

**PENERAPAN *INTERNET OF THINGS* UNTUK SISTEM  
PENGENDALIAN LAMPU TAMAN BERBASIS NODEMCU  
ESP8266**

Tugas Akhir

Untuk memenuhi sebagian persyaratan  
Mencapai derajat Sarjana S-1 Jurusan Teknik Elektro



Oleh :

**MUHAMAD KIFLAN AFFANDI**

**F1B 018 042**

**JURUSAN TEKNIK ELEKTRO**

**FAKULTAS TEKNIK**

**UNIVERSITAS MATARAM**

**2023**

**Tugas Akhir**

**PENERAPAN *INTERNET OF THINGS* UNTUK SISTEM  
PENGENDALIAN LAMPU TAMAN BERBASIS NODEMCU  
ESP8266**

Telah diperiksa dan disetujui oleh Tim Pembimbing :

1. Pembimbing Utama



Budi Darmawan, S.T., M.Eng.  
NIP. 19850526 201504 1 002

Tanggal : 31 / 05 / 2023

2. Pembimbing Pendamping



Paniran, S.T., M.T.  
NIP. 19710723 199903 1 001

Tanggal : 31 / 05 / 2023

Mengetahui,

Ketua Jurusan Teknik Elektro

Fakultas Teknik

Universitas Mataram



A. Samsjiar Rachman, S.T., M.T.  
NIP. 197111241999031004

**Tugas Akhir**

**PENERAPAN *INTERNET OF THINGS* UNTUK SISTEM  
PENGENDALIAN LAMPU TAMAN BERBASIS NODEMCU  
ESP8266**


**Oleh:  
Muhamad Kiflan Affandi  
F1B 018 042**

Telah dipertahankan di depan Dewan Penguji  
Pada tanggal 24 Mei 2023  
dan dinyatakan telah memenuhi syarat mencapai derajat Sarjana S-1  
Jurusan Teknik Elektro

**Susunan Tim Penguji**

1. Penguji 1

Pada Tanggal : 31/05/2023

  
**I Made Ari Nrartha, S.T., M.T.**  
NIP: 19730404 199903 1 002

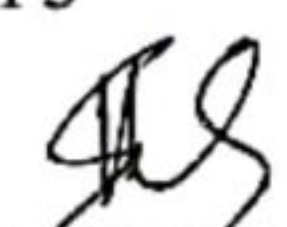
2. Penguji 2

Pada Tanggal : 30 Mei 2023

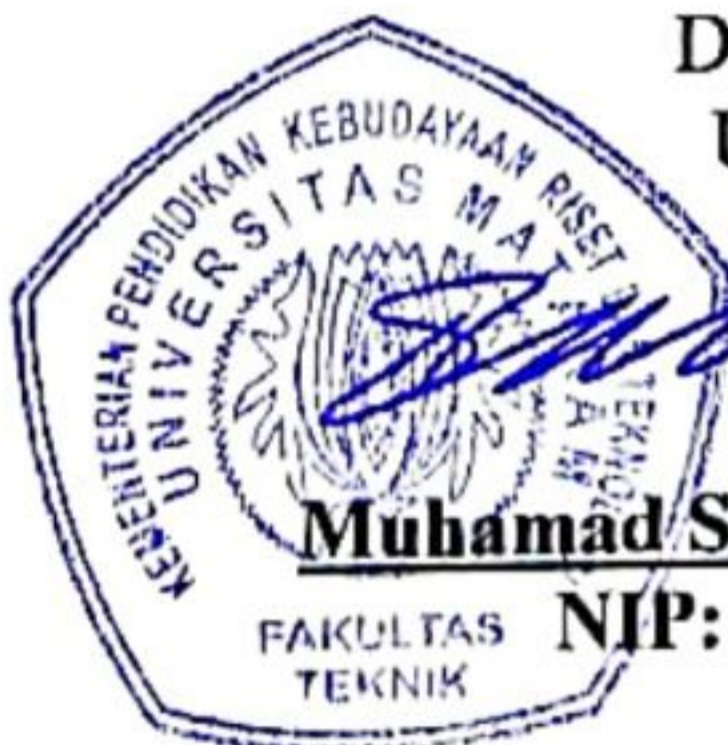
  
**Dwi Ratnasari, S.Kom., M.T.**  
NIP: 19850901 201903 2 010

3. Penguji 3

Pada Tanggal : 31/05/2023

  
**Syafarudin Ch, S.T., M.T.**  
NIP: 19690612 199702 1 001

Mataram,  
Dekan Fakultas Teknik  
Universitas Mataram



**Muhamad Syamsu Iqbal, S.T., M.T., Ph.D.**  
NIP: 19720222 199903 1 002

## **SURAT PERNYATAAN KEASLIAN TUGAS AKHIR**

Yang bertanda tangan di bawah ini:

Nama : Muhamad Kiflan Affandi  
NIM : F1B 018 042  
Program Studi : Teknik Elektro  
Perguruan Tinggi : Universitas Mataram

Dengan ini menyatakan bahwa tugas akhir yang telah saya buat dengan judul: “Penerapan *Internet Of Things* Untuk Sistem Pengendalian Lampu Taman Berbasis *NodeMCU ESP8266*” adalah asli (orsinil) atau tidak plagiat (menjiplak) dan belum pernah diterbitkan/dipublikasikan dimanapun dan dalam bentuk apapun.

Demikianlah surat pernyataan ini saya buat dengan sebenar-benarnya tanpa ada paksaan dari pihak manapun juga. Apabila dikemudian hari ternyata saya memberikan keterangan palsu dan atau ada pihak lain yang mengklaim bahwa tugas akhir yang telah saya buat adalah hasil karya milik seseorang atau badan tertentu, saya bersedia diproses baik secara pidana maupun perdata dan kelulusan saya dari Universitas Mataram dicabut/dibatalkan.

Dibuat di : Mataram  
Pada tanggal : 25 Mei 2023  
Yang menyatakan

Muhamad Kiflan Affandi

## **PRAKATA**

Puji syukur penulis panjatkan kehadiran Allah Subhanahu Wa Ta`ala atas segala berkat, rahmat, dan karunia-Nya, sehingga penulis dapat menyelesaikan mengajukan proposal Tugas Akhir ini. Dalam Tugas Akhir ini, penulis melakukan penelitian dengan judul “Penerapan *Internet Of Things* Untuk Sistem Pengendalian Lampu Taman Berbasis *NodeMCU ESP8266*”.

Proposal tugas akhir ini diajukan sebagai salah satu syarat dalam menyelesaikan studi dan untuk mendapatkan gelar Sarjana Teknik pada program studi S-1 Teknik Elektro, Fakultas Teknik, Universitas Mataram.

Penulis

25 Agustus 2022

## UCAPAN TERIMA KASIH

Tugas Akhir ini dapat diselesaikan berkat bimbingan dan dukungan ilmiah maupun materil dari berbagai pihak. Oleh karena itu pada kesempatan ini penulis menyampaikan ucapan terima kasih yang setulus-tulusnya kepada :

1. Kedua orang tua penulis yaitu bapak Anang S.Pd dan ibu Asmah S.pd yang selalu mendukung, motivasi, memberi nasihat, doa dan kasih sayang yang sebesar besarnya dan kakak penulis yaitu Kustati Nurul Hidayah yang selalu ada ketika di butuhkan untuk menyelesaikan tugas akhir ini.
2. Bapak Muhammad Syamsul Iqbal, ST., MT., Ph.D., selaku Dekan Fakultas Teknik Universitas Mataram.
3. Bapak A Sjamsjiar Rachman, ST., MT., selaku Ketua Jurusan Teknik Elektro Fakultas Teknik Universitas Mataram.
4. Bapak Budi Darmawan, ST., MT., selaku dosen pembimbing pertama yang telah memberikan bimbingan dan arahan kepada penulis selama menyusun Tugas Akhir ini, sehingga dapat terselesaikan dengan baik.
5. Bapak Paniran, ST., MT., selaku dosen pembimbing pendamping yang telah memberikan bimbingan dan arahan kepada penulis selama menyusun Tugas Akhir ini, sehingga dapat terselesaikan dengan baik.
6. Keluarga besar mansur duru dan kakek nenek yang selalu memberi semangat.
7. Team BTN aldy, roby, novian, thoriq, ajik, roby hu, angga, sindi, yang membantu menyelesaikan.
8. Team suci lahir batin adnin, diwan, rizki, irfan, tomy yang selalu memberikan motivasi semangat dalam menyelesaikan tugas akhir ini.
9. Teman- teman seperjuangan Angkatan 2018 yang telah banyak membantu dalam proses penyelesaian Tugas Akhir ini.
10. To my beloved brothers Muhamad Kiflan Afandi, Rafif Assidiqie and my beloved sister Naufal Nafisa and all my family who always pray for me Thank you so much.

## DAFTAR ISI

<b>COVER</b> .....	<b>i</b>
<b>LEMBAR PENGESAHAN TUGAS AKHIR</b> .....	<b>ii</b>
<b>LEMBAR PENGESAHAN PENGUJI</b> .....	<b>iii</b>
<b>SURAT PERNYATAAN KEASLIAN TUGAS AKHIR</b> .....	<b>iv</b>
<b>PRAKATA</b> .....	<b>v</b>
<b>UCAPAN TERIMA KASIH</b> .....	<b>vi</b>
<b>DAFTAR ISI</b> .....	<b>vii</b>
<b>DAFTAR GAMBAR</b> .....	<b>ix</b>
<b>DAFTAR TABEL</b> .....	<b>x</b>
<b>ABSTRAK</b> .....	<b>xi</b>
<b>ABSTRACT</b> .....	<b>xii</b>
<b>BAB I</b> .....	<b>1</b>
<b>1.1 Latar Belakang</b> .....	<b>1</b>
<b>1.2 Rumusan Masalah</b> .....	<b>2</b>
<b>1.3 Batasan Masalah</b> .....	<b>2</b>
<b>1.4 Tujuan Penelitian</b> .....	<b>2</b>
<b>1.5 Manfaat Penelitian</b> .....	<b>2</b>
<b>1.6 Sistematika Penulisan</b> .....	<b>3</b>
<b>BAB II</b> .....	<b>4</b>
<b>2.1 Tinjauan Pustaka</b> .....	<b>4</b>
<b>2.2 Dasar Teori</b> .....	<b>5</b>
2.2.1 Deskripsi Umum Lampu Taman.....	5
2.2.2 <i>Internet Of things</i> (IoT) .....	5
2.2.3 Android.....	5
2.2.4 Aplikasi Telegram.....	6
2.2.5 Modul <i>NodeMCU</i> ESP8266 .....	6
2.2.6 Arduino Uno .....	7
2.2.7 Sensor Arus ACS712 30A.....	8
2.2.8 LCD.....	9

2.2.9	<i>Relay</i> .....	10
2.2.10	RTC.....	10
2.2.11	Lampu LED.....	11
2.2.12	Transistor Daya.....	12
2.2.13	Persamaan Regresi Linear Sederhana.....	14
<b>BAB III.....</b>		<b>16</b>
<b>3.1 Diagram Tahap Pelaksanaan.....</b>		<b>16</b>
<b>3.2 Analisa Kebutuhan Sistem.....</b>		<b>17</b>
3.2.1	Perangkat Keras ( <i>hardware</i> ).....	17
3.2.2	Perangkat Lunak ( <i>software</i> ).....	18
<b>3.3 Prosedur Perancangan.....</b>		<b>18</b>
<b>3.4 Perancangan Sistem.....</b>		<b>19</b>
3.4.1	Perancangan Perangkat Keras ( <i>hardware</i> ).....	20
3.4.2	Perancangan Perangkat lunak ( <i>software</i> ).....	25
<b>3.5 Pengujian Alat.....</b>		<b>29</b>
<b>BAB IV.....</b>		<b>30</b>
<b>4.1 Hasil Perancangan.....</b>		<b>30</b>
4.1.1	Perangkat Keras ( <i>Hardware</i> ).....	30
4.1.2	Perangkat Lunak ( <i>Software</i> ).....	31
<b>4.2 Pengujian Sistem.....</b>		<b>34</b>
4.2.1	Pengujian Sensor ACS712 dengan Multimeter Digital Sebagai Acuan ...	34
4.2.2	Pengujian Aplikasi Telegram.....	40
4.2.3	Pengujian Sistem Keseluruhan.....	42
<b>BAB V.....</b>		<b>45</b>
<b>5.1 Kesimpulan.....</b>		<b>45</b>
<b>5.2 Saran.....</b>		<b>45</b>
<b>DAFTAR PUSTAKA.....</b>		<b>46</b>
<b>LAMPIRAN.....</b>		<b>48</b>



## DAFTAR GAMBAR

Gambar 2.1 <i>NodeMCU</i> ESP8266.....	7
Gambar 2.2 <i>Board</i> Arduino Uno.....	8
Gambar 2.3 Sensor Arus ACS712 30A.....	9
Gambar 2.4 LCD 16x2 .....	10
Gambar 2.5 Modul <i>Relay</i> .....	10
Gambar 2.6 <i>Real Time Clock</i> .....	11
Gambar 2.7 Lampu LED .....	12
Gambar 2.8 Struktur, Simbol dan Arah Arus N-P-N Transistor.....	13
Gambar 2.9 Rangkaian Transistor Mengontrol Beban.....	13
Gambar 2.10 Ilustrasi Garis Regresi Linear .....	15
Gambar 3.1 Diagram Tahap Pelaksanaan.....	16
Gambar 3.2 <i>Flowchart</i> Sistem Pengendalian Lampu Taman .....	19
Gambar 3.3 Rangkaian <i>Relay</i> .....	20
Gambar 3.4 Komponen <i>Relay</i> .....	21
Gambar 3.5 Rangkaian Perangkat Keras.....	24
Gambar 3.6 Diagram Alir Perancangan <i>Software</i> .....	26
Gambar 3.7 Diagram Alir Program <i>NodeMCU</i> ESP8266.....	27
Gambar 3.8 Diagram Alir Program Pada Arduino Uno .....	28
Gambar 4.1 Hasil perancangan <i>prototype</i> lampu taman. ....	30
Gambar 4.2 Tampilan <i>bot father</i> (a).....	31
Gambar 4.3 Tampilan <i>bot father</i> (b). ....	32
Gambar 4.4 Tampilan <i>bot</i> yang telah dibuat user. ....	33
Gambar 4.5 Pengukuran arus lampu menggunakan sensor ACS712.....	34
Gambar 4.6 Pengukuran arus lampu menggunakan Multimeter Digital DT9205A. ....	34
Gambar 4.7 Grafik Kalibrasi Linier Sensor ACS712 .....	36
Gambar 4.8 Tampilan laman bot Telegram user.....	40
Gambar 4.9 Pengujian keseluruhan dengan empat titik lampu normal.....	42
Gambar 4.10 Pengujian keseluruhan dengan salah satu titik lampu rusak.....	42

## DAFTAR TABEL

Tabel 4.1 Pengukuran arus lampu menggunakan sensor ACS712 dan pengukuran arus lampu menggunakan Multimeter Digital DT9205A .....	35
Tabel 4.2 Hasil Perhitungan Regresi Linier .....	37
Tabel 4.3 Nilai ACS712 dimasukkan ke dalam Regresi Linier .....	38
Tabel 4.4 Perintah Kendali Lampu Menggunakan Telegram.....	41
Tabel 4.5 Perintah Kendali Lampu dengan Aplikasi Telegram.....	43

## ABSTRAK

Studi ini melibatkan penggunaan *NodeMCU* ESP8266, modul mikrokontroler berbasis Wi-Fi yang terjangkau dan mudah digunakan. *NodeMCU* ESP8266 digunakan sebagai otak utama dalam sistem IoT, bertanggung jawab untuk mengumpulkan data dari sensor dan mengirimkannya melalui koneksi *Wi-Fi* ke *server*. Ketika menjelang malam, *NodeMCU* ESP8266 akan mengirimkan sinyal ke *relay* untuk menyalakan lampu. Sebaliknya, ketika menjelang pagi, *NodeMCU* ESP8266 akan mengirimkan sinyal ke *relay* untuk mematikan lampu. Selain itu, pengguna dapat mengendalikan lampu taman melalui telegram yang terhubung dengan sistem *IoT*. Telegram memungkinkan pengguna untuk secara manual mengontrol pencahayaan lampu taman sesuai kebutuhan mereka. Pengguna juga dapat mengatur secara manual jadwal untuk menghidupkan atau mematikan lampu. Dalam penelitian ini, implementasi sistem pengendalian lampu taman berbasis *IoT* menggunakan *NodeMCU* ESP8266 telah berhasil dilakukan. Sistem ini menunjukkan kehandalan dan fleksibilitas dalam mengendalikan lampu taman secara otomatis dan manual. Penerapan *IoT* dalam pengendalian lampu taman dapat memberikan manfaat seperti efisiensi energi, kenyamanan, dan kemudahan penggunaan.

**Kata Kunci:** *Internet of Things*, *NodeMCU* ESP8266, Lampu Taman, Telegram.

## **ABSTRACT**

*This study involves the use of NodeMCU ESP8266, an affordable and user-friendly Wi-Fi-based microcontroller module. NodeMCU ESP8266 is used as the main brain in the IoT system, responsible for collecting data from sensors and sending it via Wi-Fi connection to the server. As evening approaches, NodeMCU ESP8266 will send a signal to the relay to turn on the lights. Conversely, as morning approaches, NodeMCU ESP8266 will send a signal to the relay to turn off the lights. Additionally, users can control the garden lights through Telegram, which is connected to the IoT system. Telegram allows users to manually control the garden light illumination according to their needs. Users can also manually set schedules to turn the lights on or off. In this study, the implementation of an IoT-based garden light control system using NodeMCU ESP8266 has been successfully carried out. This system demonstrates reliability and flexibility in automatically and manually controlling the garden lights. The implementation of IoT in garden light control can provide benefits such as energy efficiency, convenience, and ease of use.*

**Keywords:** *Internet of Things, NodeMCU ESP8266, Garden Lights, Telegram.*

# BAB I PENDALUHUAN

## 1.1 Latar Belakang

Perkembangan teknologi saat ini berkembang dengan sangat pesat. Aktivitas kita sehari-hari juga didampingi oleh banyak sekali implementasi teknologi. Mulai dari memesan makanan online, membeli barang kebutuhan online, ojek online, komunikasi jarak jauh, dll. Sehingga masyarakat zaman sekarang sudah tidak asing lagi dengan teknologi bahkan tidak sedikit orang yang tertarik dibidang teknologi khususnya pengendalian jarak jauh, hal ini disebabkan karena kemudahan dan penghematan listrik yang bisa kita dapatkan. Salah satu kebiasaan pemborosan listrik yang masih sering dilakukan sebagian besar masyarakat adalah menyalakan lampu taman atau lampu teras rumah ketika sedang berpergian jauh dalam jangka waktu yang lama. Namun karena tidak ada siapapun yang bisa mematikan dan menyalakan sakelar lampu tersebut setiap hari, alhasil lampu taman akan tetap dibiarkan menyala sepanjang hari dalam jangka waktu lama. Padahal sebenarnya yang masyarakat inginkan adalah agar lampu taman tetap menyala pada malam hari dan mati pada siang hari ketika terang. Dengan adanya pengendalian jarak jauh menggunakan *smartphone* tentu saja masalah tersebut dapat diatasi.

Lampu taman adalah salah satu fasilitas penerangan yang ada di setiap taman. Ketika lampu taman padam masih banyak yang ditangani dengan cara manual, banyak pengguna lampu taman membiarkan lampu menyala terus menerus sepanjang malam bahkan sepanjang hari dengan tingkat kecerahan maksimal, hal tersebut merupakan pemborosan dalam penggunaan energi listrik. Pemantauan lampu dilakukan dengan mendatangi tempat dimana lampu berada, jika kontrol lampu seperti ini maka sangat banyak kekurangan salah satunya petugas membutuhkan waktu untuk mengetahui kondisi lampu taman. Oleh karena itu diperlukan penyesuaian tingkat kecerahan lampu pada jam – jam tertentu, dan diperlukan adanya sistem kontrol dan *monitoring* lampu penerangan taman yang dapat dikendali oleh sistem berbasis *IoT* dan *Android* untuk memudahkan petugas yang mengontrol lampu penerangan taman dari jarak jauh (Rijalul, dkk. 2020).

Berdasarkan permasalahan dan paparan tersebut maka dalam penelitian ini akan dibangun sistem pengendalian lampu taman berbasis *NodeMCU* menggunakan aplikasi telegram untuk memudahkan pengguna yang mengontrol lampu taman dari jarak jauh, pengontrolan lampu taman oleh pengguna juga dapat di-*setting* berdasarkan waktu.

## **1.2 Rumusan Masalah**

Berdasarkan uraian latar belakang, maka dapat dirumuskan permasalahan yaitu lampu taman yang sering menyala sepanjang malam sehingga memakan banyak daya listrik. Dengan adanya tugas akhir ini diharapkan bisa membantu mengurangi penggunaan listrik dan dapat mengetahui titik lampu yang rusak disuatu tempat menggunakan *platform IoT* tanpa harus mendatangi tempat tersebut dengan kendali menggunakan aplikasi Telegram.

## **1.3 Batasan Masalah**

Dari permasalahan yang telah diuraikan di atas terdapat batasan-batasan masalah dalam pembangunan aplikasi ini yaitu:

1. Perancangan pengendalian lampu taman menggunakan *NodeMCU* ESP8266.
2. Pengendalian lampu taman digunakan pada taman rumah.
3. Pengendalian lampu taman menggunakan aplikasi Telegram.
4. Sensor arus ACS712 digunakan untuk menunjukkan ada atau tidaknya arus yang mengalir pada lampu.

## **1.4 Tujuan Penelitian**

Tujuan yang diharapkan dari penelitian ini adalah sebagai berikut:

1. Membangun sebuah *prototype* sistem dan alat yang berfungsi untuk pengendali lampu taman dengan kendali *smartphone*.
2. Mengimplementasikan media sosial telegram sebagai pengendali lampu taman.
3. Dapat Memanajemen dalam penggunaan lampu taman sehingga mengurangi penggunaan listrik.

## **1.5 Manfaat Penelitian**

1. Dapat menambah pengetahuan tentang *Internet of Thing* dalam menyelesaikan masalah.
2. Menghasilkan alat dan sistem kendali yang dapat memudahkan penggunaanya.

3. Memberikan gambaran sebagai salah satu manfaat penggunaan media sosial telegram.
4. Memberikan gambaran dan pedoman sebagai contoh penerapan pengendalian dan pemantauan jarak jauh untuk upaya penghematan listrik.

### **1.6 Sistematika Penulisan**

Sistematika penulisan dari penelitian ini disajikan dalam beberapa bab antara lain sebagai berikut.

#### **1. Bab I Pendahuluan**

Berisi tentang penjelasan mengenai latar belakang, rumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian serta sistematika penulisan yang digunakan untuk menyusun hasil Tugas Akhir.

#### **2. Bab II Tinjauan Pustaka dan Landasan Teori**

Berisi mengenai tinjauan pustakan dan teori-teori dasar yang sebagai penunjang pembahasan permasalahan yang diperoleh dari literature yang ada di buku, jurnal, maupun *e-book*.

#### **3. Bab III Metodologi Penelitian**

Berisi tentang rencana pelaksanaan, alat dan bahan, serta sistem rancangan yang dilakukan dalam penelitian Tugas Akhir.

#### **4. Bab IV Hasil dan Pembahasan**

Berisi hasil dan pembahasan berisi mengenai temuan ilmiah yang diperoleh sebagai hasil penelitian dan memuat penjelasan logis mengenai penelitian yang dilakukan serta alasan yang rasional yang dapat menjelaskan dari hasil penelitian yang didapatkan.

#### **5. Bab V kesimpulan dan Saran**

Berisi tentang laporan tugas akhir yang merupakan kesimpulan dari proses penelitian dan saran dari penulis untuk penelitian selanjutnya dengan harapan dapat bermanfaat pada topik terkait.

## BAB II TINJAUAN PUSTAKA DAN DASAR TEORI

### 2.1 Tinjauan Pustaka

Imam, dkk. (2018) Telah melakukan penelitian membangun sistem untuk mengontrol intensitas lampu yang terdapat pada ruangan dengan menggunakan *mobile* dengan menggunakan pendekatan *User-Centered Design (UDC)*. Sistem yang dibangun ini memiliki fungsional yang tidak terbatas dalam menyalakan dan mematikan lampu, tetapi dapat mengatur tingkat intensitas cahaya dari beberapa lampu yang terpasang dalam ruangan dengan menggunakan menu *dimmer light* pada *android*, *dimmer light* memiliki pilihan dari angka 0-4 dimana semakin tinggi angka yang dipilih maka nyala lampu akan semakin terang.

Siregar dan Simon (2011) Telah melakukan penelitian ini yang dilatarbelakangi oleh permasalahan yang sering muncul ketika sudah tidak ada orang di dalam ruangan tersebut, sering kali lampu dan AC tidak dimatikan, sehingga berakibat pemborosan sumber daya listrik. Oleh karena itu, untuk menangani penggunaan setiap alat tersebut, penggunaan *embedded ethernet* sebagai sistem monitoring cahaya, suhu, dan gerak didalam ruangan berbasis *ethernet* merupakan salah satu solusi dalam efisien listrik. Penelitian ini akan dibangun sistem monitoring berbasis *ethernet* dengan menggunakan mikrokontroler AVR tipe ATMEGA8535 sebagai unit kontrol dan sebuah *ethernetcontroller* dengan IC ENC28J60 sebagai unit kontrol jaringan yang menangani komunikasi protokol TCP/IP. Penelitian ini bertujuan untuk dapat mengendalikan lampu dan AC agar dapat mengurangi pemborosan sumber daya listrik.

Budi Novianto (2016) Telah melakukan penelitian tersebut dilatar belakangi oleh bagaimana mengendalikan lampu secara jarak jauh dan mengecek kondisi lampu. Penelitian ini bertujuan untuk sebagai umpan balik peralatan ini menggunakan LDR sebagai sensor, GSM *shield* 900 sebagai penghubung informasi ke *handphone* dan Arduino uno sebagai pusat pengendalian dan pengolahan data yang nantinya akan memberikan perintah kepada *relay* dan akan di teruskan untuk menyalakan dan mematikan lampu.

Berdasarkan hasil pengujian yang telah dilakukan baik terhadap perangkat keras maupun terhadap perangkat lunak yang telah dibuat serta melihat tujuan dari penelitian maka dapat disimpulkan sebagai bahwa peralatan ini dapat membantu pemakai untuk mengendalikan dan memonitoring kondisi lampu taman dari jarak jauh menggunakan



*handphone* yang dimiliki. Adanya notifikasi balasan dari perintah yang dikeluarkan menyatakan lampu sudah dikendalikan. Ini membuktikan bahwa impan balik (*feedback*) bekerja dengan sempurna.

## **2.2 Dasar Teori**

### **2.2.1 Deskripsi Umum Lampu Taman**

Lampu taman adalah salah satu fasilitas penerangan yang ada di setiap taman. Ketika lampu taman padam masih banyak yang ditangani dengan cara manual, banyak pengguna lampu taman membiarkan lampu menyala terus menerus sepanjang malam bahkan sepanjang hari dengan tingkat kecerahan maksimal, hal tersebut merupakan pemborosan dalam penggunaan energi listrik. Dengan adanya kendali jarak jauh dan otomatis maka lampu taman akan menyala pada sore hari dan padam pada pagi hari sehingga tidak terjadi pemborosan energi listrik (Prasetya & Aulia, 2020).

### **2.2.2 Internet Of things (IoT)**

IoT merupakan suatu konsep yang bertujuan untuk memperluas manfaat dari konektivitas internet yang tersambung secara terus menerus. IoT tersendiri merupakan skenario dari suatu objek yang dapat melakukan suatu pengiriman data / informasi melalui jaringan tanpa campur tangan manusia. Teknologi IoT telah berkembang dari konvergensi *microelectromechanical systems* (MEMS), dan internet pada jaringan nirkabel. Sedangkan "A Thing" dapat didefinisikan sebagai subjek seperti orang dengan *implant* jantung, hewan peternakan dengan transponder chip dan lain-lain (Azzahra, 2018).

### **2.2.3 Android**

Android merupakan sebuah *software open-source* untuk sebagai perangkat mobile. Android adalah sistem operasi untuk perangkat *mobile* yang berbasis *Linux* untuk telepon seluler seperti telepon pintar dan komputer tablet. Telah banyak produsen *smartphone* yang menanamkan android sebagai sistem operasi perangkat mereka. *Android* merupakan tumpukan *software* yang terdiri dari tiga bagian, yaitu *operating system*, *middleware* dan *key application*. Untuk mengembangkan android dibutuhkan open handset alliance, konsorsium dari 34 perusahaan peranti keras, peranti lunak dan telekomunikasi, termasuk *google*, *HTC*, *intel*, *Motorola*, *Qualcoom*, *T mobile*, dan *Nvidia* (Margaretha, 2018).

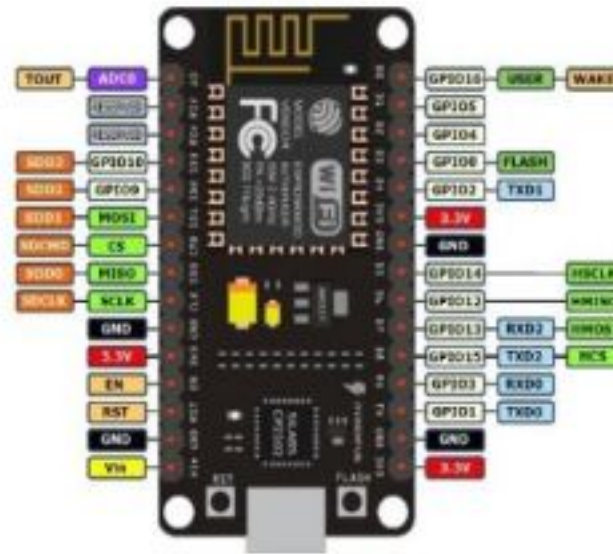
#### 2.2.4 Aplikasi Telegram

Telegram adalah sebuah aplikasi layanan pengirim pesan instan *multiplatform* berbasis awan yang bersifat gratis. Klien Telegram tersedia untuk perangkat telepon seluler (Android, iOS, Windows Phone, Ubuntu Touch) dan sistem perangkat komputer (Windows, OS X, Linux). Para pengguna dapat mengirim pesan dan bertukar foto, video, stiker, audio, dan semua tipe file atau berkas. Telegram juga menyediakan pengiriman pesan ujung ke ujung terenkripsi opsional. Pada penelitian ini, peneliti menggunakan aplikasi Telegram sebagai *switch* untuk menyalakan dan mematikan lampu (Irsyam, 2019).

#### 2.2.5 Modul *NodeMCU* ESP8266

*NodeMCU* adalah sebuah papan elektronik yang berbasis chip ESP8266 dengan kemampuan mengaplikasikan fungsi microcontroller dan juga jaringan internet (WiFi). Ditemukan jumlah pin I/O bahkan mampu mengembangkan membentuk sebuah aplikasi pemantauan ataupun pengendalian pada pekerjaan IOT. *NodeMCU* ESP8266 bisa di program dengan compiler-nya Arduino, memanfaatkan Arduino IDE. Gambaran rangka dari *NodeMCU* ESP8266, terdapat port USB (mini USB) akhirnya dapat mempermudah dalam mengatur programnya. *NodeMCU* ESP8266 merupakan modul turunan pengembangan dari modul *platform* IoT (*Internet of Things*) keluarga ESP8266 tipe ESP-12. Secara penggunaan alat ini sangat mirip dengan alat arduino, namun yang membedakannya yaitu diutamakan untuk “*Connected to Internet*” (Amin, M dkk. 2020).

ESP8266 adalah sebuah komponen *chip* terintegrasi yang didesain untuk keperluan dunia masa kini yang serba tersambung. *Chip* ini menawarkan solusi *networking Wi-Fi* yang lengkap dan menyatu, yang dapat digunakan sebagai penyedia aplikasi atau untuk memisahkan semua fungsi *networking Wi-Fi* ke pemroses aplikasi lainnya. ESP8266 memiliki kemampuan *on-board* *prosesing* dan *storage* yang memungkinkan chip tersebut untuk diintegrasikan dengan sensor-sensor atau dengan aplikasi alat tertentu melalui pin *input output* hanya dengan pemrograman singkat (Mehta, 2015).



Gambar 2.1 *NodeMCU* ESP8266.  
(Dewi, 2019)

Berikut merupakan spesifikasi dari *NodeMCU* ESP8266 adalah :

- a. Memiliki tegangan 3,3 volt.
- b. Memiliki CPU Xtensa single core L106 – 60 Mhz.
- c. Memiliki GPIO pin (ADC/DAC) sebesar 17 (1/-).
- d. Memiliki *Flash Memory* 16MB.

### 2.2.6 Arduino Uno

Arduino Uno adalah sebuah *board* yang menggunakan mikrokontroler Atmega328. Arduino Uno memiliki 14 pin digital (6 pin dapat digunakan sebagai *output* PWM), 6 *input* analog, sebuah 16 MHz isolator kristal, sebuah koneksi USB, sebuah konektor sumber tegangan, sebuah *header* ICSP, dan sebuah tombol *reset*. Arduino Uno memuat segala hal yang dibutuhkan untuk mendukung sebuah mikrokontroler. Hanya dengan menghubungkannya ke sebuah komputer melalui USB atau memberikan tegangan DC dari baterai atau adaptor AC ke DC sudah dapat membuatnya bekerja. Arduino Uno menggunakan ATmega16U2 yang diprogram sebagai *USB to serial converter* untuk komunikasi serial ke komputer melalui *port* USB (Basith, 2017).



Gambar 2.2 *Board* Arduino Uno  
(Gajah, 2018)

Spesifikasi data teknis yang terdapat pada *board* Arduino UNO R3 adalah sebagai berikut :

- a. Mikrokontroler : Atmega328
- b. Tegangan Operasi : 5V
- c. Tegangan *Input (recommended)* : 7 – 12 V
- d. Tegangan *Input (limit)* : 6 – 20 V
- e. Pin digital I/O : 14 (6 di antaranya pin PWM)
- f. Pin Analog *input* : 6 *input* pin 21
- g. Arus DC per pin I/O : 40 mA
- h. Arus DC untuk pin 3,3 V : 150 mA
- i. *Flash Memory* : 32 kb dengan 0,5 kb digunakan sebagai *bootloader*
- j. SRAM : 2 kb
- k. EEPROM : 1 kb
- l. *Clock Speed* : 16 MHz

### 2.2.7 Sensor Arus ACS712 30A

ACS712 merupakan suatu IC terpaket yang mana berguna sebagai sensor arus menggantikan transformator arus yang relatif besar dalam hal ukuran. Pada prinsipnya ACS712 sama dengan sensor *efek hall* lainnya yaitu dengan memanfaatkan medan magnetik disekitar arus kemudian dikonversi menjadi tegangan yang linier dengan perubahan arus. Nilai variabel dari sensor ini merupakan input untuk *mikrokontroller* yang kemudian diolah. Keluaran dari sensor ini masih berupa sinyal tegangan AC, agar dapat diolah *mikrokontroller* maka sinyal tegangan AC ini disearahkan oleh rangkaian penyearah (Dwi Suryawan, 2012).



Gambar 2.3 Sensor Arus ACS712 30A.

Fitur yang di miliki ACS712 30A sebagai berikut:

- a. *Rise time* output = 5  $\mu$ s.
- b. *Bandwidth* sampai dengan 80 kHz.
- c. Total kesalahan output 1,5% pada suhu kerja  $T_A = 25^\circ\text{C}$ .
- d. Tahanan konduktor internal 1,2 m $\Omega$ .
- e. Tegangan isolasi minimum 2,1 kVRMS antara pin 1-4 dan pin 5-8.
- f. Sensitivitas output 66 mV/A.
- g. Mampu mengukur arus AC atau DC hingga 30A.
- h. Tegangan output proporsional terhadap input arus AC atau DC.
- i. Tegangan kerja 4.5–5.5 VDC.

### 2.2.8 LCD

LCD (*Liquid Crystal Display*) adalah salah satu komponen elektronika yang fungsinya menampilkan suatu data, baik berupa karakter, huruf, atau grafik. Keunggulan menggunakan LCD adalah konsumsi daya yang relatif kecil dan menarik arus yang kecil (beberapa mikro ampere), sehingga alat atau sistem menjadi *portable* karena dapat menggunakan catu daya yang kecil. Keunggulan lainnya adalah ukuran LCD yang pas yakni tidak terlalu kecil dan tidak terlalu besar, kemudian tampilan yang diperlihatkan dari LCD dapat dibaca dengan mudah dan jelas.

Pada penelitian ini jenis LCD yang digunakan adalah LCD 16x2 yang sudah terpasang I2C (*Inter Integrated Circuit*) sebagai saluran untuk mengirim dan menerima data dari Arduino Uno ke LCD. Tanjung (2015), menyatakan spesifikasi pada LCD 16x2 adalah sebagai berikut:

- a. Terdiri dari 16 kolom dan 2 baris;
- b. Mempunyai 192 karakter yang tersimpan;
- c. Tegangan kerja 5V;
- d. Memiliki ukuran yang praktis.



Gambar 2.4 LCD 16x2

(Tanjung, 2015)

### 2.2.9 Relay

*Relay* merupakan komponen yang digunakan sebagai saklar penghubung atau pemutus arus beban yang cukup besar, dikontrol oleh sinyal listrik dengan arus yang kecil. *Relay* ini dapat digunakan sebagai switch untuk menjalankan berbagai peralatan elektronik. Kendali ON/OFF switch (*relay*), sepenuhnya ditentukan oleh nilai output sensor yang telah diproses pada mikrokontroler yang akan menghasilkan perintah kepada *relay* untuk melakukan fungsinya (Margaretha R. 2018).



Gambar 2.5 Modul *Relay*.

(Margaretha, 2018)

### 2.2.10 RTC

*Real Time Clock* (RTC) adalah jam elektronik berupa chip yang dapat menghitung waktu (mulai detik hingga tahun) dengan akurat dan menjaga atau menyimpan data waktu tersebut secara *real time*. Karena jam tersebut bekerja *real time*, maka setelah proses hitung waktu dilakukan output datanya langsung disimpan atau dikirim ke device lain melalui sistem antarmuka (Atmel, 2008).

Chip RTC sering dijumpai pada *motherboard* PC (biasanya terletak dekat *chip* BIOS). Semua komputer menggunakan RTC karena berfungsi menyimpan informasi jam terkini dari komputer yang bersangkutan. RTC dilengkapi dengan baterai sebagai pensuplai daya pada *chip*, sehingga jam akan tetap *up-to-date* walaupun komputer dimatikan. RTC dinilai cukup akurat sebagai perwaktu (*timer*) karena menggunakan osilator kristal. Banyak contoh chip RTC yang ada dipasaran seperti DS12C887, DS1307, DS1302, DS3234 (Febbry, 2018).



Gambar 2.6 *Real Time Clock*  
(Febbry, 2018)

### 2.2.11 Lampu LED

Lampu LED merupakan lampu terbaru yang merupakan sumber cahaya yang efisien energinya . Ketika lampu LED memancarkan cahaya nampak pada gelombang spectrum yang sempit , mereka dapat menghasilkan cahaya putih. Lampu LED memiliki usia pakai dan efisiensi listrik beberapa kali lipat lebih baik dari pada lampu pijar dan jauh lebih efisien dari lampu neon. Kelebihan dari lampu LED ini adalah lampu LED akan menghasilkan terang sesungguhnya tanpa perlu waktu pemanasan (Syamsuri, 2015).



Gambar 2.7 Lampu LED

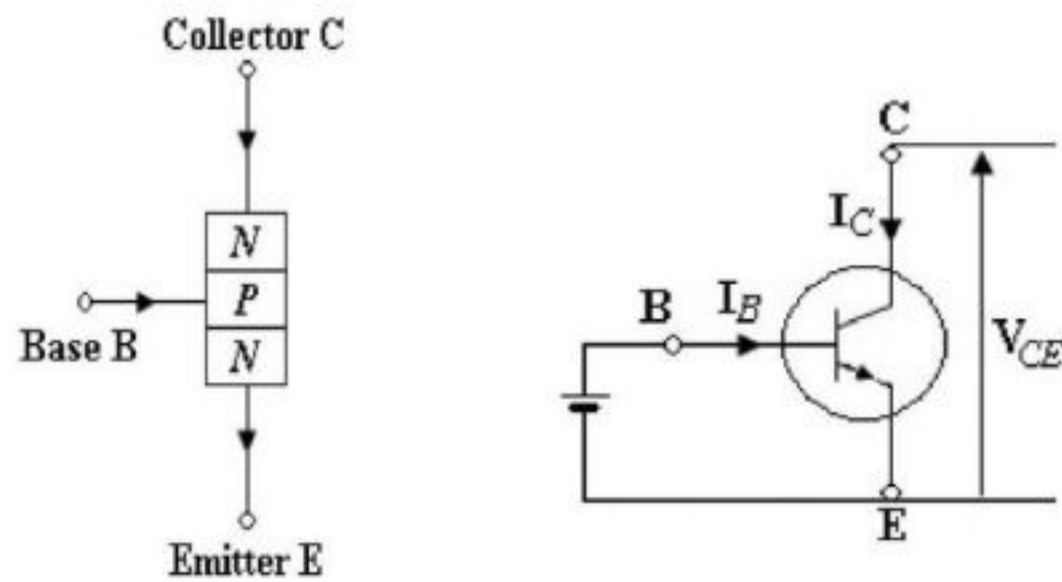
Berikut merupakan spesifikasi dari lampu LED adalah:

- a. Memiliki ukuran 10 watt.
- b. Dapat beroperasi pada tegangan 170-240 V.
- c. Tahan lama lampu sampai 10.000 jam.

#### **2.2.12 Transistor Daya**

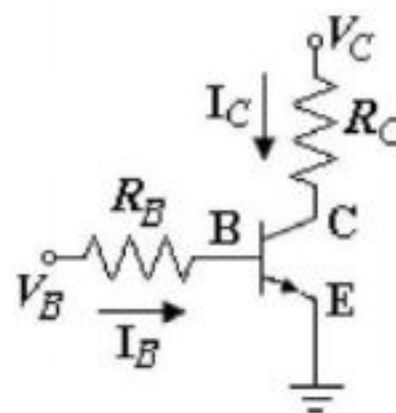
Transistor daya adalah perangkat yang terdiri dari tiga lapis N-P-N atau P-N-P. Prinsip kerjanya arus kolektor  $I_c$  yang merupakan fungsi dan arus basis  $I_b$ . perubahan pada arus basis akan mengakibatkan perubahan yang telah dikuatkan pada arus kolektor pada tegangan kolektor-emitor yang dikenakan padanya. Perbandingan kedua arus tersebut antara 15 sampai 100. Tegangan minimum listrik akan dicapai bila tegangan yang ditambahkan mencapai suatu batas. Tegangan balik kolektor-emitor yang dapat menyebabkan tegangan listrik pada gandingan basis-emitor pada level rendah misalnya 10 volt, disini transistor tidak dapat bekerja pada mode reverse. Dapat ditambahkan dioda secara seri untuk memperbesar kemampuan menahan tegangan balik (*reverse*) (Kustija, 2014).





Gambar 2.8 Struktur, Simbol dan Arah Arus N-P-N Transistor  
(Kustija, 2014)

Dengan memanfaatkan karakteristik Transistor emitor bersama, pada kondisi saturasi (jenuh) dan keadaan *cut-off* (mati) maka transistor dapat dijadikan saklar dengan pemutus dan penyambungnya berupa (tegangan pada basisnya).



Gambar 2.9 Rangkaian Transistor Mengontrol Beban  
(Kustija, 2014)

Persamaan transistor memberikan :

$$I_c = \beta I_B \quad (2-1)$$

B = penguatan transistor

dari persamaan tersebut, jika  $I_B = 0$  maka  $I_c = 0$

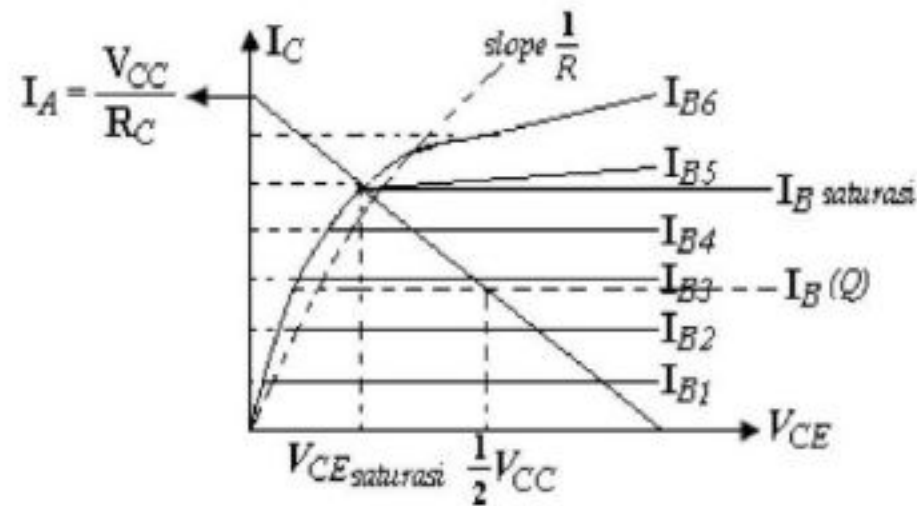
(transistor tidak mengantarkan arus  $I_c$ , dengan kata lain posisi *cut-off* atau mati).

Rangkaian tersebut memperoleh persamaan sebagai berikut :

$$I_b = \frac{V_b - V_{ce}}{R_b} \quad (2-2)$$

$$I_c = \frac{V_c - V_{ce}}{R_c} \quad (2-3)$$

Sedangkan karakteristik keluaran transistor dan garis beban adalah sebagai berikut :



Dari gambar tersebut pada kondisi saturasi (jenuh)  $I_B$  tidak dapat menaikkan  $I_C$ . Selanjutnya pada  $I_{B5} - I_{B6}$  menghasilkan  $I_C$  yang sama dengan  $I_C$  saturasi.

Pada kondisi ini, diperoleh :

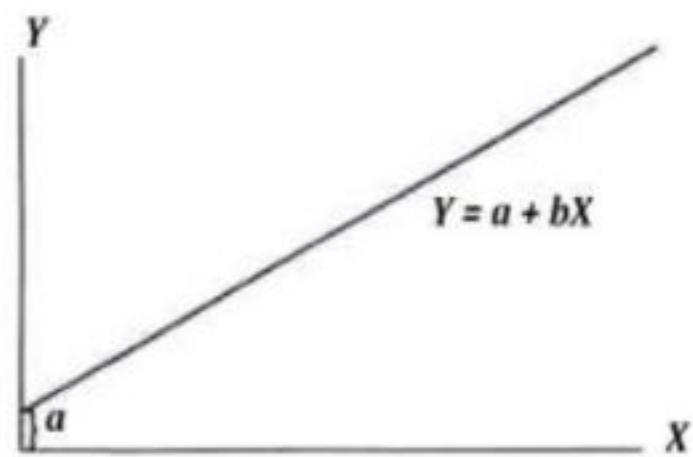
$$V_{ce} \cong 0 \quad (2-4)$$

$$I_c = \frac{V_{cc}}{R_c} \quad (2-5)$$

Artinya arus besar, tegangan menuju nol (0). Dapat dikatakan hambatan pada VCE, menuju nol (sebagai saklar ON) jadi untuk membuat transistor berlaku sebagai saklar yang ON, kita memberikan tegangan  $V_s$  yang mengakibatkan transistor saturasi. Sedang jika  $V = 0$  maka  $I = 0$ , dan  $I_c = 0$ . Maka pada kondisi ini transistor pada kondisi tidak menghantarkan arus  $I_c$  sama dengan kondisi saklar terbuka (Kustija, 2014).

### 2.2.13 Persamaan Regresi Linear Sederhana

Persamaan regresi linier sederhana merupakan suatu model persamaan yang menggambarkan hubungan satu variabel bebas/ *predictor* (X) dengan satu variabel tak bebas/ *response* (Y), yang biasanya digambarkan dengan garis lurus, seperti pada gambar berikut :



Gambar 2.10 Ilustrasi Garis Regresi Linear  
(Nazir, 1983)

Persamaan regresi linier sederhana secara matematik diekspresikan oleh :

$$Y = a + bX \quad (2-6)$$

yang mana :

$Y$  = garis regresi/ *variable response*

$a$  = konstanta (intersep), perpotongan dengan sumbu vertikal

$b$  = konstanta regresi (*slope*)

$X$  = variabel bebas/ *predictor*

Besarnya konstanta  $a$  dan  $b$  dapat ditentukan menggunakan persamaan :

$$a = \frac{(\sum y)(\sum x^2) - (\sum x)(\sum xy)}{n(\sum x^2) - (\sum x)^2} \quad (2-7)$$

$$b = \frac{n(\sum xy) - (\sum x)(\sum y)}{n(\sum x^2) - (\sum x)^2} \quad (2-8)$$

yang mana  $n$  = jumlah data

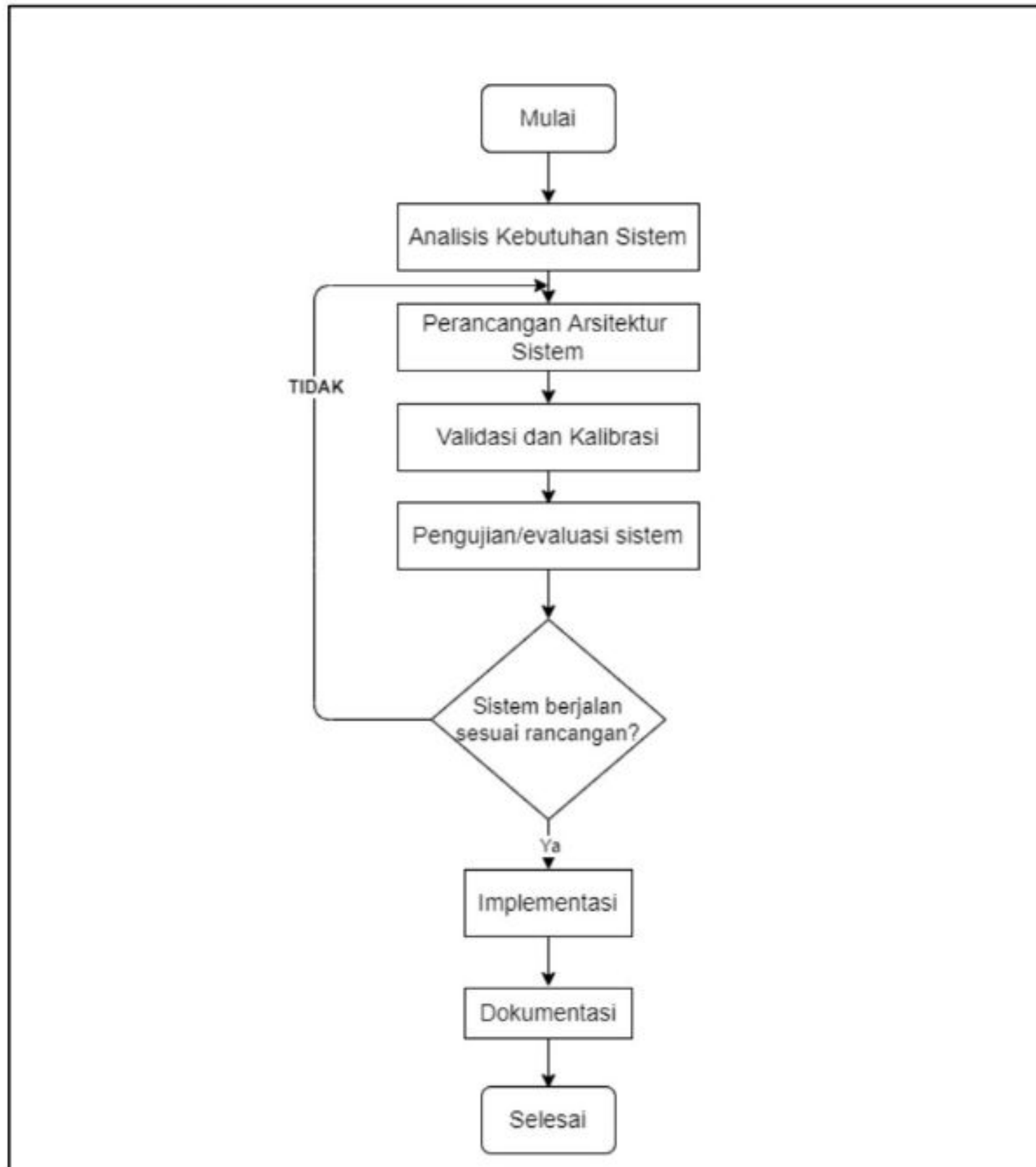
Adapun langkah-langkah yang perlu dilakukan untuk melakukan analisis dan uji regresi linier sederhana adalah sebagai berikut :

1. Menentukan tujuan dari Analisis Regresi Linear Sederhana
2. Mengidentifikasi variabel *predictor* dan variabel *response*
3. Melakukan pengumpulan data dalam bentuk tabel
4. Menghitung  $X^2$ ,  $XY$  dan total dari masing-masingnya
5. Menghitung  $a$  dan  $b$  menggunakan rumus yang telah ditentukan
6. Membuat model Persamaan Garis Regresi
7. Melakukan prediksi terhadap variabel *predictor* atau *response*
8. Uji signifikansi menggunakan Uji-t dan menentukan Taraf Signifikan (Anas, 1996).

## BAB III METODOLOGI PENELITIAN

### 3.1 Diagram Tahap Pelaksanaan

Pada bab ini akan di bahas metode penelitian syang akan peneliti gunakan. Dalam pelaksanaan penelitian ini peneliti menggunakan metode pendekatan *flowcart*. Tahapan – tahapan yang akan dilakukan dapat dilihat pada Gambar 3.1



Gambar 3.1 Diagram Tahap Pelaksanaan

Gambar 3.1 merupakan diagram tahapan rancangan dan penelitian penerapan IoT untuk sistem pengendalian lampu taman berbasis *NodeMCU*. Adapun tahapan pelaksanaan penelitian dapat dijelaskan sebagai berikut:

1. Analisis kebutuhan merupakan langkah pertama dalam membangun sistem penerapan *IoT* untuk sistem pengendalian lampu taman. Analisis kebutuhan seperti perangkat apa saja yang digunakan untuk membuat sistem pengendalian lampu taman dan *output* yang akan diperoleh dari sistem.
2. Perancangan arsitektur sistem, yang didasari dari tahap pertama dalam membuat sistem sistem pengendalian lampu taman, seperti dalam hal perancangan alur kerja, perancangan alat dan penerapan sistem berbasis Telegram.
3. Kalibrasi, pada tahapan ini akan dilaksanakan kalibrasi untuk mendapatkan nilai maksimal dan minimum dari arus yang masuk ke lampu sebagai nilai acuan dalam mengetahui keadaan dari lampu taman.
4. Pengujian dan evaluasi sistem, merupakan tahapan untuk melakukan uji coba terhadap sistem yang telah dibangun guna mengetahui kekurangan-kekurangan yang dialami oleh sistem yang dibangun berjalan sesuai kebutuhan. Jika sistem yang dirancang tidak sesuai dengan kebutuhan maka akan kembali ke tahap arsitektur sistem. Jika sistem dirancang sesuai dengan kebutuhan maka akan dilanjutkan ke tahap seterusnya.
5. Implementasi, merupakan tahapan penyusunan dalam membangun sistem pengendalian lampu taman.
6. Dokumentasi, merupakan tahapan terakhir dimana akan dibuat berupa laporan dari hasil uji coba dari sistem evaluasi sistem.

### 3.2 Analisa Kebutuhan Sistem

Pada tahap analisis kebutuhan sistem ini akan dilakukan analisis terhadap kebutuhan pada perangkat keras dan perangkat lunak.

#### 3.2.1 Perangkat Keras (*hardware*)

Perangkat keras (*hardware*) yang dibutuhkan pada perancangan ini adalah sebagai berikut :

No	Nama komponen	Jumlah komponen
1	<i>NodeMCU ESP8266</i>	1
2	Komponen <i>Relay</i>	4
3	Sensor Arus <i>ACS 712</i>	4

4	<i>Light Emitting Diode (LED)</i>	4
5	Laptop Acer Aspire E 14	1
6	<i>Smartphone POCO X3 Pro</i>	1
7	<i>Liquid Crystal Display (LCD)</i>	1
8	Arduino UNO	1
9	<i>Real Time Clock (RTC)</i>	1
10	<i>Breadboard</i>	1
11	Kabel Jumper	31

### 3.2.2 Perangkat Lunak (*software*)

Perangkat lunak (*software*) yang dibutuhkan pada perancangan ini adalah sebagai berikut :

No	Nama <i>Software</i>
1	Arduino IDE
2	Fritzing
3	Telegram
4	Draw.io

### 3.3 Prosedur Perancangan

Secara sistematis, prosedur perancangan dalam pelaksanaan rancangan tugas akhir dapat diuraikan sebagai berikut :

#### 1. Studi literatur

Proses pembelajaran teori-teori yang digunakan pengumpulan literatur-literatur berupa buku referensi, artikel-artikel, serta jurnal-jurnal untuk mendukung dalam penyusunan usulan tugas akhir ini. Tahap ini dilakukan agar penelitian memiliki pengetahuan dasar yang kuat dan menjadi dasar untuk tahap perancangan alat pengendalian lampu taman.

#### 2. Perancangan Perangkat Lunak

Perancangan ini menggunakan program aplikasi Arduino IDE yang digunakan pada modul *NodeMcu* dengan bahasa pemrograman C. Aplikasi program IDE ini digunakan untuk menampilkan data sensor. Hasil data

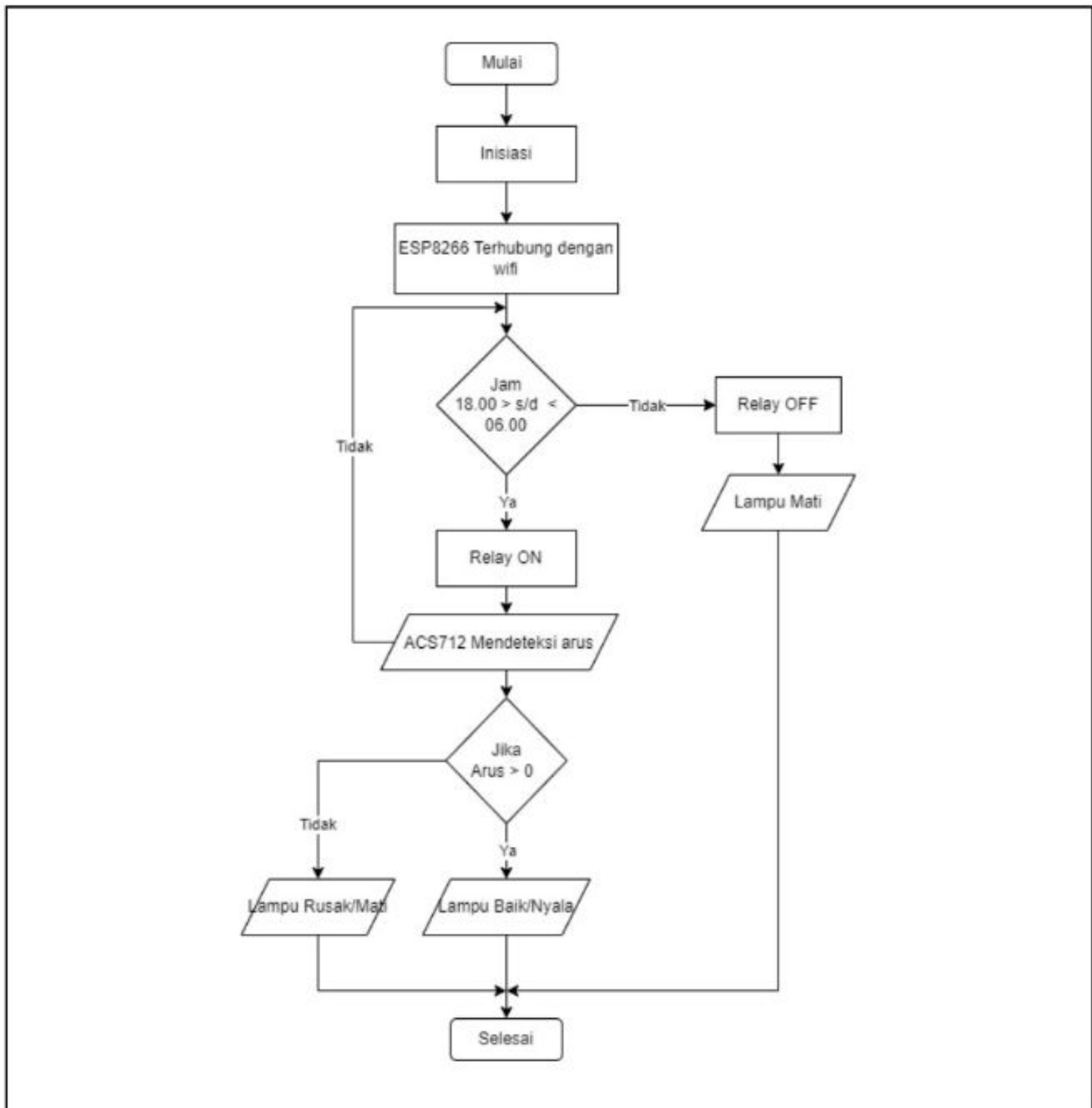
program yang dibuat ditampilkan melalui web yang ditampilkan dari keluaran *NodeMcu* dan disimpan dalam *file* dengan ekstensi *.ino*.

### 3. Perancangan Perangkat Keras

Pada tahap ini dilakukan rancangan berupa komponen pendukung dari sistem yang akan dibangun. Perancangan ini terdiri dari sensor arus ACS712, *NodeMCU*, *Arduino Uno*, *RTC*, *LCD*, *Relay*, dan *Lampu*.

### 3.4 Perancangan Sistem

Berikut merupakan *flowchart* sistem dengan menggunakan waktu pada sistem pengendalian lampu taman dapat dilihat pada Gambar 3.2.



Gambar 3.2 *Flowchart* Sistem Pengendalian Lampu Taman

Gambar 3.2 merupakan *flowcart* dari sistem manajemen pengaturan lampu dengan sistem penjadwalan. Pada pukul 18.00 > s/d < 06.00 WITA, maka *relay* akan menyala (ON). Jika sensor arus > 0, sensor akan mendeteksi arus yang masuk pada lampu dan lampu akan menyala. Jika arus = 0, sensor akan mendeteksi bahwa lampu terindikasi rusak atau mati. Jika tidak, maka *relay* (OFF) kemudian lampu mati.

Pada sistem pengendalian lampu taman terdapat mode *controller* yang di atur sesuai waktu dan jumlah lampu yang menyala yakni sebagai berikut :

*Normal Controller* :

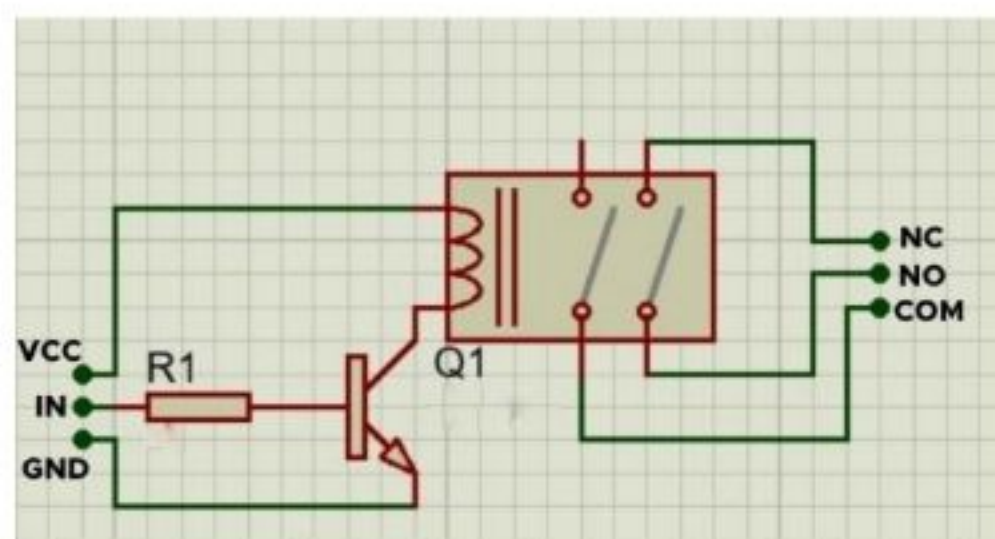
1. Pada pukul 18.00 s/d 21.00 lampu akan menyala sebanyak 4 lampu.
2. Pada pukul 21.00 s/d 24.00 lampu akan menyala sebanyak 2 lampu.
3. Pada pukul 24.00 s/d 06.00 lampu akan menyala sebanyak 1 lampu.

*Emergency Controller* :

1. Pada dalam keadaan *emergency* lampu akan menyala sebanyak 4 lampu dengan mengirim perintah "*emergency on*" maka lampu akan menyala.
2. Perintah "*3 light on*" lampu akan menyala sebanyak 3 lampu.
3. Perintah "*2 light on*" lampu akan menyala sebanyak 2 lampu.
4. Perintah "*1 light on*" lampu akan menyala sebanyak 1 lampu dan
5. Perintah kendali 4 lampu lainnya.

### 3.4.1 Perancangan Perangkat Keras (*hardware*)

Perancangan perangkat keras pengendalian lampu taman terdiri dari membuat rangkaian merujuk pada Gambar 3.3 berikut :



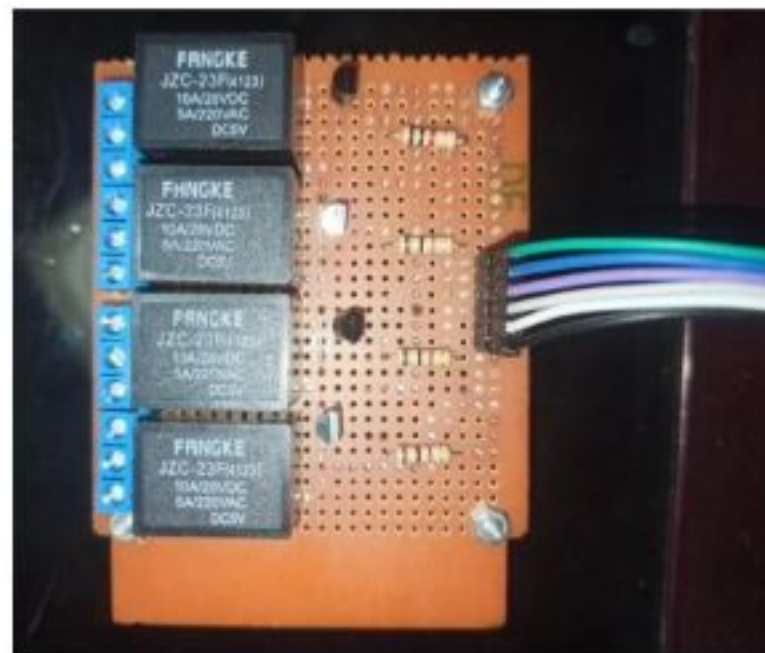
Gambar 3.3 Rangkaian *Relay*

Berdasarkan Gambar 3.3 Merupakan *relay* dalam bentuk rangkaian yang di buat menggunakan *proteus*. Koil relay terhubung pada tegangan supply (VCC) dan kaki



kolektor pada NPN Transistor. pada kaki basis pada NPN Transistor terhubung Resistor Basis (Rb) yang kemudian terhubung pada kaki mikrokontroler. Sedangkan kaki emittor pada NPN transistor terhubung pada ground. Adanya resistor tersebut dikarenakan arus di basis harus lebih kecil dari kolektor sehingga *relay* dapat menahan arus balik akibat induksi medan magnet saat koil dialiri arus listrik. Kendali *relay* sepenuhnya dari *mikrokontroller* yang menentukan besar kecilnya nilai output yang diberikan.

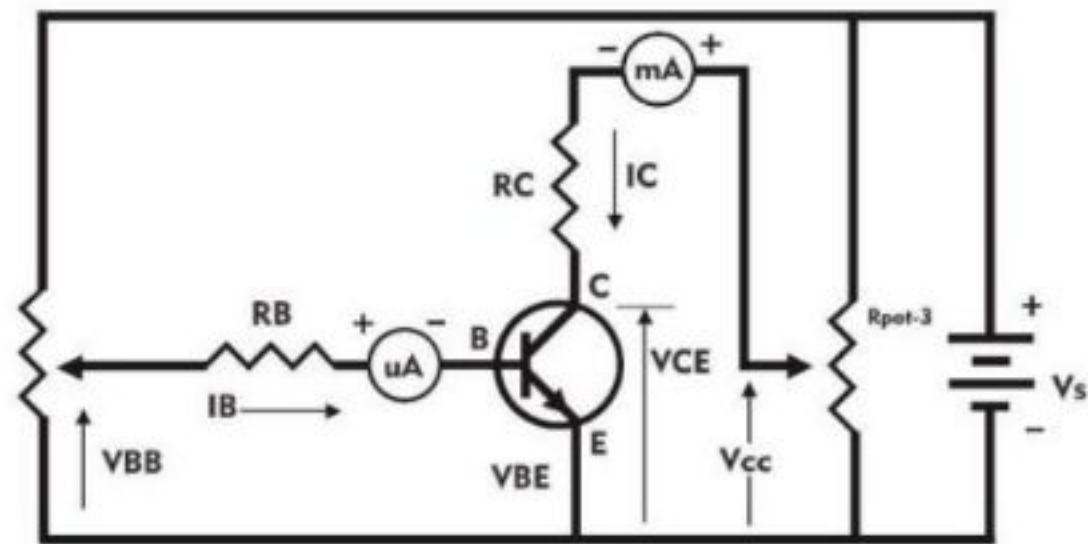
Berikut merupakan gambar dari rangkaian relay yang buat menggunakan PCB merujuk pada Gambar 3.4 berikut :



Gambar 3.4 Komponen *Relay*

Berdasarkan Gambar 3.4 Merupakan *relay* dalam bentuk rangkaian yang di buat menggunakan PCB. Sama seperti penjelasan mengenai gambar 3.4 diatas disini digunakan *relay* 5v, Transistor BC547 dengan tipe NPN sehingga agar *relay* bekerja dibasis *high* atau aktif *high* dan resistor 10k. Ketika sinyal yang dikirimkan arduino ke transistor maka akan mengubah transistor menjadi mode saturasi yakni arus basis untuk memicu kolektor ke emitter, kemudian dihubungkan melalui PCB. *Relay* digambar tersebut berjumlah 4 *channel* yang difungsikan untuk menghubungkan 4 lampu LED dikarenakan *relay* difungsikan sebagai saklar otomatis untuk lampu.

Perhitungan *relay* menggunakan *transistor* BC547 sebagai berikut :



Nilai *transistor* BC547 pada *datasheet* didapatkan nilai minimum hfe sebesar 110 dimana  $hfe = \beta$  sebagai berikut :

Diketahui :

$$V_{cc} = 5 \text{ volt}$$

$$R_c = 0.0625 \text{ k} = 62.5 \Omega$$

$$V_{be} = 0,7$$

$$\beta = 110$$

$$R_b = 10k$$

Dicari :

$$I_c = \dots?$$

$$I_b = \dots?$$

$$R_b = \dots?$$

$$V_{ce} = \dots?$$

Penyelesaian :

$$I_c = \frac{V_{cc}}{R_c} \tag{3-1}$$

$$I_c = \frac{5}{62.5}$$

$$I_c = 0.08 \text{ A}$$

$$I_c = 80000 \mu\text{A}$$

$$I_b = \frac{I_c}{\beta} \quad (3-2)$$

$$I_b = \frac{80000}{110}$$

$$I_b = 727.27 \mu A$$

$$V_{ce} = V_c - I_c R_c \quad (3-3)$$

$$V_{ce} = 5 - (0.08 \times 62.5)$$

$$V_{ce} = 5 - (5)$$

$$V_{ce} = 0 \text{ volt (Saturasi)}$$

$$R_b = \frac{V_{cc} - V_{be}}{I_b} \quad (3-4)$$

$$R_b = \frac{5 - 0.7}{727.27}$$

$$R_b = 5.9 \text{ k (Minimum)}$$

Ketika input = 0 volt ,  $I_B = 0 \text{ A}$  dan  $I_c = 0 \text{ A}$

$$V_c = V_{cc} - (I_c R_c) \quad (3-5)$$

$$V_c = 5 - (0 \times 62.5)$$

$$V_c = 5 - 0$$

$$V_c = 5 \text{ volt}$$

Ketika input = 5 volt

$$V_c = V_{cc} - (I_c R_c) \quad (3-6)$$

$$V_c = 5 - (0.08 \times 62.5)$$

$$V_c = 5 - (5)$$

$$V_c = 0 \text{ volt (saturasi)}$$

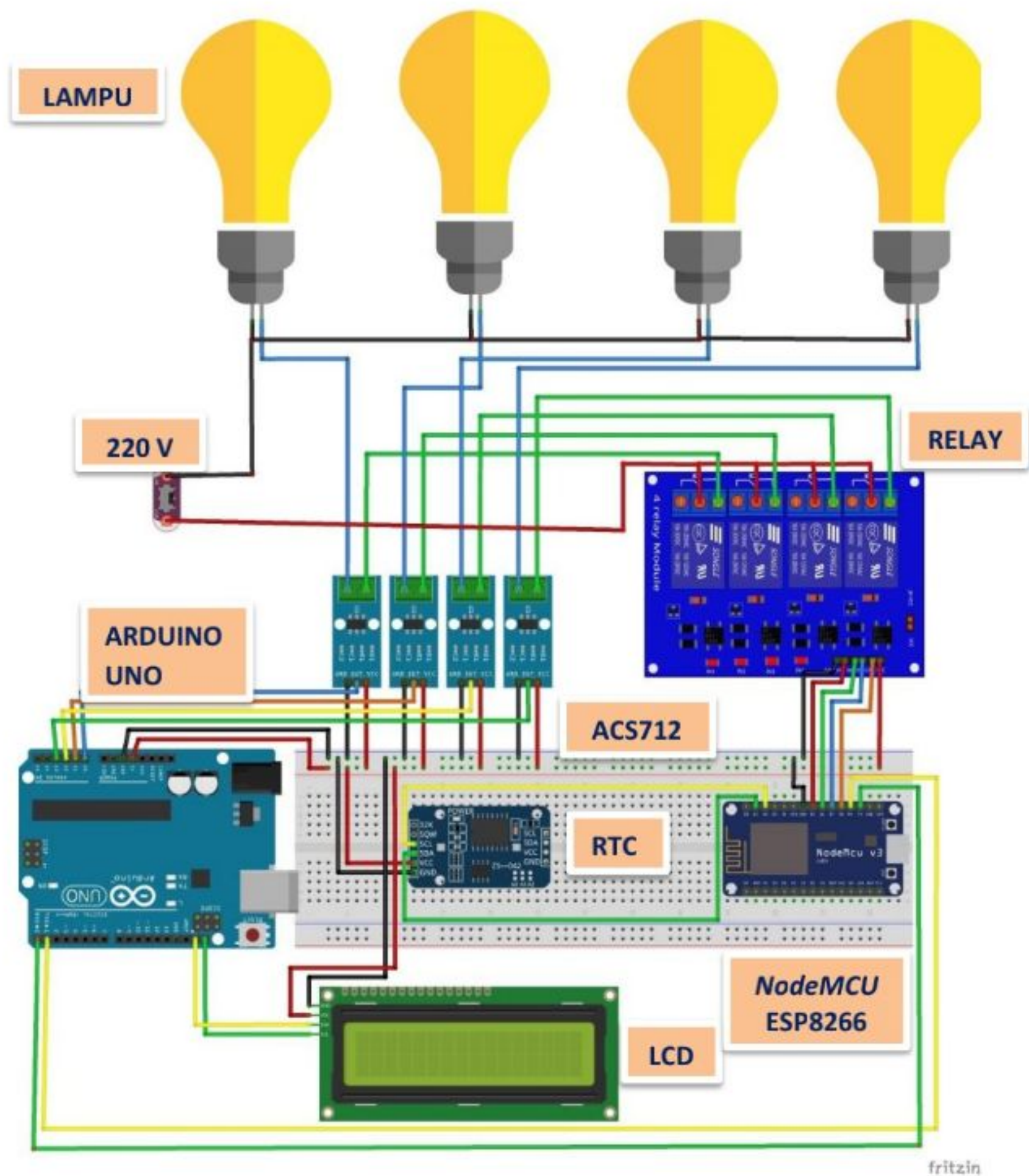
$$I_b = \frac{V_c - V_{be}}{R_b} \quad (3-7)$$

$$I_b = \frac{5 - 0.7}{10k}$$

$$I_b = 4.300 \mu A$$

Dari perhitungan tersebut dapat di lihat bahwa *resistor* sebesar  $10k$  menghasilkan arus sebesar  $4.300 \mu A$  dan arus *cutoff* nya sebesar  $727.27 \mu A$  dan dibutuhkan arus sebesar  $0.08 A$  dan untuk menggerakkan koil pada *relay*. Karena arusnya lebih besar maka transistornya dalam mode saturasi (bekerja penuh) pada saat tegangan input sebesar  $5V$ .

Dari tahapan rangkaian tersebut terbentuklah rangkaian secara utuh yang dirancang menggunakan aplikasi Fritzing. Rangkaian perangkat keras pada gambar 3.5 berikut.



Gambar 3.5 Rangkaian Perangkat Keras.

1. Sumber tegangan

Digunakan sumber tegangan dari adaptor 5V, adaptor 5V digunakan untuk mensuplai tegangan dari mikrokontroller dan juga *relay* beserta sensor dan juga digunakan untuk mensuplai Lampu.

2. Mikrokontroller

Ada dua mikrokontroller yang digunakan yaitu Arduino UNO dan ESP8266. Arduino berfungsi untuk mengirimkan data pembacaan arus pada ACS712 melalui pin analog dan RTC untuk membaca waktu yang disimpan pada RTC dan menuju ESP8266 secara serial. ESP8266 berfungsi untuk menghubungkan perangkat ke IoT menggunakan *platform* Telegram.

3. Sensor ACS712

Digunakan untuk membaca nilai arus yang masuk ke lampu dan ditampilkan pada LCD.

4. *Relay*

Digunakan sebagai saklar untuk mematikan dan menyalakan lampu lewat kendali Telegram

5. RTC

Digunakan untuk mengatur waktu untuk mematikan dan menyalakan lampu.

6. LCD

Digunakan sebagai *interface* tambahan untuk menampilkan nilai arus yang tersedia.

7. Lampu

Berfungsi untuk beban dan juga objek utama pada penelitian laporan tugas akhir ini.

### 3.4.2 Perancangan Perangkat Lunak (*software*)

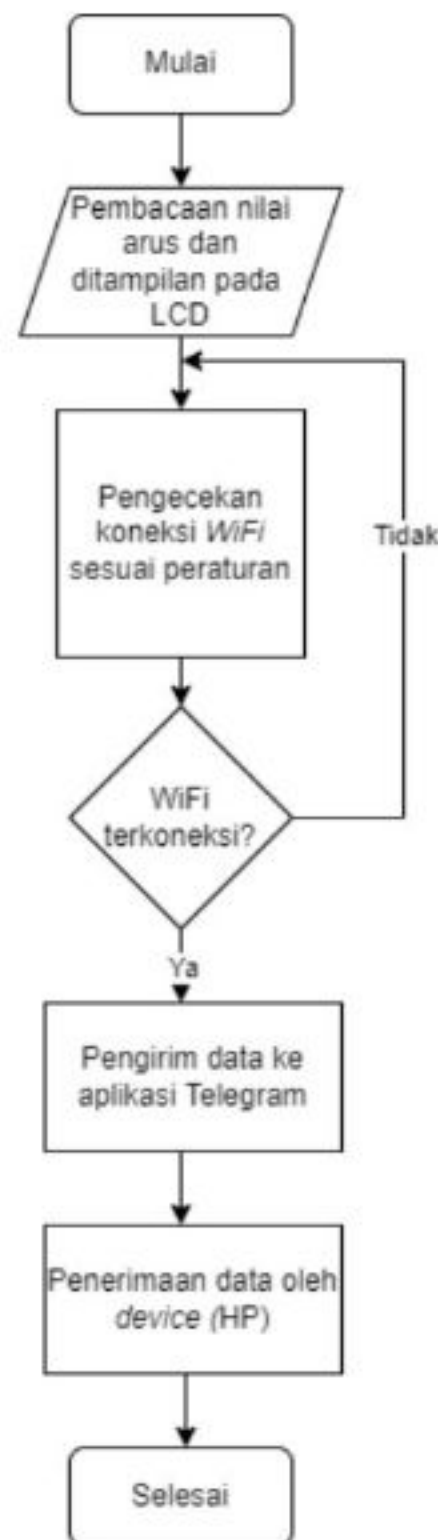
Perancangan perangkat lunak dilakukan dengan membuat program pada *software* Arduino IDE. Program yang dibuat akan berdasarkan dari skema perangkat keras yang telah dirancang dan setelah perangkat dapat dijalankan dengan program yang telah dibuat maka dilakukan pemanggilan fungsi dari mikrokontroller *NodeMCU* ESP8266 ke Bot Telegram.

## 1. Perancangan *Software*



Gambar 3.6 Diagram Alir Perancangan *Software*.

## 2. Pembuatan program *NodeMCU* ESP8266



Gambar 3.7 Diagram Alir Program *NodeMCU* ESP8266.

Berdasarkan Gambar 3.7 dapat dilihat bahwa terdapat beberapa proses yang dilakukan oleh mikrokontroler *NodeMCU* ESP8266 adalah melakukan pembacaan data yang terbaca oleh sensor yakni data dari sensor arus acs712, kemudian *NodeMCU* ESP8266 akan melakukan pengecekan koneksi internet atau *WiFi* sesuai yang telah diprogramkan. Pada proses ini terdapat dua buah kondisi yang mana apabila kondisi 'ya' yang artinya *WiFi* terkoneksi maka *NodeMCU* ESP8266 akan mengirimkan data hasil pembacaan sensor arus ke aplikasi Telegram dengan mengirim kata "status" maka telegram akan mengirimkan data sensor arus tiap lampu, sedangkan kondisi 'tidak' yang artinya *WiFi* tidak terkoneksi maka sistem harus kembali ke proses pengecekan koneksi kembali. Setelah terkoneksi dan melewati proses pengiriman data maka selanjutnya data akan diterima oleh *device* dan proses selesai.

### 3. Pembuatan program Arduino IDE



Gambar 3.8 Diagram Alir Program Pada Arduino Uno

Berdasarkan Gambar 3.8 dapat dilihat bahwa Arduino Uno bertugas untuk mengolah data hasil pembacaan sensor arus yang akan dikirimkan ke mikrokontroler *NodeMCU* ESP8266 secara serial.



### **3.5 Pengujian Alat**

Pengujian yang dilakukan meliputi :

#### **1. Pengujian Sensor**

Pengujian sensor dilakukan dengan cara mengukur tingkat kesalahan perhitungan sistem yang dilakukan oleh sensor serta menguji tingkat kesalahan perhitungan oleh sensor itu sendiri.

#### **2. Pengujian keseluruhan sistem**

Pengujian keseluruhan sistem dilakukan dengan menyambungkan blok-blok perangkat keras dan mengoperasikan sistem sehingga dapat diketahui apakah alat ini bekerja sesuai dengan spesifikasi yang diharapkan. Setelah perangkat keras telah beroperasi seperti yang diharapkan, perangkat lunak yang telah dibuat diujikan bersama perangkat kerasnya.

## BAB IV

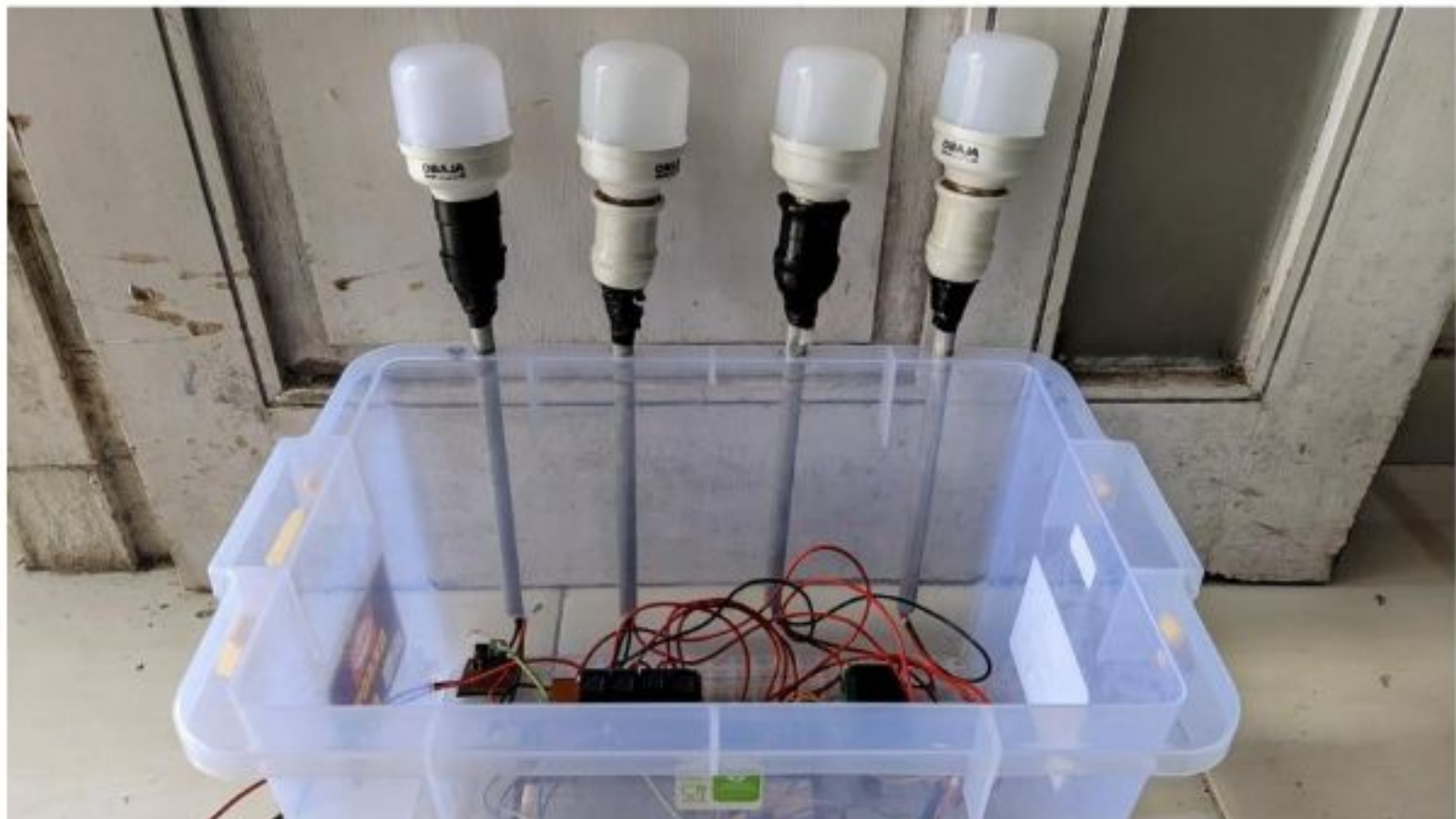
### HASIL DAN PEMBAHASAN

Pada bab ini akan dibahas hasil dari penerapan *Internet Of Things* untuk sistem pengendalian lampu taman berbasis *NodeMCU* ESP8266 yang meliputi pembahasan tentang bagaimana cara alat bekerja dan hasil pengujian yang telah dilakukan beserta pembahasannya.

#### 4.1 Hasil Perancangan

##### 4.1.1 Perangkat Keras (*Hardware*)

Gambar 4.1 menunjukkan hasil perancangan perangkat keras dari pengendalian lampu taman berbasis *NodeMCU* ESP8266.



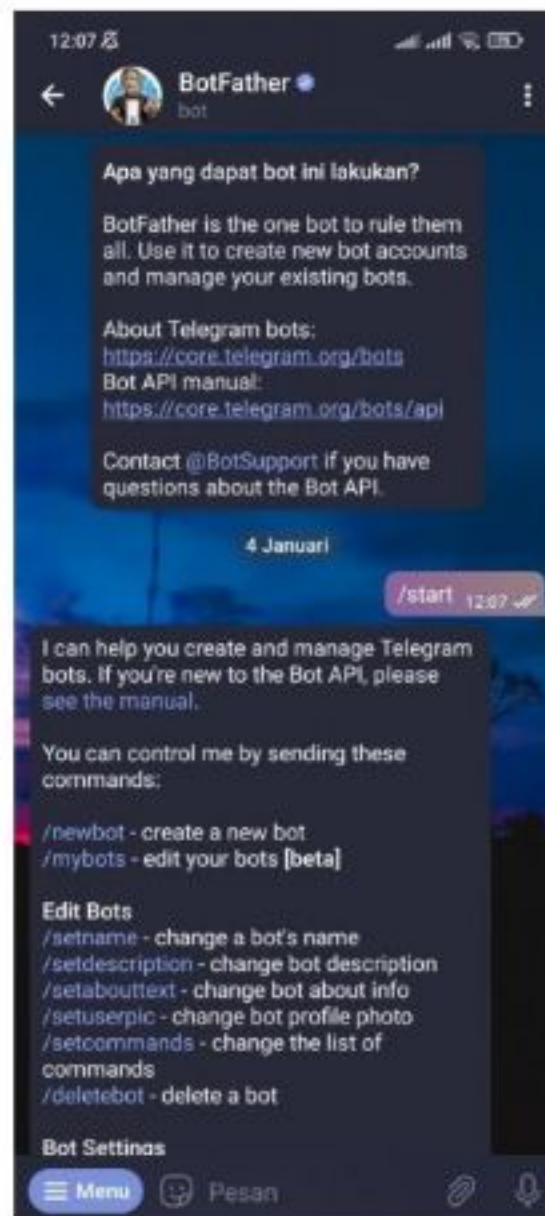
Gambar 4.1 Hasil perancangan *prototype* lampu taman.

Pada bagian pertama terdapat komponen disusun diatas akrilik yang berisi *NodeMCU* ESP8266, Arduino UNO, RTC, LCD, Komponen *relay 4 channel*. Kemudian terdapat 4 lampu yang terhubung ke *relay* sebagai beban dan juga objek utama dari penelitian tugas akhir ini.

## 4.1.2 Perangkat Lunak (*Software*)

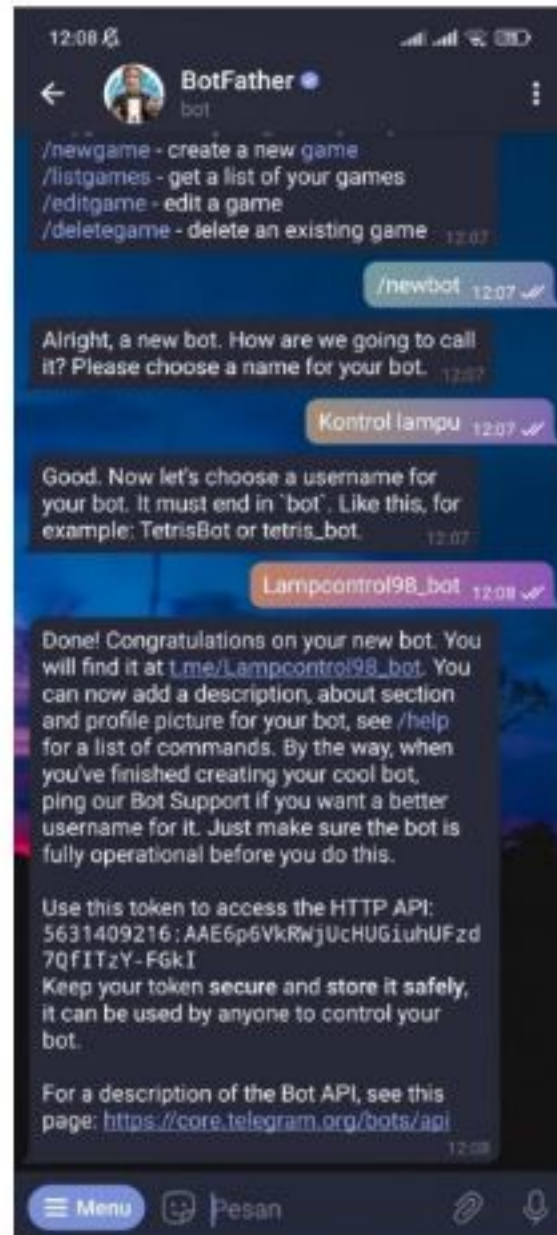
### 1. *Bot* Telegram

*Bot* Telegram digunakan untuk membuat sebuah bot baru pada akun Telegram pengguna. Telegram dilengkapi dengan sebuah fitur yang dinamakan “*Bot Father*” yang digunakan untuk memulai membuat bot. Langkah awal untuk dapat mengirim data ke aplikasi Telegram yaitu melakukan pembuatan akun dengan cara mengetikkan “*Bot Father*” pada kolom pencarian dan melakukan *setup* seperti berikut:



Gambar 4.2 Tampilan *bot father* (a).

Pada Gambar 4.2 merupakan tampilan awal setelah menambahkan akun *Bot Father* yang mana untuk memulai pembuatan dan mengatur *bot* Telegram, *user* harus menuliskan ‘/start’.



Gambar 4.3 Tampilan *bot father* (b).

*/newbot* digunakan untuk membuat sebuah *bot* baru. Setelah itu, akan muncul pesan otomatis yang berisikan perintah untuk memilih sebuah nama untuk *bot* yang akan digunakan oleh pengguna. Setelah berhasil membuat sebuah *bot* telegram, *user* akan diberikan sebuah *token* http API (*Application Programming Interface*) yang akan dimasukkan kedalam program *NodeMCU* ESP8266.

## 2. ID dan Tampilan *Bot* yang telah dibuat



Gambar 4.4 Tampilan *bot* yang telah dibuat user.

Setiap akun Telegram mempunyai ID yang berbeda-beda yang dapat digunakan untuk memberikan izin akses mengakses *bot* yang telah terhubung melalui Modul ESP8266. Sebelumnya, pada program diharuskan untuk memasukkan *token* yang telah diberikan oleh *Bot Father* lalu memasukkan ID Telegram sehingga *user* dapat mengakses *bot* tersebut menggunakan akun Telegramnya. Untuk memulai pengaksesan, *user* dapat mengetikkan "*status*" untuk mendapatkan informasi mengenai arus tiap lampu. *Bot* Telegram yang telah dibuat harus dihubungkan ke Arduino melalui Modul ESP8266 yang terhubung dengan *WiFi*.

## 4.2 Pengujian Sistem

Pengujian sistem dilakukan untuk mengetahui fungsi dari sistem yang telah dibuat apakah dapat beroperasi sesuai perancangan dengan baik atau tidak. Proses pengujian dilakukan pada masing- masing bagian atau komponen, diantaranya :

1. Pengujian sensor ACS712 dengan multimeter digital sebagai acuan
2. Pengujian aplikasi Telegram
3. Pengujian sistem keseluruhan

### 4.2.1 Pengujian Sensor ACS712 dengan Multimeter Digital Sebagai Acuan

Pengujian sensor ACS712 dilakukan dengan cara membandingkan hasil data yang terukur oleh sensor ACS712 dengan pengukuran menggunakan Multimeter Digital DT9205A. Pengujian dilakukan dengan beberapa sampel lampu dengan *watt* yang berbeda-beda. Berikut adalah hasil dari pengujian lampu menggunakan sensor arus ACS712 dan Multimeter Digital DT9205A dapat dilihat pada Tabel 4.1.



Gambar 4.5 Pengukuran arus lampu menggunakan sensor ACS712.



Gambar 4.6 Pengukuran arus lampu menggunakan Multimeter Digital DT9205A.

Pengujian sensor dilakukan untuk mengetahui apakah sensor sudah bekerja dengan baik dan sesuai dengan program. Pengujian sensor dilakukan dengan membandingkan hasil pembacaan sensor dengan alat ukur yang ada. Perbandingan dilakukan dengan perhitungan persentase *error* agar diketahui tingkat akurasi data hasil pembacaan sensor menggunakan persamaan berikut ini.

$$Error(\%) = \left| \frac{\text{nilai referensi} - \text{nilai sensor}}{\text{nilai referensi}} \right| \times 100$$

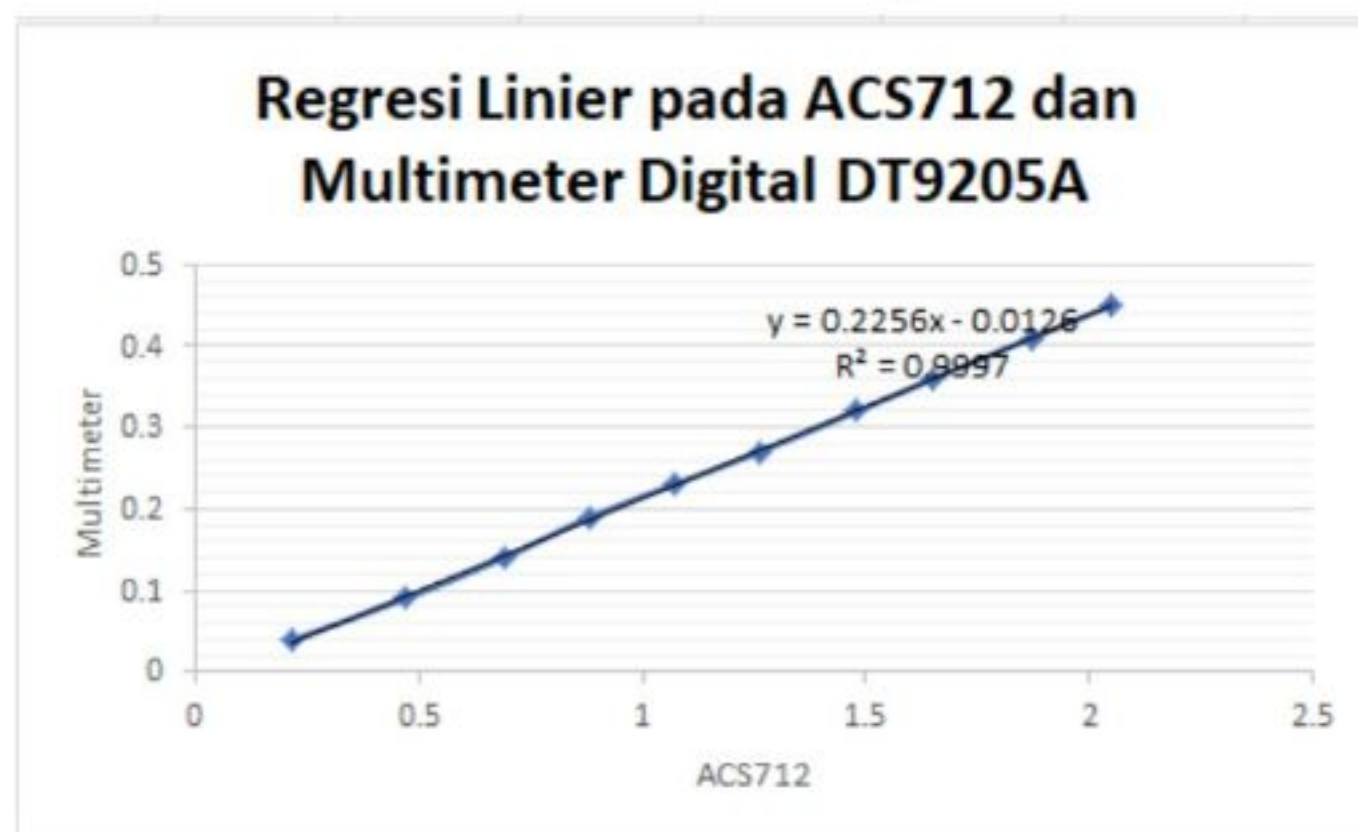
$$\text{Rata - rata error } (\%) = \frac{\sum(\text{nilai error})}{\text{banyaknya Data}}$$

Untuk mencari nilai arus pada sensor ACS712 digunakan modul Arduino untuk mengubah nilai analog menjadi nilai digital atau ADC. Kemudian modul arduino dihubungkan ke laptop untuk dimasukkan program pada aplikasi Arduino IDE berupa library `#include "ACS712.h"` dan `#define pinADC A0` untuk membaca sensor arus yang terhubung pada Arduino. Selanjutnya pada `void loop()` dimasukkan perintah `sensor1.calibrate();` dan `arus = sensor1.getCurrentAC();` sebagai script untuk menampilkan arus yang keluar pada serial monitor. Hasil dari arus yang keluar dapat dilihat pada Tabel 4.1 berikut :

Tabel 4.1 Pengukuran arus lampu menggunakan sensor ACS712 dan pengukuran arus lampu menggunakan Multimeter Digital DT9205A

No	Daya lampu (watt)	Arus Pada Sensor ACS712 (A)	Arus Pada Multimeter Digital DT9205A (A)	Tegangan pada Multimeter DT9205A (V)	Persentase Error (%)
1	10	0.22	0.04	219.3	81.81
2	20	0.47	0.09	218.7	80.85
3	30	0.69	0.14	219.1	79.71
4	40	0.88	0.19	218.9	78.40
5	50	1.07	0.23	219.3	78.50
6	60	1.26	0.27	218.8	78.57
7	70	1.48	0.32	219.4	78.37
8	80	1.65	0.36	219.9	78.18
9	90	1.87	0.41	219.8	78.07
10	100	2.05	0.45	219.9	78.04
Rata-Rata Persentase Error					79.05

Pada Tabel 4.1 terdapat 10 analisis hasil data pengukuran lampu dengan daya sebesar 10 *watt* sampai dengan 100 *watt*. Data yang diambil berupa nilai ACS712 serta nilai dari multimeter digital DT9205A. Terdapat juga nilai tegangan di tiap *watt* lampu yang dikukur dengan Multimeter Digital. Nilai dari Multimeter merupakan nilai ukur arus yang mengalir pada tiap lampu. Rata-rata persentase *error* yang didapatkan dengan perbandingan antara arus pada sensor ACS712 dan Multimeter Digital sebesar 79.05% yang dimana hasil tersebut masih jauh dari datang arus yang didapatkan pada Multimeter Digital. Data tersebut kemudian di proses dengan metode regresi linier agar nilai pada ACS712 mendekati nilai dari Multimeter Digital.



Gambar 4.7 Grafik Kalibrasi Linier Sensor ACS712

Nilai korelasi didapatkan dengan memasukkan nilai variable X dan variable Y dari sensor ACS712 dan Multimeter Digital DT9205A pada *Microsoft Excel* kemudian menggunakan fungsi *scatter* pada menu *insert*. Selanjutnya klik kanan pada diagram, pilih *select data*, pilih *add*, lalu masukkan nilai variabel x dan y, klik pilihan *Chart Layouts* dan pilih *Layout 9*, maka akan didapatkan nilai  $R^2$  seperti pada gambar 4.7. Dengan kriteria nilai korelasi sebagai berikut :

- 0,00 – 0,199 : Korelasi sangat lemah
- 0,20 – 0,399 : Korelasi lemah
- 0,40 – 0,599 : Korelasi cukup
- 0,60 – 0,799 : Korelasi kuat
- 0,80 – 1,000 : Korelasi sangat kuat



Berdasarkan gambar diatas dapat disimpulkan bahwa sensor ACS712 yang digunakan memiliki korelasi yang sangat kuat dengan nilai  $R^2$  sebesar 0.9997 sehingga layak digunakan untuk penelitian ini.

Tabel 4.2 Hasil Perhitungan Regresi Linier

No.	X	Y	X <sup>2</sup>	Y <sup>2</sup>	XY
1	0.22	0.04	0.0484	0.0016	0.0088
2	0.47	0.09	0.2209	0.0081	0.0423
3	0.69	0.14	0.4761	0.0196	0.0966
4	0.88	0.19	0.7744	0.0361	0.1672
5	1.07	0.23	1.1449	0.0529	0.2461
6	1.26	0.27	1.5876	0.0729	0.3402
7	1.48	0.32	2.1904	0.1024	0.4736
8	1.65	0.36	2.7225	0.1296	0.594
9	1.87	0.41	3.4969	0.1681	0.7667
10	2.05	0.45	4.2025	0.2025	0.9225
$\Sigma$ Total	11.64	2.5	16.8646	0.7938	3.658

Selanjutnya dilakukan perhitungan regresi untuk mengkalibrasi sensor sehingga nilai yang dikeluarkan sama dengan alat ukur sebenarnya. Dengan rumus regresi menggunakan persamaan (2.6) sebagai berikut :

$$Y = a + bX$$

Dimana :

Y= variable akibat

a= konstanta

b= koefisien regresi/kemiringan

X= variable faktor (Alat Ukur)

rumus perhitungan nilai a dan b menggunakan persamaan (2.7) dan (2.8) sebagai berikut :

$$a = \frac{(\sum y)(\sum x^2) - (\sum x)(\sum xy)}{n(\sum x^2) - (\sum x)^2}$$

$$b = \frac{n(\sum xy) - (\sum x)(\sum y)}{n(\sum x^2) - (\sum x)^2}$$

Dari persamaan regresi linier, didapatkan persamaan kalibrasi yang akan dimasukkan pada program, yaitu :

$$Y = 0.2256x - 0.0126$$

Nilai x adalah *output* dari data sensor ACS712 dan dimasukkan ke dalam nilai regresi maka hasilnya terdapat pada Tabel 4.3 berikut ini :

Tabel 4.3 Nilai ACS712 dimasukkan ke dalam Regresi Linier

No	Daya lampu (watt)	Arus Pada Sensor ACS712 (A)	Arus Pada Multimeter Digital DT9205A (A)	Tegangan pada Multimeter DT9205A (V)	Persentase Error (%)
1	10	0.04	0.04	219.3	0
2	20	0.09	0.09	218.7	0
3	30	0.14	0.14	219.1	0
4	40	0.19	0.19	218.9	0
5	50	0.23	0.23	219.3	0
6	60	0.28	0.27	218.8	3.57
7	70	0.33	0.32	219.4	3.03
8	80	0.36	0.36	219.9	0
9	90	0.41	0.41	219.8	0
10	100	0.45	0.45	219.9	0
Rata – Rata persentase Error					0.66

Pada Tabel 4.3 didapatkan hasil yang dimana nilai ACS712 dimasukkan pada hasil regresi linier. Hasil yang dapatkan berupa rata-rata persentase *error* sebesar 0.66% dimana hasil tersebut sudah mendekati hasil dari pengukuran arus lampu menggunakan Multimeter Digital DT9205A.

Berikut Script untuk pengukur sensor arus ACS712 :

```
#include "ACS712.h"
#define pinADC A0
ACS712 sensor1(ACS712_30A, A0);
float arus;

void setup() {
  // put your setup code here, to run once:
  Serial.begin(115200);
}
void loop() {
  sensor1.calibrate();
  arus = sensor1.getCurrentAC();
  Serial.print("Arus : ");
  Serial.println(arus);
  float y = (0.2256 * arus) - 0.0126;
  Serial.print("Regresi : ");
  Serial.println(y);
  Serial.println(" ");
  \
  delay(1000);
}
```

Script tersebut adalah program untuk membaca nilai arus AC menggunakan sensor ACS712, melakukan kalibrasi sensor, melakukan perhitungan regresi linier untuk mengkonversi nilai yang dibaca menjadi besaran arus, dan menampilkan nilai arus dan regresi pada serial monitor. Baris pertama program dilakukan inklusi library ACS712.h yang digunakan untuk mengakses fungsi-fungsi dari sensor ACS712. Kemudian, pada baris kedua, dilakukan definisi `pinADC` yang dihubungkan dengan pin A0 pada board Arduino. Pada blok fungsi `loop`, pertama-tama dilakukan kalibrasi sensor dengan memanggil fungsi `calibrate()` pada objek `sensor1`. Setelah itu, nilai arus AC diambil dari sensor dengan memanggil fungsi `getCurrentAC()` pada objek `sensor1` dan disimpan dalam variabel `arus`. Kemudian, nilai arus ditampilkan pada serial monitor dengan menggunakan fungsi `Serial.print()` dan `Serial.println()`. Selanjutnya, dilakukan perhitungan regresi linier pada variabel `arus` dan disimpan dalam variabel `y`. Hasil perhitungan regresi juga ditampilkan pada serial monitor.

#### 4.2.2 Pengujian Aplikasi Telegram

Pengujian pada aplikasi Telegram dilakukan untuk memastikan data hasil pengukuran dari sensor arus ACS712 dapat ditampilkan pada laman *bot* telegram yang telah diberi nama “Kontrol Lampu”. Laman bot ini berisi informasi status berupa kuat arus yang mengalir ke lampu dan juga sebagai *switch* untuk menyalakan dan mematikan lampu. Untuk mendapatkan informasi kondisi lampu, dilakukan dengan cara mengetik “*status*” pada *bubble chat* ‘Kontrol Lampu’ maka akan didapatkan data berupa kuat arus yang mengalir di tiap lampu yang dipasang dengan sensor ACS712. Tampilan hasil purwarupa kendali lampu taman dapat dilihat pada Gambar 4.8.



Gambar 4.8 Tampilan laman bot Telegram user.

Berikut merupakan perintah dari kendali lampu menggunakan aplikasi Telegram dapat di lihat pada tabel 4.4 berikut :

Tabel 4.4 Perintah Kendali Lampu Menggunakan Telegram

No	Perintah Kendali	Keterangan
1	"3 light on"	Perintah "3 light on" lampu akan menyala sebanyak 3 lampu.
2	"2 light on"	Perintah "2 light on" lampu akan menyala sebanyak 2 lampu.
3	"1 light on"	Perintah "1 light on" lampu akan menyala sebanyak 1 lampu
4	"light 1 on"	Perintah "light 1 on" lampu 1 akan menyala.
5	"light 2 on"	Perintah "light 2 on" lampu 2 akan menyala.
6	"light 3 on"	Perintah "light 3 on" lampu 3 akan menyala.
7	"light 4 on"	Perintah "light 4 on" lampu 4 akan menyala.
8	"light 1 2 on"	Perintah "light 1 2 on" lampu 1 dan 2 akan menyala.
9	"light 1 3 on"	Perintah "light 1 3 on" lampu 1 dan 3 akan menyala.
10	"light 1 4 on"	Perintah "light 1 4 on" lampu 1 dan 4 akan menyala.
11	"light 1 3 on"	Perintah "light 1 3 on" lampu 1 dan 3 akan menyala.
12	"light 2 3 on"	Perintah "light 2 3 on" lampu 2 dan 3 akan menyala.
13	"light 2 4 on"	Perintah "light 2 4 on" lampu 2 dan 4 akan menyala.
14	"light 3 4 on"	Perintah "light 3 4 on" lampu 3 dan 4 akan menyala.
15	"light 1 2 3 on"	Perintah "light 1 2 3 on" lampu 1, 2 dan 3 akan menyala.
16	"light 1 2 4 on"	Perintah "light 1 2 4 on" lampu 1, 2 dan 4 akan menyala.
17	"light 1 3 4 on"	Perintah "light 1 3 4 on" lampu 1, 3 dan 4 akan menyala.
18	"light 2 3 4 on"	Perintah "light 2 3 4 on" lampu 2, 3 dan 4 akan menyala.
19	"emergency off"	Perintah "emergency off" semua lampu akan mati.
20	"emergency on"	Perintah "emergency on" semua lampu akan menyala.
21	"Status"	Perintah "Status" melihat arus yang mengalir tiap lampu

Pada Tabel 4.4 dapat dilihat terdapat 21 perintah kendali lampu menggunakan aplikasi telegram. Ketika *NodeMCU* ESP8266 terhubung dengan internet maka telegram akan merespon dengan notifikasi "*Connected To Kiflan*" dan kendali telegram tersebut dapat digunakan dengan mengirimkan perintah sesuai tabel tersebut. Hasil percobaan dengan perintah kendali berjalan sesuai dengan perintah yang diberikan dan kendali lampu menggunakan aplikasi Telegram dinyatakan berhasil untuk digunakan.

### 4.2.3 Pengujian Sistem Keseluruhan

Pengujian sistem keseluruhan dilakukan dengan menguji ke empat titik lampu. Selanjutnya pengujian dilakukan dengan mengganti salah satu titik lampu yang rusak untuk menguji alat apakah berjalan sesuai dengan rumusan masalah.

1. Pengujian empat titik lampu dengan perintah “*Emergency on*” pada Telegram



Gambar 4.9 Pengujian keseluruhan dengan empat titik lampu normal.

Berdasarkan Gambar 4.9 pengujian empat titik lampu dengan kondisi menyala dengan memberi perintah “*status*” pada Telegram untuk mengetahui arus yang mengalir pada setiap titik lampu di dapatkan hasil arus pada lampu 1 sebesar 0.02 A sehingga tampilan pada Telegram menunjukkan adanya arus yang mengalir pada lampu 1. Pada lampu 2 didapatkan arus sebesar 0.02 A sehingga tampilan pada Telegram menunjukkan adanya arus yang mengalir pada lampu 2. Pada lampu 3 didapatkan arus sebesar 0.03 A sehingga tampilan pada Telegram menunjukkan adanya arus yang mengalir pada lampu 3. Pada lampu 4 didapatkan arus sebesar 0.02 A sehingga tampilan pada Telegram menunjukkan adanya arus yang mengalir pada lampu 4.

2. Pengujian empat titik lampu dengan perintah “*Emergency on*” pada Telegram dengan kondisi titik lampu 2 dalam keadaan rusak.



Gambar 4.10 Pengujian keseluruhan dengan salah satu titik lampu rusak.

Berdasarkan Gambar 4.10 pengujian empat titik lampu dengan kondisi menyala dengan memberi perintah “*status*” pada Telegram untuk mengetahui arus yang mengalir pada setiap titik lampu di dapatkan hasil arus pada lampu 1 sebesar 0.02 A sehingga tampilan pada Telegram menunjukkan adanya arus yang mengalir pada lampu 1. Pada lampu 2 didapatkan arus sebesar 0 A sehingga tampilan pada Telegram menunjukkan tidak adanya arus yang mengalir pada lampu 2 sehingga terindikasi lampu 2 rusak. Pada lampu 3 didapatkan arus sebesar 0.03 A sehingga tampilan pada Telegram menunjukkan adanya arus yang mengalir pada lampu 3. Pada lampu 4 didapatkan arus sebesar 0.02 A sehingga tampilan pada Telegram menunjukkan adanya arus yang mengalir pada lampu 4.

Berdasarkan dua kondisi di atas dapat disimpulkan bahwa pengujian yang dilakukan menunjukkan sesuai dengan rumusan masalah dimana arus yang mengalir ke lampu normal ditampilkan pada Telegram sedangkan lampu yang terindikasi rusak tidak ada arus yang mengalir ke lampu dan ditampilkan pada Telegram.

Tabel 4.5 Perintah Kendali Lampu dengan Aplikasi Telegram

No	Perintah Kendali	Keterangan
1	“ <i>3 light on</i> ”	Perintah “ <i>3 light on</i> ” lampu akan menyala sebanyak 3 lampu.
2	“ <i>2 light on</i> ”	Perintah “ <i>2 light on</i> ” lampu akan menyala sebanyak 2 lampu.
3	“ <i>1 light on</i> ”	Perintah “ <i>1 light on</i> ” lampu akan menyala sebanyak 1 lampu
4	“ <i>light 1 on</i> ”	Perintah “ <i>light 1 on</i> ” lampu 1 akan menyala.
5	“ <i>light 2 on</i> ”	Perintah “ <i>light 2 on</i> ” lampu 2 akan menyala.
6	“ <i>light 3 on</i> ”	Perintah “ <i>light 3 on</i> ” lampu 3 akan menyala.
7	“ <i>light 4 on</i> ”	Perintah “ <i>light 4 on</i> ” lampu 4 akan menyala.
8	“ <i>light 1 2 on</i> ”	Perintah “ <i>light 1 2 on</i> ” lampu 1 dan 2 akan menyala.
9	“ <i>light 1 3 on</i> ”	Perintah “ <i>light 1 3 on</i> ” lampu 1 dan 3 akan menyala.
10	“ <i>light 1 4 on</i> ”	Perintah “ <i>light 1 4 on</i> ” lampu 1 dan 4 akan menyala.
11	“ <i>light 1 3 on</i> ”	Perintah “ <i>light 1 3 on</i> ” lampu 1 dan 3 akan menyala.
12	“ <i>light 2 3 on</i> ”	Perintah “ <i>light 2 3 on</i> ” lampu 2 dan 3 akan menyala.
13	“ <i>light 2 4 on</i> ”	Perintah “ <i>light 2 4 on</i> ” lampu 2 dan 4 akan menyala.
14	“ <i>light 3 4 on</i> ”	Perintah “ <i>light 3 4 on</i> ” lampu 3 dan 4 akan menyala.

15	<i>"light 1 2 3 on"</i>	Perintah <i>"light 1 2 3 on"</i> lampu 1, 2 dan 3 akan menyala.
16	<i>"light 1 2 4 on"</i>	Perintah <i>"light 1 2 4 on"</i> lampu 1, 2 dan 4 akan menyala.
17	<i>"light 1 3 4 on"</i>	Perintah <i>"light 1 3 4 on"</i> lampu 1, 3 dan 4 akan menyala.
18	<i>"light 2 3 4 on"</i>	Perintah <i>"light 2 3 4 on"</i> lampu 2, 3 dan 4 akan menyala.
19	<i>"emergency off"</i>	Perintah <i>"emergency off"</i> semua lampu akan mati.
20	<i>"emergency on"</i>	Perintah <i>"emergency on"</i> semua lampu akan menyala.
21	<i>"Status"</i>	Perintah <i>"Status"</i> melihat arus yang mengalir tiap lampu

Pada Tabel 4.5 dapat dilihat terdapat 21 perintah kendali lampu menggunakan aplikasi Telegram. Perintah kendali lampu tersebut sama dengan perintah kendali pada Tabel 4.4 yang dimana ketika *NodeMCU* ESP8266 terhubung dengan internet maka telegram akan merespon dengan notifikasi *"Connected To Kiflan"* dan kendali telegram tersebut dapat digunakan dengan mengirimkan perintah sesuai tabel tersebut. Namun yang membedakan pada percobaan ini yaitu pengujian dengan salah satu titik lampu yang rusak dimana diberikan *"emergency on"* untuk menyalakan semua lampu tersebut dan kemudian memberikan perintah *"Status"* untuk melihat arus yang mengalir pada tiap lampu. Hasilnya ketika lampu rusak diberikan maka status dari arus pada menyatakan bahwa lampu itu rusak.



## BAB V

### KESIMPULAN DAN SARAN

#### 5.1 Kesimpulan

Berdasarkan perancangan, pengujian, dan pengamatan yang telah dilakukan pada penelitian Penerapan *Internet Of Things* Untuk Sistem Pengendalian Lampu Taman Berbasis *NodeMCU* ESP8266 dapat disimpulkan bahwa :

1. Setelah dilakukan pengujian, sistem yang dibuat memberikan informasi secara otomatis tentang arus yang masuk di setiap lampunya secara online menggunakan Aplikasi Telegram.
2. Berdasarkan pengujian, sistem yang dibuat berjalan sesuai dengan yang diinginkan dengan menggunakan RTC. Lampu akan menyala pada malam hari sesuai waktu yang ditentukan dan mati saat pagi sampai sore, sehingga menghindari penerangan yang terus-menerus sepanjang hari. Hal ini membuktikan bahwa alat tersebut efektif dalam menghemat penggunaan listrik pada lampu.
3. Berdasarkan pengujian dan analisis 10 data lampu dan diukur menggunakan sensor ACS712 dan dilakukan perhitungan menggunakan metode regresi linier dengan nilai pada multimeter digital didapatkan rata-rata *error* sebesar 0.66% yang dimana sudah mendekati hasil dari nilai pengukuran multimeter digital.
4. Sistem alat ini mampu diterapkan pada kendali lampu lainnya seperti *smarthome*, lampu jalan dan kendali lampu lainnya yang menggunakan *platform IoT*.

#### 5.2 Saran

Adapun yang dapat penulis sarankan dalam penulisan tugas akhir ini adalah karena tugas akhir ini bersifat prototype sehingga perancangan masih jauh dari sempurna, saran dari penulis antara lain:

1. Menambah parameter lain yang dapat dikontrol dan dimonitoring.
2. Menambah sensor cahaya untuk mendeteksi kondisi keadaan cahaya yang dikarenakan oleh cuaca hujan.

## DAFTAR PUSTAKA

- Amin, M. (2020). "*InfoTekjar : Jurnal Nasional Informatika dan Teknologi Jaringan Sistem Cerdas Kontrol Kran Air Menggunakan Arduino dan Sensor Ultrasonic.*"2, pp. 1-4.
- Atmel. (2008). *Belajar Sendiri Mikrokontroler*. Yogyakarta: Yayasan PUIL.
- Azzahra. N.F. (2018). "Rancang Bangun Purwarupa iot ( Internet Of Things ) Kendali Lampu dengan Protokol Mqtt pada Studi Kasus Skala Rumah Tangga," *J. Ilm.*, vol. 3, no. 1, pp. 1-4.
- Budi Novianto. (2016). "Rancang Bangun Kendali Dan Monitoring Lampu Dengan Teknologi Short Messege Service (SMS)". Fakultas Ilmu Komputer, Universitas Narotama, Surabaya.
- Basith, M. A. (2017). *Penerapan Sensor Ultrasonik HC-SR04 Pada Sistem Pengukur Volume pada Mobil Tangki Air Bersih*. Laporan Akhir, Program Studi Teknik Elektronika, Jurusan Teknik Elektro, Pendidikan Diploma III, Politeknik Negeri Sriwijaya, Palembang.
- Dwi W. Suryana. (2012). "Rancang Bangun Sistem Monitoring Tegangan, Arus dan Temperatur Pada Sistem Pencatu Daya Listrik Di Teknik Elektro Berbasis Mikrokontroler ATmega 128." 4:4. 44-500.
- Dewi. N. H. L. (2019). "Prototype Smart Home Dengan Modul NodeMCU ESP8266 Berbasis Internet of Things (IoT)," Repository UNIM, vol. 2019, pp. 1-9.
- Febbry. (2018). *Belajar Elektronika*. Jakarta: Elex Media Komputiondo.
- Gajah, C. N. (2018). *Memfaatkan Sensor DHT22 Sebagai Pendeteksi Kelembaban Tanah Berbasis Arduino*. Tugas Akhir, Program Studi D-III Fisika, Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Sumatera Utara, Medan.
- Imam, M.K, G. Abdillah, and R. Yuniarti. (2018). "Pembuatan Sistem Pengontrol Intensitas Cahaya Lampu Menggunakan Raspberry Pi Berbasis Mobile," pp. 302-306.

- Irsyam, M. (2019). *Sistem Otomasi Penyiraman Tanaman Berbasis Telegram*. Sigma Teknika, 2(1), 81. <https://doi.org/10.33373/sigma.v2i1.1834>
- Jaja Kustija, M.Sc. (2014). *Transistor Sebagai Saklar*. Elektronika Industri. <http://maulana.lecture.ub.ac.id/files/2014/09/TR-SBG-SAKLAR.pdf>
- Mugi Alan Prasetya, & Rachmat Aulia, (2020). "*Prototype Penerangan Lampu Taman Otomatis Menggunakan Arduino Uno*".
- Margaretha, R. (2018). "*Pengendali Lampu Rumah Menggunakan Aplikasi HP Android Melalui Komunikasi Bluetooth Berbasis Mikrokontroler Atmega 328,*" Universitas Sumatera Utara.
- Mehta, M. (2015). *Esp 8266 : a Breakthrough in Wireless Sensor Networks and Internet Of Things*, 6(8), 7–11.
- Nazir, M. (1983). *Metode Statistika Dasar I*, Gramedia Pustaka Utama:Jakarta
- Rijalul Imam, I Gede Putu Wirarama Wedashwara W, & Fitri Bimantoro. (2020). "*Rancang Bangun Sistem Monitoring dan Controlling Penerangan Jalan Umum Berbasis IoT dan Android,*" *JTIKA*. Vol. 2, No. 1, p. 101.
- Siregar, & Simon. (2011). "*Rancang Bangun Sistem Monitoring Ruangan Terintegrasi Berbasis Ethernet*".
- Sudijono Anas, (1996). "*Pengantar Statistik Pendidikan*", Jakarta:Rajawa
- Syamsuri, T.U. (2015). "*Kontrol Lampu Jalan Untuk Menghemat Energi,*" vol. 7, pp. 302-33.
- Tanjung, Akbar. (2015). *Aplikasi Liquid Crystal Display (LCD) 16x2 Sebagai Tampilan pada Coconut Milk Auto Machine*. Laporan Akhir, Jurusan Teknik Elektro, Program Studi Teknik Elektronika, Politeknik Negeri Sriwijaya, Palembang.

## LAMPIRAN

### Lampiran 1. Program NodeMCU ESP8266

```
#include "CTBot.h"
#include "RTClib.h"
#include <Wire.h>

CTBot myBot;

RTC_DS3231 rtc;

String ssid = "Kiflan";
String pass = "12345678";
String token = "5631409216:AAE6p6VkrWjUcHUGiuhUFzd7QfITzY-FGkI";
const int id = 1570970193;
TBMessage msg;

int relay1 = 14; //12;
int relay2 = 12; //14;
int relay3 = 13; //13;
int relay4 = 15; //15;
float arus1, arus2, arus3, arus4;
int jam, menit, detik;
int flag = 0;
String data, koneksi;
String arrData[6];
String status1, status2, status3, status4;
float batas = 0;

void setup() {
  // put your setup code here, to run once:
  Serial.begin(115200);
  rtc.begin();
  Wire.begin(5, 4);
  rtc.adjust(DateTime(__DATE__, __TIME__));
  // rtc.adjust(DateTime(2014, 1, 21, 3, 0, 0));
  myBot.wifiConnect(ssid, pass);
  myBot.setTelegramToken(token);
  koneksi = (String)"Connected To " + ssid;
  if (myBot.testConnection()){
```

```

// Serial.println("Connected To");
myBot.sendMessage(id, koneksi);
}

pinMode(relay1, OUTPUT);
pinMode(relay2, OUTPUT);
pinMode(relay3, OUTPUT);
pinMode(relay4, OUTPUT);
}

void loop() {
    DateTime now = rtc.now();
    jam = now.hour();
    menit = now.minute();

    String data = "";
    Serial.println("Meminta Data Dari Arduino...");
    while (Serial.available() > 0) {
        data += char (Serial.read());
    }
    data.trim();
    if(data != ""){
        int index = 0;
        for (int i=0; i<= data.length(); i++){
            char delimiter = '#';
            if (data[i] != delimiter)
                arrData [index] += data[i];
            else
                index ++;
        }
    }
    // Serial.print(index);
    Serial.print(jam);
    if(index == 3){
        Serial.println("Berhasil menerima data dari Arduino");
        arus1 = arrData[0].toFloat();
        arus2 = arrData[1].toFloat();
        arus3 = arrData[2].toFloat();
        arus4 = arrData[3].toFloat();
    }
    // jam = arrData[4].toInt();
    // menit = arrData[5].toInt();
}

```

```

}
arrData[0] = "";
arrData[1] = "";
arrData[2] = "";
arrData[3] = "";
// arrData[4] = "";
// arrData[5] = "";
}

if (arus1 <= batas){
    status1 = "Lampu Rusak";
}
else{
    status1 = "Lampu Nyala";
}
if (arus2 <= batas){
    status2 = "Lampu Rusak";
}
else{
    status2 = "Lampu Nyala";
}
if (arus3 <= batas){
    status3 = "Lampu Rusak";
}
else{
    status3 = "Lampu Nyala";
}
if (arus4 <= batas){
    status4 = "Lampu Rusak";
}
else{
    status4 = "Lampu Nyala";
}

// put your main code here, to run repeatedly:
data = (String)"Waktu " + jam + (String)" : " + menit +
// '\n' + (String)"RTC Status : " + flag +
'\n' + (String)"Lampu 1 : " + arus1 + (String)" A " + status1 +
'\n' + (String)"Lampu 2 : " + arus2 + (String)" A " + status2 +
'\n' + (String)"Lampu 3 : " + arus3 + (String)" A " + status3 +

```

```

'\n' + (String)"Lampu 4 : " + arus4 + (String)" A " + status4;

if (flag == 0){
  if (jam >= 18 && jam <= 21){
    digitalWrite(relay1 ,HIGH);
    digitalWrite(relay2 ,HIGH);
    digitalWrite(relay3 ,HIGH);
    digitalWrite(relay4 ,HIGH);
  }

  else if (jam >= 21 && jam <= 24){
    digitalWrite(relay1 ,HIGH);
    digitalWrite(relay2 ,HIGH);
    digitalWrite(relay3 ,LOW);
    digitalWrite(relay4 ,LOW);
  }

  else if (jam >= 24 || jam <= 6){
    digitalWrite(relay1 ,HIGH);
    digitalWrite(relay2 ,LOW);
    digitalWrite(relay3 ,LOW);
    digitalWrite(relay4 ,LOW);
  }

  else{
    digitalWrite(relay1 ,LOW);
    digitalWrite(relay2 ,LOW);
    digitalWrite(relay3 ,LOW);
    digitalWrite(relay4 ,LOW);
  }
}

// if (jam == 0 || menit == 0){
//   flag = 0;
// }

if (myBot.getNewMessage(msg)) {
  if (msg.text.equalsIgnoreCase("status")) {
    myBot.sendMessage(id, data);
  }
}

```

```

else if (msg.text.equalsIgnoreCase("emergency on")) {
    myBot.sendMessage(id, "Emergency Light On");
    if (flag == 0 || flag ==1){
        flag = 1;
        digitalWrite(relay1 ,HIGH);
        digitalWrite(relay2 ,HIGH);
        digitalWrite(relay3 ,HIGH);
        digitalWrite(relay4 ,HIGH);
    }
//    myBot.sendMessage(id, "flag " + (String)flag);
}
else if (msg.text.equalsIgnoreCase("rtc on")) {
myBot.sendMessage(id, "RTC On");
if (flag == 0 || flag ==1){
    flag = 0;
}
}
else if (msg.text.equalsIgnoreCase("rtc off")) {
myBot.sendMessage(id, "RTC Off");
if (flag == 0 || flag ==1){
    flag = 1;
}
}
else if (msg.text.equalsIgnoreCase("light 1 on")) {
myBot.sendMessage(id, "Light 1 On");
if (flag == 0 || flag ==1){
    flag = 1;
    digitalWrite(relay1 ,HIGH);
    digitalWrite(relay2 ,LOW);
    digitalWrite(relay3 ,LOW);
    digitalWrite(relay4 ,LOW);
}
//    myBot.sendMessage(id, "flag " + (String)flag);
}
else if (msg.text.equalsIgnoreCase("light 2 on")) {
myBot.sendMessage(id, "Light 2 On");
if (flag == 0 || flag ==1){
    flag = 1;
    digitalWrite(relay1 ,LOW);
    digitalWrite(relay2 ,HIGH);
}
}

```



```

        digitalWrite(relay3 ,LOW);
        digitalWrite(relay4 ,LOW);
    }
//    myBot.sendMessage(id, "flag " + (String)flag);
}
else if (msg.text.equalsIgnoreCase("light 3 on")) {
myBot.sendMessage(id, "Light 3 On");
if (flag == 0 || flag ==1){
    flag = 1;
    digitalWrite(relay1 ,LOW);
    digitalWrite(relay2 ,LOW);
    digitalWrite(relay3 ,HIGH);
    digitalWrite(relay4 ,LOW);
}
//    myBot.sendMessage(id, "flag " + (String)flag);
}

else if (msg.text.equalsIgnoreCase("light 4 on")) {
myBot.sendMessage(id, "Light 4 On");
if (flag == 0 || flag ==1){
    flag = 1;
    digitalWrite(relay1 ,LOW);
    digitalWrite(relay2 ,LOW);
    digitalWrite(relay3 ,LOW);
    digitalWrite(relay4 ,HIGH);
}
//    myBot.sendMessage(id, "flag " + (String)flag);
}
else if (msg.text.equalsIgnoreCase("light 1 2 on")) {
myBot.sendMessage(id, "Light 1 And 2 On");
if (flag == 0 || flag ==1){
    flag = 1;
    digitalWrite(relay1 ,HIGH);
    digitalWrite(relay2 ,HIGH);
    digitalWrite(relay3 ,LOW);
    digitalWrite(relay4 ,LOW);
}
//    myBot.sendMessage(id, "flag " + (String)flag);
}
else if (msg.text.equalsIgnoreCase("light 1 3 on")) {

```

```

myBot.sendMessage(id, "Light 1 And 3 On");
if (flag == 0 || flag ==1){
    flag = 1;
    digitalWrite(relay1 ,HIGH);
    digitalWrite(relay2 ,LOW);
    digitalWrite(relay3 ,HIGH);
    digitalWrite(relay4 ,LOW);
}
//    myBot.sendMessage(id, "flag " + (String)flag);
}
else if (msg.text.equalsIgnoreCase("light 1 4 on")) {
myBot.sendMessage(id, "Light 1 And 4 On");
if (flag == 0 || flag ==1){
    flag = 1;
    digitalWrite(relay1 ,HIGH);
    digitalWrite(relay2 ,LOW);
    digitalWrite(relay3 ,LOW);
    digitalWrite(relay4 ,HIGH);
}
//    myBot.sendMessage(id, "flag " + (String)flag);
}
else if (msg.text.equalsIgnoreCase("light 2 3 on")) {
myBot.sendMessage(id, "Light 2 And 3 On");
if (flag == 0 || flag ==1){
    flag = 1;
    digitalWrite(relay1 ,LOW);
    digitalWrite(relay2 ,HIGH);
    digitalWrite(relay3 ,HIGH);
    digitalWrite(relay4 ,LOW);
}
//    myBot.sendMessage(id, "flag " + (String)flag);
}
else if (msg.text.equalsIgnoreCase("light 2 4 on")) {
myBot.sendMessage(id, "Light 2 And 4 On");
if (flag == 0 || flag ==1){
    flag = 1;
    digitalWrite(relay1 ,LOW);
    digitalWrite(relay2 ,HIGH);
    digitalWrite(relay3 ,LOW);
    digitalWrite(relay4 ,HIGH);
}
}

```

```

    )
//    myBot.sendMessage(id, "flag " + (String)flag);
}
else if (msg.text.equalsIgnoreCase("light 3 4 on")) {
myBot.sendMessage(id, "Light 3 And 4 On");
if (flag == 0 || flag ==1){
    flag = 1;
    digitalWrite(relay1 ,LOW);
    digitalWrite(relay2 ,LOW);
    digitalWrite(relay3 ,HIGH);
    digitalWrite(relay4 ,HIGH);
}
//    myBot.sendMessage(id, "flag " + (String)flag);
}
else if (msg.text.equalsIgnoreCase("light 1 2 3 on")) {
myBot.sendMessage(id, "Light 1,2 And 3 On");
if (flag == 0 || flag ==1){
    flag = 1;
    digitalWrite(relay1 ,HIGH);
    digitalWrite(relay2 ,HIGH);
    digitalWrite(relay3 ,HIGH);
    digitalWrite(relay4 ,LOW);
}
//    myBot.sendMessage(id, "flag " + (String)flag);
}
else if (msg.text.equalsIgnoreCase("light 1 3 4 on")) {
myBot.sendMessage(id, "Light 1,3 And 4 On");
if (flag == 0 || flag ==1){
    flag = 1;
    digitalWrite(relay1 ,HIGH);
    digitalWrite(relay2 ,LOW);
    digitalWrite(relay3 ,HIGH);
    digitalWrite(relay4 ,HIGH);
}
//    myBot.sendMessage(id, "flag " + (String)flag);
}
else if (msg.text.equalsIgnoreCase("light 1 2 4 on")) {
myBot.sendMessage(id, "Light 1,2 And 4 On");
if (flag == 0 || flag ==1){
    flag = 1;

```

```

        digitalWrite(relay1 ,HIGH);
        digitalWrite(relay2 ,HIGH);
        digitalWrite(relay3 ,LOW);
        digitalWrite(relay4 ,HIGH);
    }
//    myBot.sendMessage(id, "flag " + (String)flag);
}
else if (msg.text.equalsIgnoreCase("light 2 3 4 on")) {
myBot.sendMessage(id, "Light 2,3 And 4 On");
if (flag == 0 || flag ==1){
    flag = 1;
    digitalWrite(relay1 ,LOW);
    digitalWrite(relay2 ,HIGH);
    digitalWrite(relay3 ,HIGH);
    digitalWrite(relay4 ,HIGH);
}
//    myBot.sendMessage(id, "flag " + (String)flag);
}
else if (msg.text.equalsIgnoreCase("1 light on")) {
myBot.sendMessage(id, "1 Light On");
if (flag == 0 || flag ==1){
    flag = 1;
    digitalWrite(relay1 ,HIGH);
//    digitalWrite(relay2 ,LOW);
//    digitalWrite(relay3 ,LOW);
//    digitalWrite(relay4 ,LOW);
}
//    myBot.sendMessage(id, "flag " + (String)flag);
}
else if (msg.text.equalsIgnoreCase("2 light on")) {
myBot.sendMessage(id, "2 Light On");
if (flag == 0 || flag ==1){
    flag = 1;
    digitalWrite(relay1 ,HIGH);
    digitalWrite(relay2 ,HIGH);
//    digitalWrite(relay3 ,LOW);
//    digitalWrite(relay4 ,LOW);
}
//    myBot.sendMessage(id, "flag " + (String)flag);
}
}

```

```

else if (msg.text.equalsIgnoreCase("3 light on")) {
myBot.sendMessage(id, "3 Light On");
if (flag == 0 || flag ==1){
    flag = 1;
    digitalWrite(relay1 ,HIGH);
    digitalWrite(relay2 ,HIGH);
    digitalWrite(relay3 ,HIGH);
//    digitalWrite(relay4 ,LOW);
}
//    myBot.sendMessage(id, "flag " + (String)flag);
}

else if (msg.text.equalsIgnoreCase("emergency off")) {
myBot.sendMessage(id, "Emergency Light Off");
if(flag == 1){
    flag = 0;
}
//    myBot.sendMessage(id, "flag " + (String)flag);
}
}
delay(500);
status1 = "";
status2 = "";
status3 = "";
status4 = "";
}

```

## Lampiran 2. Program Arduino

```
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
#include "ACS712.h"

ACS712 sensor1(ACS712_30A, A0);
ACS712 sensor2(ACS712_30A, A1);
ACS712 sensor3(ACS712_30A, A2);
ACS712 sensor4(ACS712_30A, A3);

LiquidCrystal_I2C lcd(0x27, 16, 2);

float arus1, arus2, arus3, arus4;
float reg1, reg2, reg3, reg4;
int jam, menit;
String kirimesp;
float batas = 0;

void setup() {
  // put your setup code here, to run once:
  Serial.begin(115200);
  lcd.begin();
}

void loop() {

  sensor1.calibrate();
  sensor2.calibrate();
  sensor3.calibrate();
  sensor4.calibrate();

  reg1 = sensor1.getCurrentAC();
  reg2 = sensor2.getCurrentAC();
  reg3 = sensor3.getCurrentAC();
  reg4 = sensor4.getCurrentAC();

  // arus1 = sensor1.getCurrentAC();
  // arus2 = sensor2.getCurrentAC();
```

```

// arus3 = sensor3.getCurrentAC();
// arus4 = sensor4.getCurrentAC();

    arus1 = (0.2256 * reg1) - 0.0126;
    arus2 = (0.2256 * reg2) - 0.0126;
    arus3 = (0.2256 * reg3) - 0.0126;
    arus4 = (0.2256 * reg4) - 0.0126;

    if (arus1 < batas){
        arus1 = 0;
    }
    if (arus2 < batas){
        arus2 = 0;
    }
    if (arus3 < batas){
        arus3 = 0;
    }
    if (arus4 < batas){
        arus4 = 0;
    }
    lcd.setCursor(0, 0);
    lcd.print("L1:");
    lcd.print(arus1);
    lcd.setCursor(8, 0);
    lcd.print("L2:");
    lcd.print(arus2);
    lcd.setCursor(0, 1);
    lcd.print("L3:");
    lcd.print(arus3);
    lcd.setCursor(8, 1);
    lcd.print("L4:");
    lcd.print(arus4);

    String kirim = "";
    while(Serial.available()>0){
        kirim += char(Serial.read());
    }
    kirim.trim();
    if(kirim == "Meminta Data Dari Arduino..."){

```

```
    kirimesp = String(arus1) + "#" + String(arus2) + "#" +  
String(arus3) + "#" + String(arus4);  
    Serial.println(kirimesp);  
}  
  
delay(500);  
lcd.clear();  
kirim = "";  
// put your main code here, to run repeatedly:  
  
}
```