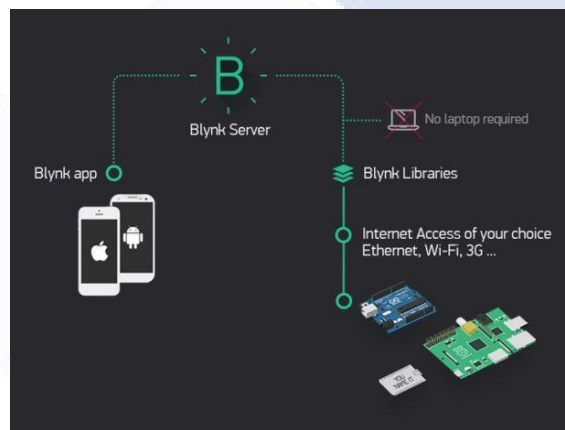
Pengontrolan sistem lampu ini menggunakan mikrokontroler dalam penggunaannya. Mikrokontroler yang dipakai berupa ESP8266 karena dapat langsung digunakan untuk menghubungkan ke perangkat di *smartphone* melalui Wi-Fi. ESP8266 merupakan pengembangan dari modul *Internet of Things* (IoT) yang fungsinya hampir sama dengan modul Arduino. Hanya saja yang membedakan ESP8266 dengan Arduino secara fungsi adalah ESP8266 ini dikhususkan untuk terhubung di internet. Walaupun ukuran dari modul ini terbilang kecil, namun ESP8266 sudah dilengkapi oleh *fitur wifi* dan *firmware* yang kode sumber atau kode dasarnya dapat dimodifikasi oleh pengguna.

Pengontrolan menggunakan ESP8266 dilakukan dengan menghubungkan pin *relay* ke pin yang ada di ESP8266. Pin yang digunakan pada ESP8266 adalah pin digital untuk mengontrol sistem. Untuk mengontrol lampu dihidupkan atau dimatikan, maka dikontrol melalui pin yang ada di ESP8266 yang terhubung ke *relay*. *Relay* yang digunakan yaitu *relay normally open* yang akan aktif ketika dialiri arus. Pengontrolan sistem dengan menggunakan ESP8266 dilakukan dengan melakukan pemrograman pada ESP8266 sehingga dapat menentukan *output* mana saja yang akan aktif jika diberi perintah.



Gambar 2.4 ESP8266 [11]

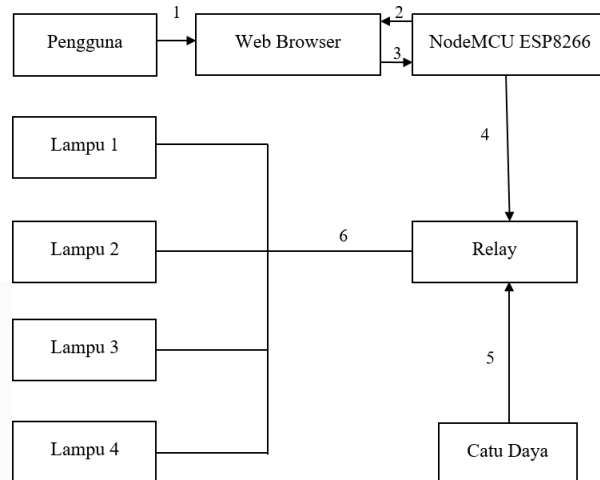
Pengontrolan lampu dengan memanfaatkan teknologi IoT ini dapat dilakukan dengan berbagai cara. Contoh dengan menggunakan aplikasi *Blynk*. Untuk mengontrol lampu menggunakan Aplikasi *Blynk* dilakukan dengan cara awal memastikan perangkat keras terhubung ke jaringan WiFi yang sama dengan ponsel, kemudian memulai perangkat keras, lalu membuka aplikasi *Blynk* di ponsel dan masuk ke proyek yang telah di buat sebelumnya. Di dalam proyek tersebut, akan terdapat tombol yang mewakili *widget* tombol yang telah di tambahkan untuk mengontrol lampu. Selanjutnya, dengan menekan tombol tersebut di aplikasi *Blynk*, sinyal akan dikirim ke perangkat keras, yang pada gilirannya akan mengontrol lampu sesuai dengan kode yang telah di atur sebelumnya [12]. Sistem kerja aplikasi *Blynk* yang bisa mengontrol sistem bisa dilihat dalam Gambar 2.5 berikut ini.



Gambar 2.5 Prinsip Kerja Aplikasi *Blynk* [13]

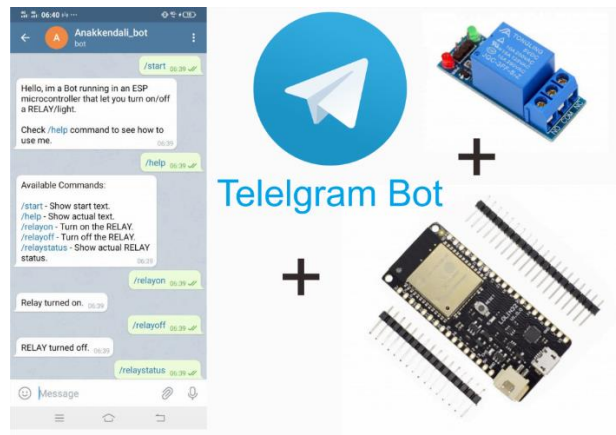
Selain, menggunakan aplikasi *Blynk*, pengontrolan sistem lampu ini juga bisa memanfaatkan *Web Browser*. Untuk mengontrol lampu menggunakan sistem kendali *Web Browser* dilakukan dengan cara awal memprogram Arduino IDE yang terhubung dengan ESP8266 agar dapat dikendalikan dengan *Web Browser*. Kemudian, mempersiapkan program yang akan di-*upload* melalui Arduino IDE yang harus terkoneksi dengan jaringan Wi-Fi. Selanjutnya, menghubungkan ESP8266 ke komputer menggunakan kabel USB dan menyambungkannya ke *port* yang sudah di tentukan. Setelah itu, dengan cara mengklik menu *upload* pada Arduino IDE untuk mengunggah program tersebut. Jika program sudah berhasil diunggah, alamat IP dapat dilihat pada serial monitor untuk mengetahui *IP Address*

yang terhubung. Setelah berhasil, laptop atau ponsel dapat dihubungkan ke internet yang telah dibuat oleh program. Kemudian, memasukan alamat IP yang ada di serial monitor ke *browser*. Pada pengujian sistem kendali lampu, untuk memastikan keberhasilan pemrograman yang telah dilakukan dengan menekan tombol On/Off pada *Web Browser* yang telah terhubung ke alamat *IP Address Android Smartphone* pemilik [14]. Adapun cara kerja pengontrolan lampu dengan menggunakan *Web Browser* dapat dilihat pada Gambar 2.6 berikut ini.



Gambar 2.6 Cara Kerja Pengontrolan Lampu Dengan Menggunakan *Web Browser*[14]

Kemudian, pengontrolan lampu dapat juga menggunakan aplikasi Telegram. Penggunaan dengan Telegram ini dilakukan dengan Bot Telegram. Bot Telegram ini akan mengirim kode agar kita dapat membuat Bot untuk mengontrol lampu. Barulah ESP8266 diprogram agar bisa terhubung ke telegram menggunakan Wi-Fi. Setelah selesai membuat perintah yang dibutuhkan, Bot Telegram untuk mengontrol lampu bisa digunakan [15]. Adapun contoh pengontrolan sistem menggunakan bot telegram dengan menggunakan ESP8266 berikut ini.



Gambar 2.7 Pengontrolan Sistem Menggunakan Telegram [15]

Dari banyaknya cara yang digunakan untuk memanfaatkan IoT, pada proyek akhir ini digunakan cara memanfaatkan aplikasi Telegram untuk menghidupkan atau mematikan lampu. Aplikasi Telegram ini dipilih sebagai media pengontrol dikarenakan pembuatan dan penggunaan yang lebih mudah dan *simple*, serta dapat berfungsi dengan baik.

Pengontrolan lampu menggunakan Telegram ini sudah pernah dilakukan oleh Muntaha Nega dkk yang berjudul *Internet Of Things (IoT) Kontrol Lampu Rumah Menggunakan NodeMcu dan ESP-12E Berbasis Telegram Chatbot*. Pada penelitian ini dilakukan pengontrolan lampu sebanyak 4 buah lampu yang dikontrol dengan 1 sistem [16]. Pada penelitian yang dilakukan pada proyek akhir ini menggunakan 9 buah lampu dan dikontrol menggunakan 3 buah sistem yang saling terkoneksi dalam aplikasi Telegram. Tentu sebagai alternatif lain digunakanlah saklar yang berguna sebagai pengontrolan lampu apabila sistem IoT tidak dapat bekerja sebagaimana mestinya.

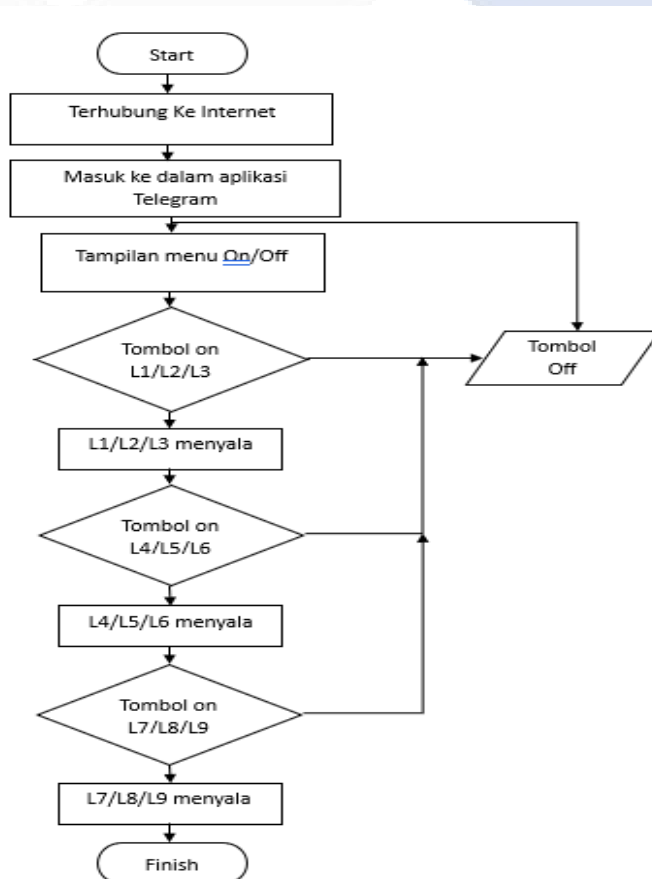
### BAB III

#### METODE PELAKSANAAN

Dari pelaksanaan tugas akhir ini, dijalankan beberapa langkah yang memudahkan penulis untuk penyelesaian tugas akhir. Langkah-langkah pengerjaan tugas akhir ini dapat digambarkan dengan menggunakan *flowchart* berikut.

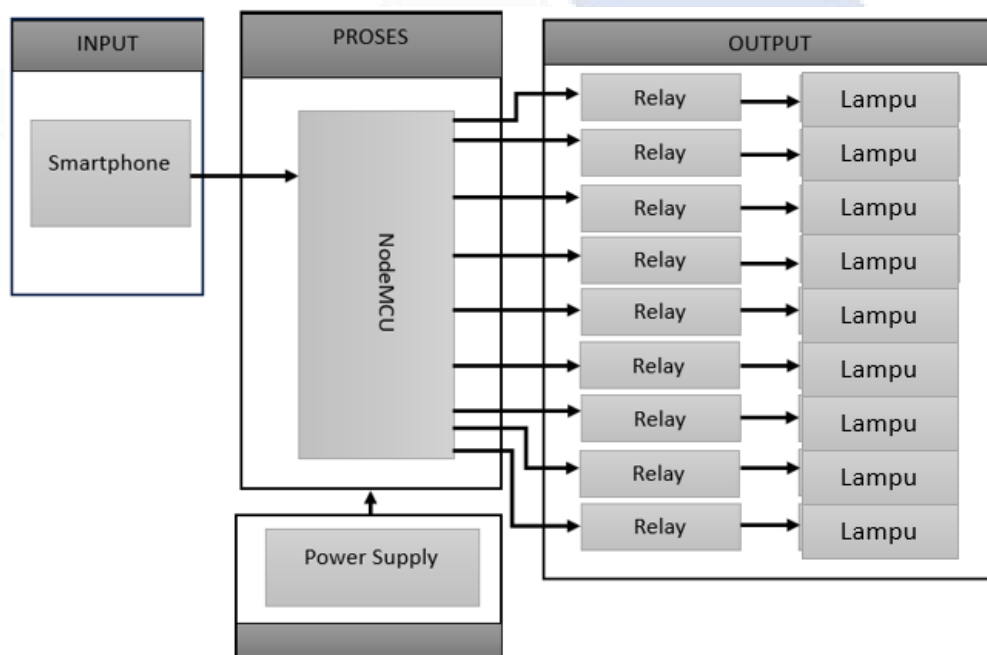
#### 3.1 Perancangan Sistem yang Digunakan

Sistem kontrol lampu dilakukan dengan memasukan perintah dari aplikasi Telegram dengan menggunakan ESP8266 sebagai penghubungnya. Adapun *flowchart* sistem yang digunakan dalam kontrol lampu dapat dilihat pada gambar 3.1 dibawah ini.



Berikut ini adalah penjelasan *flowchart* sistem kontrol lampu berbasis *Internet Of Things* (IoT) menggunakan ESP8266:

*Start* adalah tahap awal sistem kerja, kemudian buka aplikasi telegram yang sudah terlebih dahulu terkoneksi ke internet dengan mikrokontroler yang sudah terhubung yaitu ESP8266. Selanjutnya pada aplikasi android yaitu telegram akan menampilkan menu tampilan L1/L2/L3, L4/L5/L5 dan L7/L8/L9 tombol tersebut adalah perintah untuk menyalakan lampu. Jika menekan L1/L2/L3 maka kita akan memilih lampu mana yang akan kita nyalakan. Lalu jika menekan tombol tersebut untuk kedua kali maka akan akan mematikan lampu secara otomatis sesuai lampu mana yang akan *user* matikan. Adapun Diagram Blok perancangan sistem sebagai berikut:



Gambar 3.2 Diagram Blok

Penjelasan lebih lanjut dari tahapan pelaksanaan Diagram sebagai berikut:

*User* dapat memasukan perintah melalui aplikasi telegram seperti menghidupkan dan mematikan lampu. Ketika mikrokontroler menerima masukan dari aplikasi telegram, maka masukan tersebut akan menjadi dasar ESP8266 untuk menjalankan perintah yang sudah dibuat. Data yang sudah dikirimkan akan masuk ke sistem mikrokontroler melalui *port* serial yang ada pada ESP8266. Selanjutnya

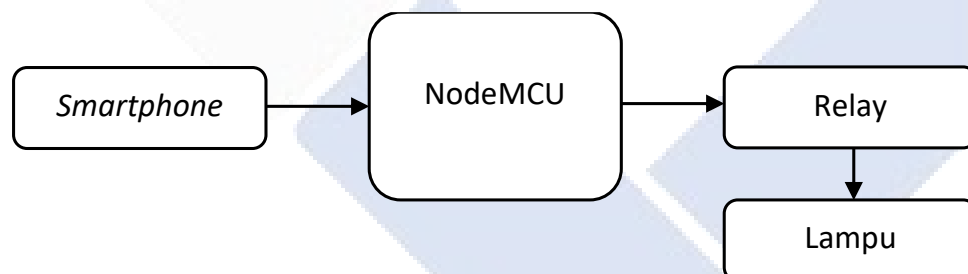


untuk mengaktifkan *relay*, data tersebut diolah pada ESP8266 kemudian data tersebut diteruskan ke pin GPIO untuk kendali pada *relay*.

### 3.2 Perancangan *Hardware*

Tahap merancang *hardware* dilakukan untuk merancang sistem kontrol otomatis pada lampu. Perancangan *hardware* bertujuan untuk menentukan komponen yang akan digunakan seperti, *relay*, ESP8266 dan lampu. Rancangan *hardware* pada sistem kontrol otomatis pada lampu ini ada 2 jenis, yaitu rancangan *hardware* mekanik dan rancangan *hardware* elektrik. Untuk perancangan *hardware* mekanik yang akan dibuat, yaitu dengan merancang pemasangan ESP8266 dan *relay* pada *box*.

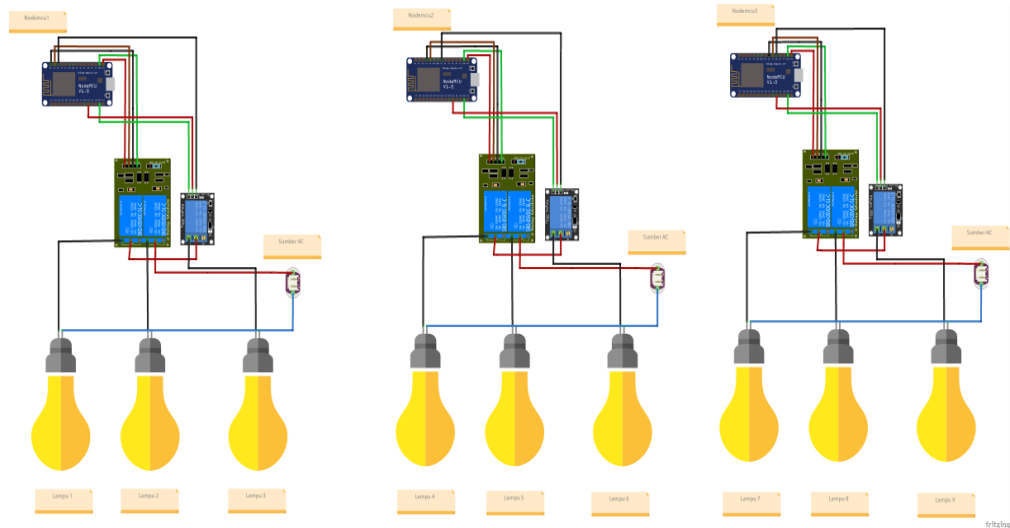
Sementara pada tahap perancangan *hardware* elektrik membuat rancangan skema rangkaian menggunakan ESP8266 yang berfungsi untuk mengontrol sistem pada lampu secara otomatis. Rancangan *hardware* elektrik dapat dilihat pada Gambar 3.3 sebagai berikut.



Gambar 3.3 Rancangan *Hardware* Elektrik

Komponen perangkat keras merupakan bagian utama dalam pengendalian lampu otomatis berbasis mikrokontroler menggunakan *smartphone* android, karena sistem ini sepenuhnya menggunakan *input*, proses dan *output*.

Terdapat skema rangkaian atau wiring bisa dilihat pada Gambar 3.4 dibawah ini.



Gambar 3.4 Skema Rangkaian

### 3.3 Perancangan *Software*

Pada tahap perancangan *software* merupakan tahapan yang dilakukan untuk menentukan dan merancang *software* yang akan digunakan dalam pembuatan proyek akhir ini serta bagaimana cara kerja sistem yang digunakan pada aplikasi. Adapun tahap perancangan *software* berupa pemrograman ESP8266 untuk mengontrol lampu yang ingin dihidupkan atau dimatikan yang dapat dikontrol menggunakan aplikasi Telegram dari *smartphone*. Perancangan *software* ini juga merupakan proses pembuatan rancangan tampilan pada Bot Telegram. Adapun *software* kontrol *monitoring* lampu parkir ini meliputi:

- Pencarian Bot Telegram untuk membuat bot baru agar dapat dikelola serta diberikan daftar perintah untuk mengontrol lampu.
- Pembuatan program keseluruhan pada ESP8266.

Pada tahap perancangan dan pembuatan *software* menggunakan aplikasi telegram bot terdapat beberapa tahapan - tahapan seperti pembuatan rancangan tampilan pada aplikasi telegram.

### 3.4 Pengujian Keseluruhan Sistem

Pengujian keseluruhan sistem dilakukan untuk mengetahui sistem kontrol otomatis pada lampu berbasis *Internet of Things* (IoT) bisa berfungsi dengan baik. Pengujian ini dijalankan dengan memberikan perintah pada aplikasi Telegram untuk menghidupkan atau mematikan lampu. Pengujian dilakukan satu persatu dari lampu 1 sampai dengan lampu 9. Kemudian, setelah dilakukan perintah mematikan atau menghidupkan lampu, akan muncul notifikasi *feedback* atau pemberitahuan bahwa lampu yang dikontrol sudah dalam keadaan mati atau hidup.

### 3.5 Pengumpulan Data Uji

Pengumpulan atau pengambilan data dilakukan adalah mengontrol *relay* dari aplikasi telegram untuk menghidupkan atau mematikan lampu. Kemudian data yang diambil berupa berapa lama waktu *delay* yang dibutuhkan untuk menghidupkan kesembilan lampu. Selanjutnya data yang di ambil adalah pengujian provider jaringan yang tersedia untuk mengetahui provider jaringan mana yang tercepat dalam melakukan pengontrolan lampu.

Dalam tahap pengumpulan data penulis menjalankan beberapa cara diantaranya:

1. Observasi: Observasi Lampu Parkir Polmanbabel konvensional yang Terpasang di Politeknik Manufaktur Negeri Bangka Belitung.
2. Studi Pustaka: Penulisan data dari buku referensi, jurnal, artikel internet serta prosiding sesuai dengan judul yang diangkat.

### 3.6 Analisis Data

Tahap Selanjutnya yaitu analisis data berdasarkan dari hasil pengujian sistem yang sudah dilakukan pada *hardware* maupun *software*. Pada tahap ini merupakan tahap yang tidak bisa dipisahkan dari pembuatan seluruh sistem. Pengujian dan pengambilan data diambil guna untuk memeriksa kekompakan atau kinerja antar komponen sistem yang telah diimplementasikan. Karena dengan melakukan pengujian serta pengambilan data pada sistem kemudian akan bisa diketahui apakah sistem tersebut bekerja sesuai dengan keinginan baik atau tidak. Tujuan utama yaitu

memastikan bahwa semua komponen dari sistem telah berfungsi sesuai dengan yang diharapkan.

### **3.7 Pembuatan Makalah Proyek Akhir**

Tahap akhir dari Proses pembuatan proyek akhir ini adalah pembuatan artikel dari hasil yang mempunyai hubungan dengan pembuatan tugas akhir dimulai dari pendahuluan, landasan teori, metode pelaksanaan, pembahasan, serta kesimpulan dan saran.



## BAB IV

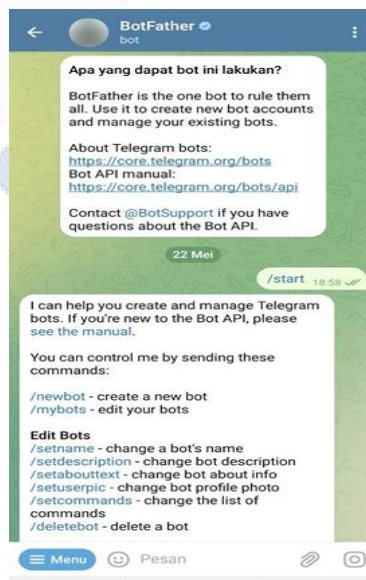
### PEMBAHASAN

Pada bab ini akan dijelaskan mengenai proses dalam pembuatan proyek akhir berjudul “SISTEM KONTROL LAMPU PARKIR POLMAN BABEL BERBASIS *INTERNET OF THINGS* (IoT)” mulai dari proses pembuatan sistem kontrol menggunakan IoT sampai dengan pengujian alat.

#### 4.1 Pembuatan *Software*

Pada tahap pembuatan Bot Father di Telegram dilakukan tahap-tahap berikut ini:

- Buka aplikasi Telegram dan cari “*BotFather*” di kotak pencarian atau dikunjungi [t.me/BotFather](https://t.me/BotFather) langsung.
- Pilih *BotFather* dari hasil pencarian dan mulai obrolan dengan *BotFather*.
- Ketik “*/start*” untuk memulai interaksi dengan *BotFather* seperti Gambar 4.1 berikut ini.



Gambar 4.1 Tampilan *start* pada *BotFather*

- Selanjutnya *BotFather* akan memberikan daftar perintah yang dapat digunakan untuk mengelola bot. Adapun perintah yang umum digunakan antara lain:

1. `/Newbot`: membuat bot baru. Ikuti instruksi *BotFather* untuk memberikan nama dan *username* untuk bot. Setelah bot berhasil dibuat, *BotFather* akan memberikan token akses API yang unik. Pastikan untuk menyimpan token ini dengan aman karena akan membutuhkannya untuk berinteraksi dengan API Telegram.
2. `/Token`: mendapatkan token akses API untuk bot yang sudah ada. Jika sudah memiliki bot sebelumnya dan ingin mendapatkan token aksesnya, gunakan perintah ini.
3. `/Set name`: mengubah nama bot.

Selanjutnya melakukan pembuatan *software* tampilan kontrol lampu secara otomatis dapat dilihat pada Gambar 4.2 berikut ini.



Gambar 4.2 Tampilan Kontrol Lampu dengan Aplikasi Telegram bot

Cara kerja untuk pengoperasian sistem telegram bot sebagai berikut :

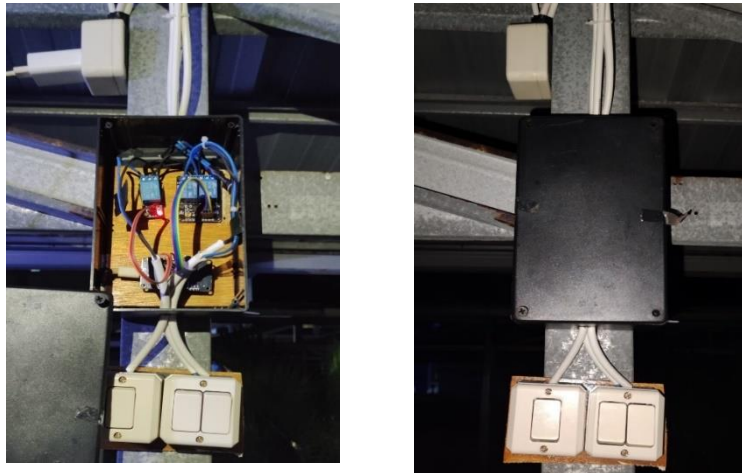
1. *User* harus terhubung dengan koneksi internet menggunakan Smartphone Android untuk memberikan perintah menyalakan atau mematikan lampu melalui Telegram chatbot
2. ESP8266 bertugas memberi instruksi yang masuk dari chatbot selanjutnya mengirimkan perintah ke *relay* agar kondisi lampu dapat menyala atau mati.
3. Ketika lampu sudah sesuai dengan perintah, kemudian lampu akan memberikan respons *feedback* ke chatbot melalui server Telegram, apakah saat ini lampu dalam keadaan menyala atau mati.
4. Respons dari lampu akan dibaca oleh ESP8266 yang dihubungkan dengan *Relay*, selanjutnya instruksi akan dikirimkan ke server dan *user* akan mendapat pesan notifikasi dari chatbot.
5. Kemudian hasil dari perintah respons lampu dari ESP8266 yang terkoneksi dengan internet dan diakses oleh server Telegram, selanjutnya *user* akan menerima respons berupa status dari keadaan lampu.

#### **4.2 Pembuatan *Hardware***

Pembuatan konstruksi kontrol lampu merupakan proses pembuatan desain konstruksi lampu tersebut mulai dari peletakan seluruh komponen yang akan digunakan, kemudian melakukan pembuatan konstruksi yang akan digunakan. Setelah melakukan pembuatan konstruksi selesai, Selanjutnya melakukan perakitan dan peletakan alat di dalam tempat konstruksi yaitu pada *black box*.

Pembuatan *hardware* dilakukan dengan dengan memasang ESP8266 dan juga *relay* pada *black box* serta pemasangan lampu yang digunakan agar terhubung ke *relay* untuk dikontrol.

Pembuatan hardware dilakukan dengan menggabungkan perangkat antar komponen kedalam *black box*. Adapun kontroler yang digunakan dapat dilihat pada Gambar 4.3 di bawah ini.



Gambar 4.3 Pembuatan *Hardware*


### 4.3 Pengujian Sistem

Pengujian sistem merupakan kegiatan melakukan pengujian terhadap sistem yang telah dibuat dengan tujuan untuk mengetahui apakah sistem yang dibangun sudah sesuai dengan rencana yang telah dibuat. Proses pengujian dilakukan agar aplikasi yang telah dibuat sesuai dengan yang diharapkan. Tahap pengujian dilakukan dengan metode *black box testing* yang terfokus pada fungsional *software*.

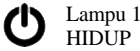
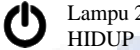

Berikut ini adalah tabel pengujian *black box* berdasarkan sistem kendali lampu berbasis *Internet of Things* (IoT) dengan menggunakan ESP8266 untuk fungsi tombol lampu 1 dan 2 didalam aplikasi telegram, yaitu sebagai berikut:

Tabel 4.1 Pengujian Fungsi Tombol Lampu 1 Dan 2 Untuk Menyalakan dan Mematikan Lampu Dalam Aplikasi Telegram

NO	Skenario Pengujian	Hasil yang diiharapkan	Kesimpulan
1.	Tekan tombol 1 untuk pertama kali	Status lampu 1 berubah menjadi "Hidup".	Tombol berfungsi


Lampu 1  
HIDUP


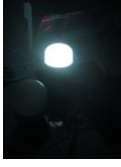








NO	Skenario Pengujian	Hasil yang diiharapkan	Kesimpulan
2.	Tekan tombol 1 untuk kedua kali	Status lampu 1 “Hidup” berubah menjadi “Mati” 	Tombol berfungsi
3.	Tekan tombol 2 untuk pertama kali	Status lampu 2 berubah menjadi “Hidup”. 	Tombol berfungsi
4.	Tekan tombol 2 untuk kedua kali	Status lampu 2 “Hidup” berubah menjadi “Mati” 	Tombol berfungsi

Pada pengujian tabel 4.1 ini dilakukan proses menyalakan dan mematikan lampu pada aplikasi telegram dengan tampilan status pada aplikasi untuk mendapatkan status dari proses menyalakan dan mematikan lampu.

Berikut ini adalah tabel pengujian *black box* berdasarkan sistem kendali lampu berbasis *Internet Of Things (IoT)* dengan menggunakan ESP8266 untuk fungsi tombol lampu 1 dan 2 secara langsung dengan aplikasi Telegram, yaitu sebagai berikut:

Tabel 4.2 Pengujian Fungsi Tombol Lampu 1 dan 2 Menyalakan dan Mematikan Lampu Secara Langsung Dengan Telegram

NO	Skenario Pengujian	Hasil yang diharapkan	Kesimpulan
1.	Status lampu 1 “Hidup” Test Case:  Lampu 1 HIDUP	Lampu 1 menyala Hasil test: 	Tombol berfungsi
2.	Status lampu 1 “Mati” Test Case:  Lampu 1 HIDUP	Lampu 1 mati Hasil test: 	Tombol berfungsi
3.	Status lampu 2 “Hidup” Test Case:  Lampu 2 HIDUP	Lampu 2 menyala Hasil test: 	Tombol berfungsi
4.	Status lampu 2 “Mati” Test Case:  Lampu 2 MATI	Lampu 2 mati Hasil test: 	Tombol berfungsi

Pengujian tabel 4.2 ini berfungsi untuk memastikan kondisi tombol pada aplikasi telegram pada saat menyalakan dan mematikan lampu dengan cara cek status lampu menyala dan lampu mati pada saat tombol ditekan.

Dari pengujian sistem dilakukan untuk memeriksa bagaimana sistem ini bekerja dengan baik atau tidak. Pengujian dilakukan dengan mengaktifkan seluruh komponen yang terhubung pada sistem, kemudian melakukan perintah kepada sistem sesuai dengan *coding* yang telah ditanamkan sebelumnya. Setelah dilakukan

pengujian sistem telah berhasil sesuai perintah yang diberikan dan menghasilkan *output* sesuai dengan yang diharapkan.

#### **4.4 Pengujian dengan Aplikasi Telegram**

Pengujian sistem kendali menggunakan aplikasi telegram dengan koneksi internet dari Handphone maupun koneksi wifi, Dimana komunikasi data antar mikrokontroler menggunakan internet atau *wireless* dan komunikasi data seperti ini pasti mempunyai waktu *delay* pada pengiriman data. Lama waktu *delay* pada pengiriman data tergantung pada koneksi internet yang sedang digunakan, semakin tidak bagus koneksi internet maka pengiriman data akan semakin memakan waktu, sebaliknya jika koneksi internet stabil maka pengiriman data akan semakin cepat. Dalam proses pengujian dengan menggunakan aplikasi telegram ini digunakan 9 buah lampu dimana setiap masing-masing lampu dilakukan percobaan sebanyak 10 kali percobaan menyalakan dan mematikan lampu. Berikut ini merupakan tabel data hasil pengujian waktu *delay* dengan menggunakan aplikasi telegram.

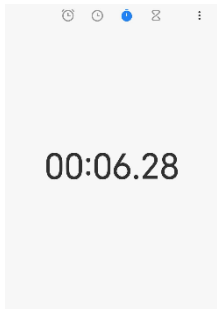
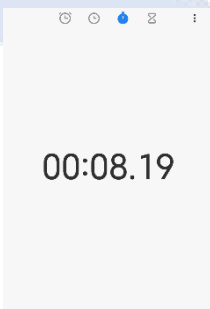
Tabel 4.3 Hasil Pengujian Dengan Aplikasi Telegram

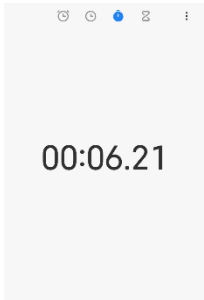
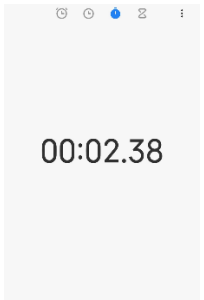
Delay (detik)																			
Pengujian	Lampu 1		Lampu 2		Lampu 3		Lampu 4		Lampu 5		Lampu 6		Lampu 7		Lampu 8		Lampu 9		
	ON	OFF	ON	OFF	ON	OFF	ON	OFF	ON	OFF	ON	OFF	ON	OFF	ON	OFF	ON	OFF	
1	6.28	8.11	6.16	2.34	6.22	5.33	8.34	12	5.17	5.11	1	6	11	1	5	4	6	10	
2	5.23	3.32	3.23	5.34	4.45	5.34	3.47	8.11	6.34	3.22	2.66	7.11	6.48	2.11	9.11	4.48	6.95	2	
3	4.45	5.54	2.12	5.25	5.11	2.34	5.15	5.45	4.19	8.12	4	3.19	6.49	4.39	8.12	6.55	2.12	4.29	
4	6.34	7.43	5.43	6.25	5.55	5.45	6.73	6.34	4.40	7.34	3.12	3.18	6.29	5.39	4.38	7.12	5.34	5.39	
5	6.55	4.54	4.56	6.34	4.23	7.56	5.88	4.25	5.53	4.19	5.86	9.12	4.19	5.55	5.38	4.29	6.18	6.29	
6	3.33	4.34	7.12	2.67	6.67	5.45	4.46	4.14	8.19	3.19	5.16	8.47	3.19	6.40	6.48	5.38	5.38	5.28	
7	4.21	8.15	3.45	7.12	5.14	6.47	6.46	4.33	3.18	4.38	4.28	5.28	4.39	6.48	6.23	5	6.18	7.29	
8	5.32	7.19	5.15	6.45	4.35	8.45	5.56	5.15	4.38	7.11	6.39	5.45	5.49	6.48	6.49	7.87	5.20	6.29	
9	7.21	5.18	2.15	5.35	3.56	5.78	7.12	4.79	5.27	4.27	4.49	5.45	6.59	6.66	7.49	6.49	5.32	6.29	
10	4.12	7.56	3.56	6.11	6.11	5.17	6.45	5.18	4.12	4.23	2.49	6	6.59	7	7.77	7	5.28	5.29	
Rata-rata	5,3	6.13	4.65	5.32	4.88	5.63	5.96	5.74	5.77	5.11	3.94	5.92	6.07	5.14	6.64	5.81	5.39	6.36	

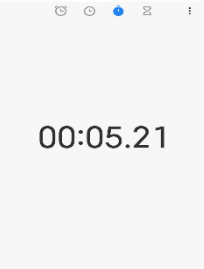
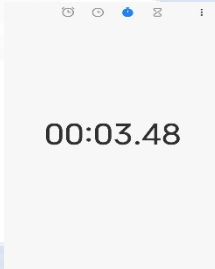
Pada hasil pengujian yang telah dilakukan pada Tabel 4.3 data menunjukkan bahwa rentan waktu penggunaan sistem *IoT* pada aplikasi telegram berbeda – beda. Pengujian pada aplikasi telegram dikatakan gagal apabila pengiriman data atau perintah tidak sesuai yang diharapkan dan dapat dilihat pada pesan *feedback* pada aplikasi telegram. Berdasarkan dari beberapa data yang didapatkan untuk menyalakan lampu 1 6.28 detik, 5.23 detik, 4.45 detik dan 6.34 detik. Dari 4 percobaan menyalakan lampu 1 memiliki rentan waktu yang berbeda antara yang satu dan lainnya hal ini tentu bisa saja terjadi karena dalam mengaktifkan lampu memerlukan koneksi internet yang stabil dan tentu pada koneksi internet yang stabil sekalipun rentan waktu tetap berbeda. Salah satu faktor adalah gangguan dari perangkat lain, interferensi sinyal atau kondisi lainnya pada lingkungan dapat mempengaruhi kinerja perangkat *IoT* dan menyebabkan variasi waktu *delay*.

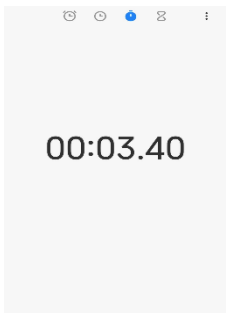
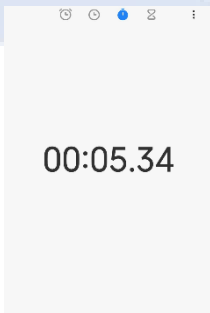
Berikut ini adalah tabel pengujian *black box* berdasarkan sistem kendali lampu berbasis *Internet Of Things* (IoT) dengan menggunakan ESP8266 untuk fungsi tombol lampu 1 dan 2 didalam aplikasi telegram pengujian berdasarkan *delay* waktu yang diperlukan untuk menghidupkan dan mematikan lampu yaitu sebagai berikut:

Tabel 4.4 Pengujian Delay Lampu 1 dan 2 didalam Aplikasi Telegram

Pengujian	Lampu 1	Lampu 2	Kesimpulan
	Delay lampu 1 menyala Hasil test:	Delay lampu 2 menyala Hasil test:	Lampu 2 lebih lama menyala nya dibandingkan lampu 1
1.			

Pengujian	Lampu 1	Lampu 2	Kesimpulan
	Delay lampu 1 pada saat mati. Hasil test:	Delay lampu 2 pada saat mati. Hasil test:	Lampu 2 lebih cepat dimatikan dibandingkan lampu 1
			

Pengujian	Lampu 1	Lampu 2	Kesimpulan
	Delay lampu 1 menyala Hasil test:	Delay lampu 2 menyala Hasil test:	Lampu 1 lebih lama menyala nya dibandingkan lampu 2
			

2.	Delay lampu 1 pada saat mati Hasil test:	Delay lampu 2 pada saat mati Hasil test:	Lampu 1 lebih cepat dimatikan dibandingkan lampu 2
			

Pada hasil pengujian yang telah dilakukan pada Tabel 4.4 dimana menggunakan 2 buah lampu dengan 2 kali percobaan dimana percobaan ini untuk mengambil nilai

*delay* pada lampu 1 dan 2 sebagai sampel pengambilan nilai *delay* dalam menyalakan dan mematikan lampu.

#### **4.5 Pengujian Kontrol Lampu**

Pengujian merupakan bagian yang tidak dapat dipisahkan dalam membangun sebuah sistem. Pengujian dilakukan untuk memeriksa kinerja antar komponen yang telah diimplementasikan. Tujuan utama dilakukan pengujian agar dapat mengetahui apakah sistem yang telah dibuat tersebut bekerja sesuai yang diinginkan atau tidak. Pengujian juga diartikan sebagai proses untuk memastikan apakah terlaksana penerapan sistem itu dengan baik atau tidak. Dari pengujian dan implementasi sistem penulis dapat mengetahui kesalahan apa yang terjadi seperti kekeliruan penulisan sintaks yang membuat sistem tidak bekerja sesuai dengan harapan pengguna. Pengujian ini dengan menggunakan metode *black box* untuk mengetahui dan melatih syarat fungsional suatu program.

Pengujian ini dilakukan untuk memberi perintah pada *relay* dengan cara mengontrol nyala lampu dengan aplikasi Telegram. Tujuan dari pengujian ini untuk memastikan fungsi sistem kontrol lampu apakah berfungsi dengan baik. Pengujian ini dilakukan dengan dua metode yaitu, pengujian kontrol menggunakan aplikasi telegram untuk mengetahui tingkat keberhasilan kontrol lampu. Yang kedua metode pengujian menggunakan provider, metode pengujian ini dilakukan untuk mengetahui respon *delay* dari beberapa provider yang digunakan.

Tabel 4.5 Hasil Pengujian Kontrol Lampu dengan Aplikasi Telegram

Pengujian	Relay 1		Relay 2		Relay 3		Relay 4		Relay 5		Relay 6		Relay 7		Relay 8		Relay 9	
	ON	OFF	ON	OFF	ON	OFF	ON	OFF	ON	OFF	ON	OFF	ON	OFF	ON	OFF	ON	OFF
1	Berhasil	Berhasil	Berhasil	Berhasil	Berhasil	Berhasil	Berhasil	Berhasil	Berhasil	Berhasil	Berhasil	Berhasil	Berhasil	Berhasil	Berhasil	Berhasil	Berhasil	Berhasil
2	Berhasil	Berhasil	Berhasil	Berhasil	Berhasil	Berhasil	Berhasil	Berhasil	Berhasil	Berhasil	Berhasil	Berhasil	Berhasil	Berhasil	Berhasil	Berhasil	Berhasil	Berhasil
3	Berhasil	Berhasil	Berhasil	Berhasil	Berhasil	Berhasil	Berhasil	Berhasil	Berhasil	Berhasil	Berhasil	Berhasil	Berhasil	Berhasil	Berhasil	Berhasil	Berhasil	Berhasil
4	Berhasil	Berhasil	Berhasil	Berhasil	Berhasil	Berhasil	Berhasil	Berhasil	Berhasil	Berhasil	Berhasil	Berhasil	Berhasil	Berhasil	Berhasil	Berhasil	Berhasil	Berhasil
5	Berhasil	Berhasil	Berhasil	Berhasil	Berhasil	Berhasil	Berhasil	Berhasil	Berhasil	Berhasil	Berhasil	Berhasil	Berhasil	Berhasil	Berhasil	Berhasil	Berhasil	Berhasil
6	Berhasil	Berhasil	Berhasil	Berhasil	Berhasil	Berhasil	Berhasil	Berhasil	Berhasil	Berhasil	Berhasil	Berhasil	Berhasil	Berhasil	Berhasil	Berhasil	Berhasil	Berhasil
7	Berhasil	Berhasil	Berhasil	Berhasil	Berhasil	Berhasil	Berhasil	Berhasil	Berhasil	Berhasil	Berhasil	Berhasil	Berhasil	Berhasil	Berhasil	Berhasil	Berhasil	Berhasil
8	Berhasil	Berhasil	Berhasil	Berhasil	Berhasil	Berhasil	Berhasil	Berhasil	Berhasil	Berhasil	Berhasil	Berhasil	Berhasil	Berhasil	Berhasil	Berhasil	Berhasil	Berhasil
9	Berhasil	Berhasil	Berhasil	Berhasil	Berhasil	Berhasil	Berhasil	Berhasil	Berhasil	Berhasil	Berhasil	Berhasil	Berhasil	Berhasil	Berhasil	Berhasil	Berhasil	Berhasil



Berdasarkan hasil pengujian pada tabel 4.5 merupakan data hasil yang didapatkan dari pengontrolan relay dari aplikasi telegram. Pada percobaan ini user mencoba memberi perintah untuk menghidupkan dan mematikan relay untuk mengetahui apakah perintah tersebut sudah sesuai dengan yang diharapkan atau tidak. Dari 9 percobaan yang telah dilakukan hasil yang didapatkan bahwa relay dapat dikontrol dengan baik dan tingkat keberhasilan kontrol relay adalah 100% dan sistem sudah sesuai dengan apa yang diharapkan.

#### 4.6 Pengujian Menggunakan Provider

Pada pengujian ini menggunakan beberapa jaringan provider yang diambil dari wifi maupun dari beberapa jaringan tersedia lainnya yang biasa digunakan. Tujuan pengujian ini berguna untuk mengetahui sistem yang dipakai ini berfungsi lebih baik/lebih efisien dengan menggunakan wifi dari maupun dari jaringan provider lainnya.

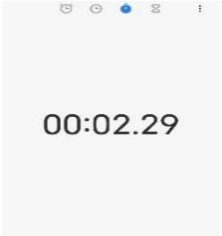
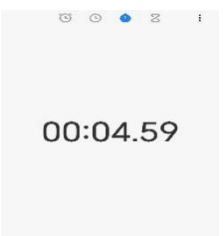
Tabel 4.6 Hasil pengujian dengan beberapa Provider

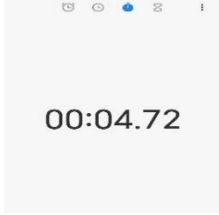
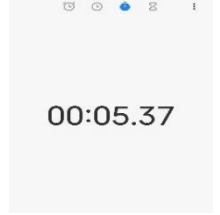
Nama Jaringan Internet	Keadaan Lampu	Lampu 1	Lampu 2	Lampu 3	Lampu 4	Lampu 5	Lampu 6	Lampu 7	Lampu 8	Lampu 9
		Delay(Detik)								
Wifi Indihome	ON	2.29	4.59	4.45	3.1	2.11	3.3	4.40	4.38	4.29
	OFF	4.72	2.37	4.45	5.34	3.47	8.11	6.34	6.27	5.39
Ichonnect	ON	3.13	4.42	5.11	2.34	5.15	5.45	4.19	5.23	6.29
	OFF	5.59	6.50	5.55	5.45	6.73	6.34	4.40	4.45	5.28
Smartfren	ON	7.77	6.63	5.15	6.45	4.35	8.45	5.56	5.15	7.29
	OFF	7.54	3.24	2.15	5.35	3.56	5.78	7.12	4.79	6.29
Telkomsel	ON	3.35	3.23	3.56	6.11	6.11	5.17	6.45	5.18	6
	OFF	3.18	5.54	4.65	5.32	4.88	5.63	5.96	5.74	5.29
Indosat	ON	5.55	4.66	4.56	6.34	4.23	7.56	5.88	5.39	4.18
	OFF	6.72	5,34	6.13	4.65	5.32	4.88	5.63	5.96	5.74
XL Axiata	ON	7.45	3.39	4.34	7.12	2.67	6.67	5.45	4.46	4.19
	OFF	7.37	6.39	5.17	6.45	5.18	4.12	4.23	2.49	6

Pengujian sistem ini menggunakan beberapa jenis provider yang dapat dilihat di Tabel 4.6. Dari data tabel tersebut dapat dianalisa bahwa sistem lebih baik atau lebih efisien dengan sumber dari provider wifi indihome, dimana waktu respon alat ketika menggunakan wifi indihome waktu *delay* kurang dari 5 detik dengan kecepatan menyalakan lampu dengan *delay* waktu tercepat 2.1 detik dan mematikan lampu dengan *delay* 2.37 detik. Berbeda ketika menggunakan provider jaringan lainnya waktu *delay* relatif lebih dari 7 detik dan tidak stabil karena sinyal dari hotspot HP menggunakan provider relatif tidak stabil. Data ini menunjukkan pengujian antara provider satu dan provider lainnya memiliki perbedaan yang signifikan karena kestabilan jaringan yang digunakan oleh perangkat dapat berbeda dari waktu ke waktu dan tentu jarak antara perangkat ke router juga dapat mempengaruhi kecepatan responnya.

Berikut ini adalah tabel pengujian *black box* berdasarkan sistem kendali lampu otomatis berbasis *Internet Of Things* (IoT) dengan menggunakan ESP8266 pengujian perkiraan kecepatan *delay* menyalakan dan mematikan lampu 1 dan lampu 2 saat *smartphone* terkoneksi ke Wi-Fi Indihome, Ichonnect, Smartfren, Telkomsel, Indosat, dan XL-Axiata.

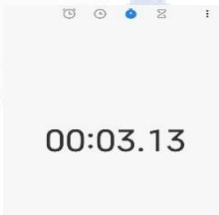
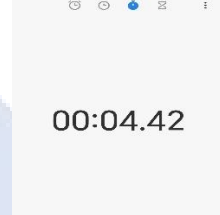
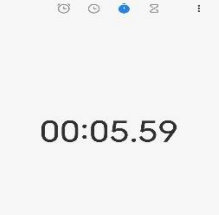
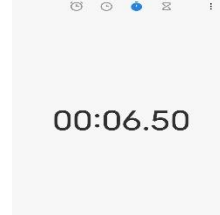
Tabel 4.7 Pengujian Kecepatan *delay* Menyalakan dan Mematikan Lampu 1 Dan Lampu 2 dengan Menggunakan Wi-Fi Indihome

No	Skenario Pengujian	Lampu 1	Lampu 2	Kesimpulan
1	Saat terkoneksi dengan Wi-Fi Indihome	Lampu 1 (ON) Hasil test: 	Lampu 2 (ON) Hasil test: 	Lampu 2 lebih lama menyala nya dibandingkan lampu 1

2	Saat terkoneksi dengan Wi-Fi Indihome	Lampu 1 (OFF) Hasil test: 	Lampu 2 (OFF) Hasil test: 	Lampu 1 lebih cepat dimatikan dibandingkan lampu 2
---	---------------------------------------	---	--	--

Pengujian tabel 4.7 ini menggunakan Wi-Fi untuk menguji kekuatan sinyal dari Wi-Fi Indihome serta *delay* yang diberikan oleh sinyal dari Wi-Fi Indihome dengan 2 kali percobaan dengan nilai 2.29 detik paling rendah untuk menyalakan lampu dan 4.72 detik untuk mematikan lampu.

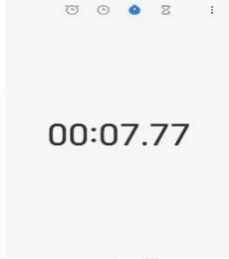
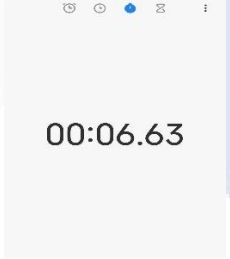
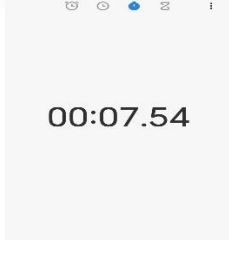
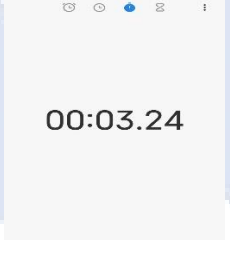
Tabel 4.8 Pengujian Kecepatan *delay* Menyalakan dan Mematikan Lampu 1 Dan Lampu 2 dengan Menggunakan Ichonnect

No	Skenario Pengujian	Lampu 1	Lampu 2	Kesimpulan
1	Saat terkoneksi dengan Ichonnect	Lampu 1 (ON) Hasil test: 	Lampu 2 (ON) Hasil test: 	Lampu 2 lebih lama menyala nya dibandingkan lampu 1
2	Saat terkoneksi dengan Ichonnect	Lampu 1 (OFF) Hasil test: 	Lampu 2 (OFF) Hasil test: 	Lampu 1 lebih cepat dimatikan dibandingkan lampu 2

Pengujian tabel 4.8 ini menggunakan sinyal Ichonnect untuk menguji kekuatan sinyal dari Ichonnect serta *delay* yang diberikan oleh sinyal dari Ichonnect dengan

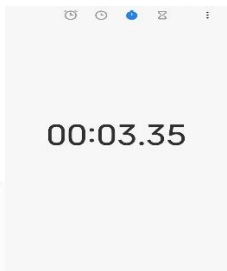
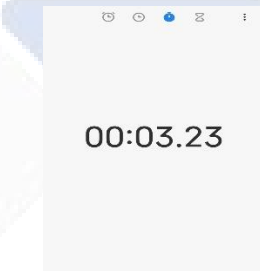
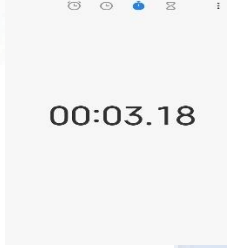
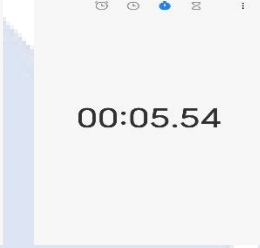
2 kali percobaan dengan nilai 3.13 detik paling rendah untuk menyalakan lampu dan 5.59 detik untuk mematikan lampu.

Tabel 4.9 Pengujian Kecepatan *delay* Menyalakan dan Mematikan Lampu 1 Dan Lampu 2 dengan Menggunakan Smartfren

No	Skenario Pengujian	Lampu 1	Lampu 2	Kesimpulan
1	Saat terkoneksi dengan Provider Smartfren	Lampu 1 (ON) Hasil test: 	Lampu 2 (ON) Hasil test: 	Lampu 1 lebih lama menyala nya dibandingkan lampu 2
2	Saat terkoneksi dengan Provider Smartfren	Lampu 1 (OFF) Hasil test: 	Lampu 2 (OFF) Hasil test: 	Lampu 2 lebih cepat dimatikan dibandingkan lampu 1

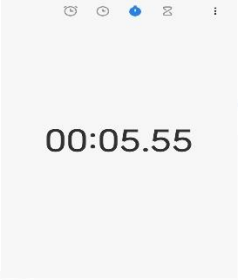
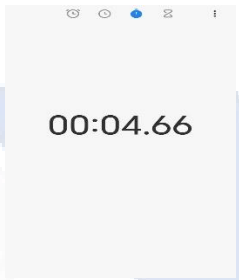
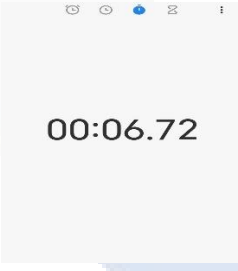
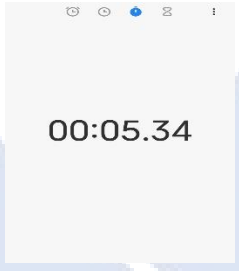
Pengujian tabel 4.9 ini menggunakan sinyal Smartfren untuk menguji kekuatan sinyal dari Smartfren serta *delay* yang diberikan oleh sinyal dari Smartfren dengan 2 kali percobaan dengan nilai 6.63 detik paling rendah untuk menyalakan lampu dan 3.24 detik untuk mematikan lampu.

Tabel 4.10 Pengujian Kecepatan *delay* Menyalakan dan Mematikan Lampu 1 Dan Lampu 2 dengan Menggunakan Telkomsel

No	Skenario Pengujian	Lampu 1	Lampu 2	Kesimpulan
1	Saat terkoneksi dengan Provider Telkomsel	Lampu 1 (ON) Hasil test: 	Lampu 2 (ON) Hasil test: 	Lampu 1 lebih lama menyala nya dibandingkan lampu 2
2	Saat terkoneksi dengan Provider Telkomsel	Lampu 1 (OFF) Hasil test: 	Lampu 2 (OFF) Hasil test: 	Lampu 1 lebih cepat dimatikan dibandingkan lampu 2

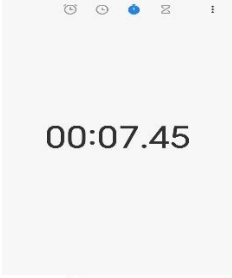
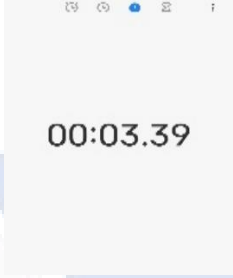
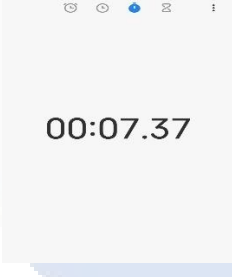
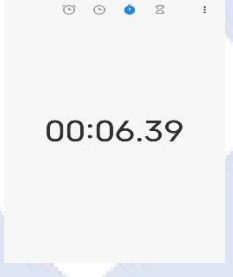
Pengujian tabel 4.10 ini menggunakan sinyal Telkomsel untuk menguji kekuatan sinyal dari Telkomsel serta *delay* yang diberikan oleh sinyal dari Telkomsel dengan 2 kali percobaan dengan nilai 3.23 detik paling rendah untuk menyalakan lampu dan 3.18 detik untuk mematikan lampu.

Tabel 4.11 Pengujian Kecepatan *delay* Menyalakan dan Mematikan Lampu 1 Dan Lampu 2 dengan Menggunakan Indosat

No	Skenario Pengujian	Lampu 1	Lampu 2	Kesimpulan
1	Saat terkoneksi dengan Provider Indosat	Lampu 1 (ON) Hasil test: 	Lampu 2 (ON) Hasil test: 	Lampu 1 lebih lama menyala nya dibandingkan lampu 2
2	Saat terkoneksi dengan Provider Indosat	Lampu 1 (OFF) Hasil test: 	Lampu 2 (OFF) Hasil test: 	Lampu 2 lebih cepat dimatikan dibandingkan lampu 1

Pengujian tabel 4.11 ini menggunakan sinyal Indosat untuk menguji kekuatan sinyal dari Indosat serta *delay* yang diberikan oleh sinyal dari Indosat dengan 2 kali percobaan dengan nilai 4.66 detik paling rendah untuk menyalakan lampu dan 5.34 detik untuk mematikan lampu.

Tabel 4.12 Pengujian Kecepatan *delay* Menyalakan dan Mematikan Lampu 1 Dan Lampu 2 dengan Menggunakan XL-Axiata

No	Skenario Pengujian	Lampu 1	Lampu 2	Kesimpulan
1	Saat terkoneksi dengan Provider XL-Axiata	Lampu 1 (ON) Hasil test: 	Lampu 2 (ON) Hasil test: 	Lampu 1 lebih lama menyala nya dibandingkan lampu 2
2	Saat terkoneksi dengan Provider XL-Axiata	Lampu 1 (OFF) Hasil test: 	Lampu 2 (OFF) Hasil test: 	Lampu 2 lebih cepat dimatikan dibandingkan lampu 1

Pengujian tabel 4.12 ini menggunakan sinyal XL-Axiata untuk menguji kekuatan sinyal dari XL-Axiata serta *delay* yang diberikan oleh sinyal dari XL-Axiata dengan 2 kali percobaan dengan nilai 3.39 detik paling rendah untuk menyalakan lampu dan 6.39 detik untuk mematikan lampu.

#### 4.7 Analisa Seluruh Pengujian

Dari tiap tiap pengujian bagian sistem yang telah dilakukan atau dengan kata lain pengujian ini dilakukan terhadap keseluruhan sistem yang telah dibuat. Sistem bekerja dengan baik rata-rata respon penyalaaan lampu berbeda-beda yaitu dengan media aplikasi telegram 10 detik sedangkan untuk mematikan lampu 6.64 detik,

kemudian dari pengujian menggunakan media berbagai jenis provider dengan rata-rata menyalakan lampu tercepat adalah dengan provider Wifi indihome yaitu dengan menyalakan lampu kurang dari 5 detik dengan *delay* tercepat 2.1 detik sedangkan mematikan lampu dengan *delay* 2.7 detik. Berdasarkan dari pengujian yang telah dijelaskan secara keseluruhan maka kesimpulan yang didapat konsistensi waktu *delay* yang didapat dari berbagai pengujian memberikan gambaran stabilitas kinerja sistem. Jika waktu *delay* bervariasi secara signifikan dari pengujian ke pengujian lain, dapat menunjukkan bahwa beban perbedaan kinerja tentang seberapa merespons perintah setelah menerima intruksi dari pengguna semakin rendah waktu *delay* rata-rata maka semakin responsif sistem ini bekerja dan waktu *delay* yang minimum menunjukkan bahwa kinerja terbaik yang dapat dicapai sistem.



## **BAB V**

### **PENUTUP**

#### **5.1 Kesimpulan**

Berdasarkan hasil pengujian dan analisis fungsionalitas alat pada proyek akhir berjudul “SISTEM KONTROL LAMPU PARKIR POLMAN BABEL BERBASIS *INTERNET OF THINGS* (IoT)” ini maka bisa ditarik kesimpulan sebagai berikut:

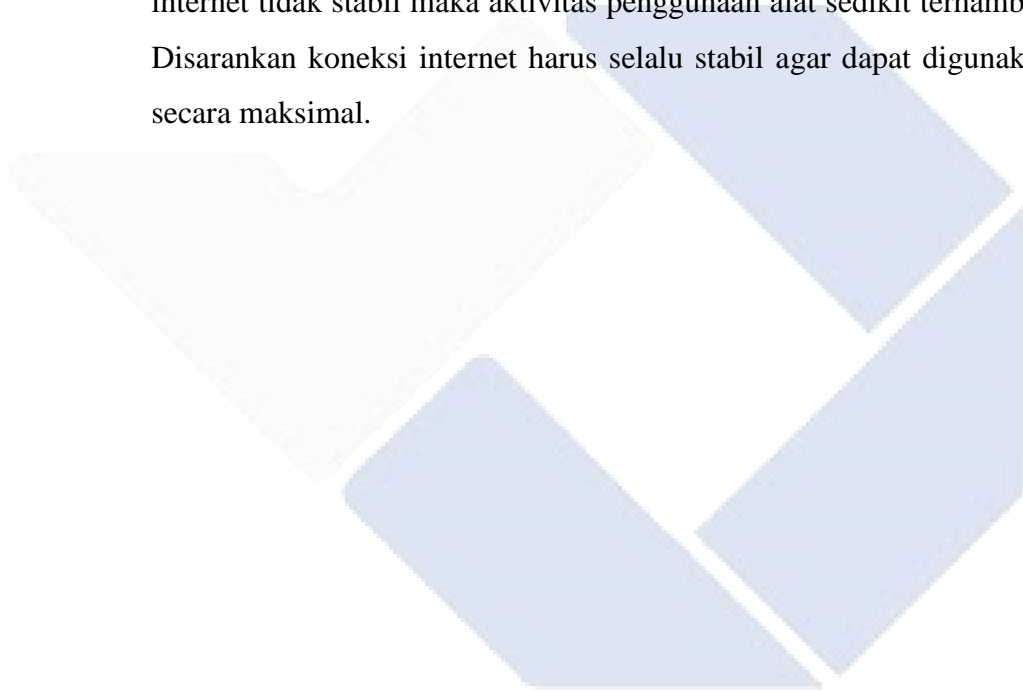
1. Sistem kontrol lampu berbasis *Internet Of Things (IoT)* dengan menggunakan ESP8266 dirancang dengan berbagai komponen *hardware* dan dukungan *software* sehingga tersusun menjadi sebuah sistem kendali lampu yang dapat dikontrol menggunakan aplikasi telegram dengan syarat terkoneksi ke Internet dan koneksi stabil sehingga meminimalisir terjadinya kegagalan sistem atau kecepatan delay yang maksimum.
2. Pengontrolan lampu dilakukan dengan cara menekan tombol L1/L2/L3 untuk menyalakan atau mematikan lampu 1 sampai lampu 3, tombol L4/L5/L6 untuk menyalakan dan mematikan lampu 4 sampai lampu 6, tombol L7/L8/L9 untuk menyalakan dan mematikan lampu 7 sampai lampu 9.

#### **5.2 Saran**

Dari pembuatan proyek akhir yang telah dilakukan, penulis memiliki beberapa saran yang ingin disampaikan untuk mengembangkan proyek akhir ini dengan harapan agar lebih baik lagi kedepannya, yaitu:

1. Aplikasi Telegram bot ini memiliki kekurangan dimana masih banyak kekurangan fitur salah satunya masalah keamanan karena siapa saja dapat meng-*add* dan menjalankan perintah pada program sehingga dapat dikendalikan oleh siapa saja tanpa sepengetahuan pembuat. Kedepannya diharapkan bisa menggunakan aplikasi lain yang fitur nya lebih banyak untuk kebutuhan penggunaan.

2. Alat ini hanya sekedar dapat memberikan perintah menyalakan dan mematikan lampu, desain yang tidak terlalu rumit kedepannya mungkin bisa ditambahkan fitur dan fungsi lain seperti menyalakan dan mematikan lampu dengan *timer* dan menambahkan sensor cahaya dengan harapan apabila koneksi internet sedang buruk dan tidak stabil jadi sensor dapat menyesuaikan dengan kondisi cahaya dilokasi apabila sudah gelap dan otomatis hidup apabila sudah terang sehingga tercapainya efisiensi dan hemat energi.
3. Kendala utama dari alat ini adalah koneksi internet apabila koneksi internet tidak stabil maka aktivitas penggunaan alat sedikit terhambat. Disarankan koneksi internet harus selalu stabil agar dapat digunakan secara maksimal.



## DAFTAR PUSTAKA

- [1] Kartika, M. Y. Iqbar and K. Paranita, "Rancang Bangun Lampu Portable Otomatis Menggunakan Rtc Berbasis Arduino," *Jurnal Ilmiah Teknik Informatika*, vol. 14, no. 1, pp. 61-72, 2020.
- [2] A. A. G. S. Utama, N. M. Janani, S. T. N. A. Wulandari and B. , "Automation Of Electrical Energy Savings System: Hemat Listrik, Hemat Biaya," *Jurnal Pendidikan Ekonomi*, vol. 6, no. 2, pp. 79-87, 2018.
- [3] S. J. H. P. Sitorus and R. S. Saragih, "Perancangan Pengontrol Lampu Rumah Miniatur Dengan Menggunakan Micro Controler Arduino Berbasis Android," *Jurnal Bisantra Informatika (JBI)*, vol. 4, no. 1, pp. 1-11, 2020.
- [4] E. A. Satya, Y. Christiyono and M. Somantri, "Pengontrolan Lampu Melalui Internet Menggunakan Mikrokontroler Arduino Berbasis Android," *Transient*, vol. 5, no. 3, pp. 359-364, 2016.
- [5] F. Abdaoe, H. Setiawan and K. Perdana, "Sistem Kendali Lampu Otomatis Berbasis Iot(Internet Of Things)Menggunakan Nodencum," *Bangkit Indonesia*, vol. 9, no. 1, pp. 76-91, 2020.
- [6] D. Setiawan, "Sistem Kontrol Lampu Menggunakan Metode Manual & Otomatis Berbasis Handphone," *SainETIn (Jurnal Sain, Energi, Teknologi & Industri)*, vol. 1, no. 2, pp. 24-31, 2017.
- [7] S. and J. Jordan, "Rancang Bangun Sistem Lampu Otomatis Berbasis Internet of Things dengan Aplikasi Raspberry Pi," *Jurnal Pendidikan Sains dan Komputer*, vol. 2, no. 2, pp. 345-349, 2022.
- [8] A. Virginia, P. Handoko and H. Hermawan, "Rancang Bangun Sistem Kontrol Lampu Berbasis Arduino Mega 2560," *Jurnal of pembangunan jaya university*, vol. 5, no. 2, pp. 146-155, 2018.
- [9] M. H. Abdullah, "Rancang Bangun Sistem Kontrol Lampu Listrik Menggunakan Remote Berbasis Mikrokontroler ATMega 8535," *Jurnal ilmiah ILKOMINFO*, vol. 2, no. 1, pp. 40-47, 2019.
- [10] L. Y. I. Frare and . S. Ramos, "Rancang Bangun Alat Kontrol Lampu Jarak Jauh Menggunakan Mikrokontroler Berbasis Website," *Sistem Komputer & Teknologi Intelegensi Artifisial (SIKOMTIA)*, vol. 1, no. 1, pp. 78-90, 2022.
- [11] I. Update, "Datashet NodeMCU ESP8266 Lengkap dengan Pin dan Cara Akses," Indobot Academy, 27 Maret 2023. [Online]. Available: <https://indobot.co.id/blog/datasheet-nodemcu-esp8266-lengkap-dengan-pin-dan-cara-akses/>. [Accessed 29 Mei 2023].
- [12] ardutech, "Kontrol Lampu WiFi dengan Blynk dan Arduino," 7 September 2022. [Online]. Available: <https://www.ardutech.com/kontrol-lampu-wifi-dengan-blynk-dan-arduino/>. [Accessed 5 Juli 2023].
- [13] Askix, "Simple Led de Control con Blynk y NodeMCU Esp8266 12E (2 / 8 paso)," 2023. [Online]. Available: <https://www.askix.com/simple-led-de-control-con-blynk-y-nodemcu-esp8266-12e.html>. [Accessed 10 Juli 2023].

- [14] D. Susilo, C. Sari and G. W. Krisna, "Sistem Kendali Lampu pada Smart Home Berbasis IoT(Internet of Things)," *Jurnal ELECTRA : Electrical Engineering Articles*, vol. 2, no. 1, p. 23~30, 2021.
- [15] I. Update, "Tutorial IoT Mengendalikan Nyala Lampu Halaman dengan Telegram," 6 April 2023. [Online]. Available: <https://indobot.co.id/blog/tutorial-iot-mengendalikan-nyala-lampu-halaman-dengan-telegram/>. [Accessed 5 Juli 2023].



## LAMPIRAN 1

### DAFTAR RIWAYAT HIDUP

#### Data Pribadi

Nama Lengkap : Eby Jhoanca  
Tempat/Tanggal Lahir : Mentok, 8 Mei 2000  
Alamat Rumah : Kp. Senang Hati, Mentok  
No. Telepon :085378398901  
E-mail : JhoancaEby0808@gmail.com  
Jenis Kelamin : Laki-Laki



#### Riwayat Pendidikan

SDN 21 MUNTOK	2012
SMPN 1 MUNTOK	2015
SMAN 1 MUNTOK	2018
POLITEKNIK MANUFAKTUR NEGERI BANGKA BELITUNG	2023

#### Pendidikan Non- Formal

Praktik Kerja Lapangan di PT. Global Sahabat Otomasi, Jakarta Pusat.

## **DAFTAR RIWAYAT HIDUP**

### **Data Pribadi**

Nama Lengkap : Muhammmad Asqal  
Tempat/Tanggal Lahir : Mentok, 12 Juli 2000  
Alamat Rumah : Kp. Teluk Rubiah Laut  
No. Telepon : 087894745095  
E-mail : muhammadasqal12@gmail.com  
Jenis Kelamin : Laki-Laki



### **Riwayat Pendidikan**

SDN 21 MUNTOK	2012
SMPN 1 MUNTOK	2015
SMAN 1 MUNTOK	2018
POLITEKNIK MANUFAKTUR NEGERI BANGKA BELITUNG	2023

### **Pendidikan Non- Formal**

Praktik Kerja Lapangan di PT. Global Sahabat Otomasi, Jakarta Pusat.

## LAMPIRAN 2

### Program ESP8266 ESP8266

```
#include <CTBot.h>
#include <CTBotDataStructures.h>
#include <CTBotDefines.h>
#include <CTBotInlineKeyboard.h>
#include <CTBotReplyKeyboard.h>
#include <CTBotSecureConnection.h>
#include <CTBotStatusPin.h>
#include <CTBotWifiSetup.h>
#include <Utilities.h>

#include "CTBot.h"
CTBot myBot;
CTBotReplyKeyboard Tbl;
CTBotInlineKeyboard TbLp1on, TbLp1off, TbLp2on, TbLp2off, TbLp3on,
TbLp3off, TbLpAllon, TbLpAlloff;

#define Lampu1 D0
#define Lampu2 D1
#define Lampu3 D2
#define Lampu4 D3
#define Lampu5 D4
#define Lampu6 D5
#define Lampu7 D6
#define Lampu8 D7
#define Lampu9 D8

#define LAMPU_1_ON "Lampu_1_ON"
#define LAMPU_1_OFF "Lampu_1_OFF"
#define LAMPU_2_ON "Lampu_2_ON"
#define LAMPU_2_OFF "Lampu_2_OFF"
#define LAMPU_3_ON "Lampu_3_ON"
#define LAMPU_3_OFF "Lampu_3_OFF"
#define LAMPU_4_ON "Lampu_4_ON"
#define LAMPU_4_OFF "Lampu_4_OFF"
#define LAMPU_5_ON "Lampu_5_ON"
#define LAMPU_5_OFF "Lampu_5_OFF"
#define LAMPU_6_ON "Lampu_6_ON"
#define LAMPU_6_OFF "Lampu_6_OFF"
#define LAMPU_7_ON "Lampu_7_ON"
#define LAMPU_7_OFF "Lampu_7_OFF"
```

```
#define LAMPU_8_ON "Lampu_8_ON"
#define LAMPU_8_OFF "Lampu_8_OFF"
#define LAMPU_9_ON "Lampu_9_ON"
#define LAMPU_9_OFF "Lampu_9_OFF"

#define SEMUA_LAMPU_ON "Semua_Lampu_ON"
#define SEMUA_LAMPU_OFF "Semua_Lampu_OFF"
#define TIDAK "Tidak"

String Status_Lampu_1;
String Status_Lampu_2;
String Status_Lampu_3;
String Status_Lampu_4;
String Status_Lampu_5;
String Status_Lampu_6;
String Status_Lampu_7;
String Status_Lampu_8;
String Status_Lampu_9;

int Lampu_1 = HIGH;
int Lampu_2 = HIGH;
int Lampu_3 = HIGH;
int Lampu_4 = HIGH;
int Lampu_5 = HIGH;
int Lampu_6 = HIGH;
int Lampu_7 = HIGH;
int Lampu_8 = HIGH;
int Lampu_9 = HIGH;

bool TampilkanTombol;

String ssid = "TIDAK DIKETAHUI";
String pass = "aal12345";
String token = "5988485964:AAHGke4QPQTccKxnLAgtiO-Mh5pxXxknCWE";
const int id = 1318157633;

void setup() {
  Serial.begin(115200);

  Serial.println("Memulai Koneksi...");

  myBot.wifiConnect("TIDAK DIKETAHUI","aal12345");
  myBot.setTelegramToken("5988485964:AAHGke4QPQTccKxnLAgtiO-
Mh5pxXxknCWE");

  if (myBot.testConnection())
```



```
Serial.println("\nKoneksi Ke Telegram BOT Berhasil!");
Serial.println("\nTidak Terkoneksi Ke Telegram BOT");
```

```
pinMode(Lampu1, OUTPUT);
pinMode(Lampu2, OUTPUT);
pinMode(Lampu3, OUTPUT);
pinMode(Lampu4, OUTPUT);
pinMode(Lampu5, OUTPUT);
pinMode(Lampu6, OUTPUT);
pinMode(Lampu7, OUTPUT);
pinMode(Lampu8, OUTPUT);
pinMode(Lampu9, OUTPUT);
```

```
digitalWrite(Lampu1, HIGH);
digitalWrite(Lampu2, HIGH);
digitalWrite(Lampu3, HIGH);
digitalWrite(Lampu4, HIGH);
digitalWrite(Lampu5, HIGH);
digitalWrite(Lampu6, HIGH);
digitalWrite(Lampu7, HIGH);
digitalWrite(Lampu8, HIGH);
digitalWrite(Lampu9, HIGH);
```

```
Tbl.addButton("L1/L2/L3");
Tbl.addButton("L4/L5/L6");
Tbl.addButton("L7/L8/L9");
```

```
Tbl.addRow();
Tbl.addButton("Semua Lampu ON");
Tbl.addButton("Semua Lampu OFF");
Tbl.addRow();
Tbl.addButton("Cek Status");
Tbl.enableResize();
TampilkanTombol = false;
```

```
TbLp1on.addButton("✔💡 1", LAMPU_1_ON,
CTBotKeyboardButtonQuery);
TbLp1on.addButton("🚫💡 1", TIDAK, CTBotKeyboardButtonQuery);
```

```
TbLp1on.addButton("✔💡 2", LAMPU_2_ON,
CTBotKeyboardButtonQuery);
TbLp1on.addButton("🚫💡 2", TIDAK, CTBotKeyboardButtonQuery);
```

```
TbLp1on.addButton("✔💡 3", LAMPU_3_ON,
CTBotKeyboardButtonQuery);
```

```
TbLp1on.addButton("🚫💡 3", TIDAK, CTBotKeyboardButtonQuery);

TbLp2on.addButton("✅💡 4", LAMPU_4_ON,
CTBotKeyboardButtonQuery);
TbLp2on.addButton("🚫💡 4", TIDAK, CTBotKeyboardButtonQuery);
TbLp2on.addButton("✅💡 5", LAMPU_5_ON,
CTBotKeyboardButtonQuery);
TbLp2on.addButton("🚫💡 5", TIDAK, CTBotKeyboardButtonQuery);

TbLp2on.addButton("✅💡 6", LAMPU_6_ON,
CTBotKeyboardButtonQuery);
TbLp2on.addButton("🚫💡 6", TIDAK, CTBotKeyboardButtonQuery);

TbLp3on.addButton("✅💡 7", LAMPU_7_ON,
CTBotKeyboardButtonQuery);
TbLp3on.addButton("🚫💡 7", TIDAK, CTBotKeyboardButtonQuery);

TbLp3on.addButton("✅💡 8", LAMPU_8_ON,
CTBotKeyboardButtonQuery);
TbLp3on.addButton("🚫💡 8", TIDAK, CTBotKeyboardButtonQuery);

TbLp3on.addButton("✅💡 9", LAMPU_9_ON,
CTBotKeyboardButtonQuery);
TbLp3on.addButton("🚫💡 9", TIDAK, CTBotKeyboardButtonQuery);

TbLp1off.addButton("✅💡 1", LAMPU_1_OFF,
CTBotKeyboardButtonQuery);
TbLp1off.addButton("🚫💡 1", TIDAK, CTBotKeyboardButtonQuery);

TbLp1off.addButton("✅💡 2", LAMPU_2_OFF,
CTBotKeyboardButtonQuery);
TbLp1off.addButton("🚫💡 2", TIDAK, CTBotKeyboardButtonQuery);

TbLp1off.addButton("✅💡 3", LAMPU_3_OFF,
CTBotKeyboardButtonQuery);
TbLp1off.addButton("🚫💡 3", TIDAK, CTBotKeyboardButtonQuery);

TbLp2off.addButton("✅💡 4", LAMPU_4_OFF,
CTBotKeyboardButtonQuery);
TbLp2off.addButton("🚫💡 4", TIDAK, CTBotKeyboardButtonQuery);

TbLp2off.addButton("✅💡 5", LAMPU_5_OFF,
CTBotKeyboardButtonQuery);
```

```

    TbLp2off.addButton("🚫💡5", TIDAK, CTBotKeyboardButtonQuery);

    TbLp2off.addButton("✅💡6", LAMPU_6_OFF,
CTBotKeyboardButtonQuery);
    TbLp2off.addButton("🚫💡6", TIDAK, CTBotKeyboardButtonQuery);

    TbLp3off.addButton("✅💡7", LAMPU_7_OFF,
CTBotKeyboardButtonQuery);
    TbLp3off.addButton("🚫💡7", TIDAK, CTBotKeyboardButtonQuery);

    TbLp3off.addButton("✅💡8", LAMPU_8_OFF,
CTBotKeyboardButtonQuery);
    TbLp3off.addButton("🚫💡8", TIDAK, CTBotKeyboardButtonQuery);

    TbLp3off.addButton("✅💡9", LAMPU_9_OFF,
CTBotKeyboardButtonQuery);
    TbLp3off.addButton("🚫💡9", TIDAK, CTBotKeyboardButtonQuery);

    TbLpAllon.addButton("✅ Ya", SEMUA_LAMPU_ON,
CTBotKeyboardButtonQuery);
    TbLpAllon.addButton("❌ Tidak", TIDAK, CTBotKeyboardButtonQuery);

    TbLpAlloff.addButton("✅ Ya", SEMUA_LAMPU_OFF,
CTBotKeyboardButtonQuery);
    TbLpAlloff.addButton("❌ Tidak", TIDAK,
CTBotKeyboardButtonQuery);
}
void loop() {
TBMessage msg;
    if (myBot.getNewMessage(msg)) {
    if (msg.text.equalsIgnoreCase("/start")) {
        myBot.sendMessage(msg.sender.id, "Hello...\nWelcome to Polman Babel
auto parking light.\n\nGunakan tombol dibawah untuk mengontrol lampu 🗑️",
Tb1);
        Serial.println("\nUser memulai Bot \n");
        TampilkanTombol = true;
    } else if (msg.text.equalsIgnoreCase("L1/L2/L3")) {
        if ((Lampu_1 == LOW)&&(Lampu_2 == LOW)&&(Lampu_3 == LOW)){
            myBot.sendMessage(msg.sender.id, "Saat ini Lampu 1/2/3 dalam keadaan
ON\nApakah anda ingin mematikan Lampu 1/2/3?", TbLp1off);
            Serial.println("\nUser mengirim perintah 'Lampu 1/2/3' \nSaat ini Lampu
1/2/3 dalam keadaan ON \n");
        } else if ((Lampu_1 == HIGH)&&(Lampu_2 == HIGH)&&(Lampu_3 ==
HIGH)){

```

```

        myBot.sendMessage(msg.sender.id, "Saat ini Lampu 1/2/3 dalam keadaan OFF\nApakah anda ingin menyalakan Lampu 1/2/3?", TbLp1on);
        Serial.println("\nUser mengirim perintah 'Lampu 1/2/3' \nSaat ini Lampu 1/2/3 dalam keadaan OFF \n");
    }
}
else if (msg.text.equalsIgnoreCase("L4/L5/L6")) {
    if ((Lampu_4 == LOW)&&(Lampu_5 == LOW)&&(Lampu_6 == LOW)) {
        myBot.sendMessage(msg.sender.id, "Saat ini Lampu 4/5/6 dalam keadaan ON\nApakah anda ingin mematikan Lampu 4/5/6?", TbLp2off);
        Serial.println("\nUser mengirim perintah 'Lampu 4/5/6' \nSaat ini Lampu 4/5/6 dalam keadaan ON \n");
    } else if ((Lampu_4 == HIGH)&&(Lampu_5 == HIGH)&&(Lampu_6 == HIGH)){
        myBot.sendMessage(msg.sender.id, "Saat ini Lampu 4/5/6 dalam keadaan OFF\nApakah anda ingin menyalakan Lampu 4/5/6?", TbLp2on);
        Serial.println("\nUser mengirim perintah 'Lampu 4/5/6' \nSaat ini Lampu 4/5/6 dalam keadaan OFF \n");
    }
}
else if (msg.text.equalsIgnoreCase("L7/L8/L9")) {
    if ((Lampu_7 == LOW)&&(Lampu_8 == LOW)&&(Lampu_9 == LOW)) {
        myBot.sendMessage(msg.sender.id, "Saat ini Lampu 7/8/9 dalam keadaan ON\nApakah anda ingin mematikan Lampu 7/8/9?", TbLp3off);
        Serial.println("\nUser mengirim perintah 'Lampu 7/8/9' \nSaat ini Lampu 7/8/9 dalam keadaan ON \n");
    } else if ((Lampu_7 == HIGH)&&(Lampu_8 == HIGH)&&(Lampu_9 == HIGH)){
        myBot.sendMessage(msg.sender.id, "Saat ini Lampu 7/8/9 dalam keadaan OFF\nApakah anda ingin menyalakan Lampu 7/8/9?", TbLp3on);
        Serial.println("\nUser mengirim perintah 'Lampu 7/8/9' \nSaat ini Lampu 7/8/9 dalam keadaan OFF \n");
    }
}

} else if (msg.text.equalsIgnoreCase("Semua Lampu ON")) {
    myBot.sendMessage(msg.sender.id, "Apakah anda ingin menyalakan Semua Lampu?", TbLpAllon);
    Serial.println("\nUser ingin menyalakan semua lampu\n");
} else if (msg.text.equalsIgnoreCase("Semua Lampu OFF")) {
    myBot.sendMessage(msg.sender.id, "Apakah anda ingin mematikan Semua Lampu?", TbLpAlloff);
    Serial.println("\nUser ingin mematikan semua lampu\n");
} else if (msg.text.equalsIgnoreCase("Cek Status")) {
    if (Lampu_1 == LOW) {
        Status_Lampu_1 = "Lampu 1 ON";
    } else {

```

```

        Status_Lampu_1 = "Lampu 1 OFF";
    }
    if (Lampu_2 == LOW) {
        Status_Lampu_2 = "Lampu 2 ON";
    } else {
        Status_Lampu_2 = "Lampu 2 OFF";
    }
    if (Lampu_3 == LOW) {
        Status_Lampu_3 = "Lampu 3 ON";
    } else {
        Status_Lampu_3 = "Lampu 3 OFF";
    }
    if (Lampu_4 == LOW) {
        Status_Lampu_4 = "Lampu 4 ON";
    } else {
        Status_Lampu_4 = "Lampu 4 OFF";
    }
    if (Lampu_5 == LOW) {
        Status_Lampu_5 = "Lampu 5 ON";
    } else {
        Status_Lampu_5 = "Lampu 5 OFF";
    }
    if (Lampu_6 == LOW) {
        Status_Lampu_6 = "Lampu 6 ON";
    } else {
        Status_Lampu_6 = "Lampu 6 OFF";
    }
    if (Lampu_7 == LOW) {
        Status_Lampu_7 = "Lampu 7 ON";
    } else {
        Status_Lampu_7 = "Lampu 7 OFF";
    }
    if (Lampu_8 == LOW) {
        Status_Lampu_8 = "Lampu 8 ON";
    } else {
        Status_Lampu_8 = "Lampu 8 OFF";
    }
    if (Lampu_9 == LOW) {
        Status_Lampu_9 = "Lampu 9 ON";
    } else {
        Status_Lampu_9 = "Lampu 9 OFF";
    }
}

Serial.println("\nUser mengirim perintah 'Cek Status'\n");
myBot.sendMessage(msg.sender.id, "Status saat ini : \n\n☹ " +
Status_Lampu_1 + "\n\n☹ " + Status_Lampu_2 + "\n\n☹ " + Status_Lampu_3 +

```

```
"\n\n☒ " + Status_Lampu_4 + "\n\n☒ " + Status_Lampu_5 + "\n\n☒ " +  
Status_Lampu_6 + "\n\n☒ " + Status_Lampu_7 + "\n\n☒ " + Status_Lampu_8 +  
"\n\n☒ " + Status_Lampu_9 + "\n\nGunakan tombol dibawah untuk mengontrol  
lampu ☒", Tbl);
```

```
Serial.println("\nStatus saat ini : \n\n☒ " + Status_Lampu_1 + "\n☒ " +  
Status_Lampu_2 + "\n☒ " + Status_Lampu_3 + "\n☒ " + Status_Lampu_4 +  
"\n☒ " + Status_Lampu_5 + "\n☒ " + Status_Lampu_6 + "\n☒ " +  
Status_Lampu_7 + "\n☒ " + Status_Lampu_8 + "\n☒ " + Status_Lampu_9 +  
"\n");
```

```
} else if (msg.messageType == CTBotMessageQuery) {  
if (msg.callbackQueryData.equals(LAMPU_1_ON)) {  
digitalWrite(Lampu1, LOW);  
Lampu_1 = LOW;  
myBot.sendMessage(msg.sender.id, "💡 Lampu 1 Telah Dinyalakan");  
Serial.println("\nUser menyalakan Lampu 1\n");  
} else if (msg.callbackQueryData.equals(LAMPU_1_OFF)) {  
digitalWrite(Lampu1, HIGH);  
Lampu_1 = HIGH;  
myBot.sendMessage(msg.sender.id, "🚫 Lampu 1 Telah Dimatikan");  
Serial.println("\nUser mematikan Lampu 1\n");  
} else if (msg.callbackQueryData.equals(LAMPU_2_ON)) {  
digitalWrite(Lampu2, LOW);  
Lampu_2 = LOW;  
myBot.sendMessage(msg.sender.id, "💡 Lampu 2 Telah Dinyalakan");  
Serial.println("\nUser menyalakan Lampu 2\n");  
} else if (msg.callbackQueryData.equals(LAMPU_2_OFF)) {  
digitalWrite(Lampu2, HIGH);  
Lampu_2 = HIGH;  
myBot.sendMessage(msg.sender.id, "🚫 Lampu 2 Telah Dimatikan");  
Serial.println("\nUser mematikan Lampu 2\n");  
} else if (msg.callbackQueryData.equals(LAMPU_3_ON)) {  
digitalWrite(Lampu3, LOW);  
Lampu_3 = LOW;  
myBot.sendMessage(msg.sender.id, "💡 Lampu 3 Telah Dinyalakan");  
Serial.println("\nUser menyalakan Lampu 3\n");  
} else if (msg.callbackQueryData.equals(LAMPU_3_OFF)) {  
digitalWrite(Lampu3, HIGH);  
Lampu_3 = HIGH;  
myBot.sendMessage(msg.sender.id, "🚫 Lampu 3 Telah Dimatikan");  
Serial.println("\nUser mematikan Lampu 3\n");  
} else if (msg.callbackQueryData.equals(LAMPU_4_ON)) {  
digitalWrite(Lampu4, LOW);  
Lampu_4 = LOW;  
myBot.sendMessage(msg.sender.id, "💡 Lampu 4 Telah Dinyalakan");
```

```

Serial.println("\nUser menyalakan Lampu 4\n");
} else if (msg.callbackQueryData.equals(LAMPU_4_OFF)) {
    digitalWrite(Lampu4, HIGH);
    Lampu_4 = HIGH;
    myBot.sendMessage(msg.sender.id, "🚫 Lampu 4 Telah Dimatikan");
    Serial.println("\nUser mematikan Lampu 4\n");
} else if (msg.callbackQueryData.equals(LAMPU_5_ON)) {
    digitalWrite(Lampu5, LOW);
    Lampu_5 = LOW;
    myBot.sendMessage(msg.sender.id, "💡 Lampu 5 Telah Dinyalakan");
    Serial.println("\nUser menyalakan Lampu 5\n");
} else if (msg.callbackQueryData.equals(LAMPU_5_OFF)) {
    digitalWrite(Lampu5, HIGH);
    Lampu_5 = HIGH;
    myBot.sendMessage(msg.sender.id, "🚫 Lampu 5 Telah Dimatikan");
    Serial.println("\nUser mematikan Lampu 5\n");
} else if (msg.callbackQueryData.equals(LAMPU_6_ON)) {
    digitalWrite(Lampu6, LOW);
    Lampu_6 = LOW;
    myBot.sendMessage(msg.sender.id, "💡 Lampu 6 Telah Dinyalakan");
    Serial.println("\nUser menyalakan Lampu 2\n");
} else if (msg.callbackQueryData.equals(LAMPU_6_OFF)) {
    digitalWrite(Lampu6, HIGH);
    Lampu_6 = HIGH;
    myBot.sendMessage(msg.sender.id, "🚫 Lampu 6 Telah Dimatikan");
    Serial.println("\nUser mematikan Lampu 6\n");
} else if (msg.callbackQueryData.equals(LAMPU_7_ON)) {
    digitalWrite(Lampu7, LOW);
    Lampu_7 = LOW;
    myBot.sendMessage(msg.sender.id, "💡 Lampu 7 Telah Dinyalakan");
    Serial.println("\nUser menyalakan Lampu 7\n");
} else if (msg.callbackQueryData.equals(LAMPU_7_OFF)) {
    digitalWrite(Lampu7, HIGH);
    Lampu_7 = HIGH;
    myBot.sendMessage(msg.sender.id, "🚫 Lampu 7 Telah Dimatikan");
    Serial.println("\nUser mematikan Lampu 7\n");
} else if (msg.callbackQueryData.equals(LAMPU_8_ON)) {
    digitalWrite(Lampu8, LOW);
    Lampu_8 = LOW;
    myBot.sendMessage(msg.sender.id, "💡 Lampu 8 Telah Dinyalakan");
    Serial.println("\nUser menyalakan Lampu 8\n");
} else if (msg.callbackQueryData.equals(LAMPU_8_OFF)) {
    digitalWrite(Lampu8, HIGH);
    Lampu_8 = HIGH;
    myBot.sendMessage(msg.sender.id, "🚫 Lampu 8 Telah Dimatikan");

```

```

        Serial.println("\nUser mematikan Lampu 8\n");
    } else if (msg.callbackQueryData.equals(LAMPU_9_ON)) {
        digitalWrite(Lampu9, LOW);
        Lampu_9 = LOW;
        myBot.sendMessage(msg.sender.id, "💡 Lampu 9 Telah Dinyalakan");
        Serial.println("\nUser menyalakan Lampu 9\n");
    } else if (msg.callbackQueryData.equals(LAMPU_9_OFF)) {
        digitalWrite(Lampu9, HIGH);
        Lampu_9 = HIGH;
        myBot.sendMessage(msg.sender.id, "🚫 Lampu 9 Telah Dimatikan");
        Serial.println("\nUser mematikan Lampu 9\n");
    } else if (msg.callbackQueryData.equals(SEMUA_LAMPU_ON)) {
        digitalWrite(Lampu1, LOW);
        digitalWrite(Lampu2, LOW);
        digitalWrite(Lampu3, LOW);
        digitalWrite(Lampu4, LOW);
        digitalWrite(Lampu5, LOW);
        digitalWrite(Lampu6, LOW);
        digitalWrite(Lampu7, LOW);
        digitalWrite(Lampu8, LOW);
        digitalWrite(Lampu9, LOW);

        Lampu_1 = LOW;
        Lampu_2 = LOW;
        Lampu_3 = LOW;
        Lampu_4 = LOW;
        Lampu_5 = LOW;
        Lampu_6 = LOW;
        Lampu_7 = LOW;
        Lampu_8 = LOW;
        Lampu_9 = LOW;

        myBot.sendMessage(msg.sender.id, "Semua 💡💡💡💡 Lampu Telah
Dinyalakan");
        Serial.println("\nUser telah menyalakan Semua Lampu\n");
    } else if (msg.callbackQueryData.equals(SEMUA_LAMPU_OFF)) {
        digitalWrite(Lampu1, HIGH);
        digitalWrite(Lampu2, HIGH);
        digitalWrite(Lampu3, HIGH);
        digitalWrite(Lampu4, HIGH);
        digitalWrite(Lampu5, HIGH);
        digitalWrite(Lampu6, HIGH);
        digitalWrite(Lampu7, HIGH);
        digitalWrite(Lampu8, HIGH);
        digitalWrite(Lampu9, HIGH);

```



```
Lampu_1 = HIGH;
Lampu_2 = HIGH;
Lampu_3 = HIGH;
Lampu_4 = HIGH;
Lampu_5 = HIGH;
Lampu_6 = HIGH;
Lampu_7 = HIGH;
Lampu_8 = HIGH;
Lampu_9 = HIGH;
```

```
myBot.sendMessage(msg.sender.id, "⊙ Semua Lampu Telah
Dimatikan");
Serial.println("\nUser telah mematikan Semua Lampu\n");
} else if (msg.callbackQueryData.equals(TIDAK)) {
myBot.sendMessage(msg.sender.id, "✘ Perintah Telah Dibatalkan!!!");
Serial.println("\nUser telah membatalkan perintah\n");
}
} else {
myBot.sendMessage(msg.sender.id, "🚫 ERROR:
Perintah tidak dikenal?!\n\nGunakan tombol dibawah untuk mengontrol lampu
👉", Tbl);
Serial.println("\nUser mengirim perintah yang tidak dikenal\n");
}
}
delay(500);
}
```